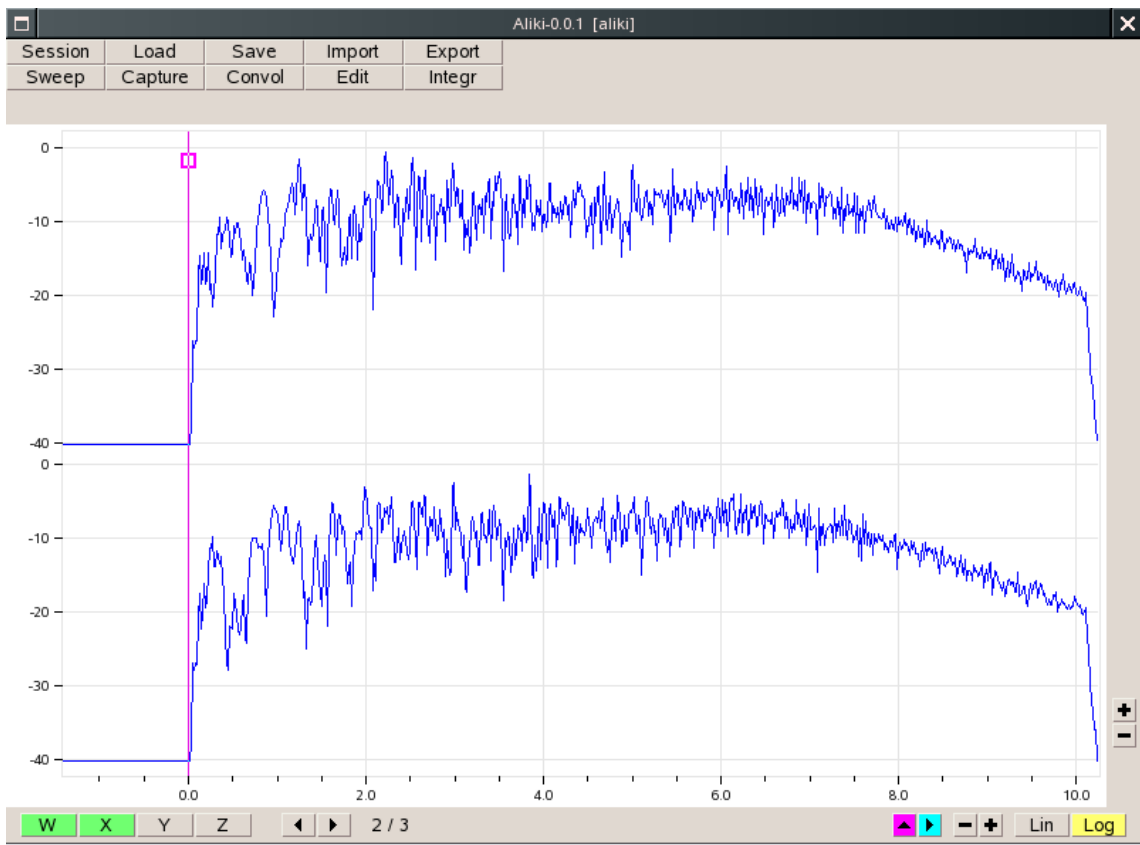


# Aliki - 0.0.1

## User Manual





---

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Running Aliko . . . . .	3
1.3	The Aliko file format . . . . .	4
1.4	The session directory . . . . .	4
<b>2</b>	<b>Viewing waveforms</b>	<b>5</b>
<b>3</b>	<b>The sweep generation window</b>	<b>7</b>
<b>4</b>	<b>The capture window</b>	<b>8</b>
4.1	Inputs and outputs . . . . .	8
4.2	Measurement parameters . . . . .	9
4.3	Performing the measurements . . . . .	9
<b>5</b>	<b>The convolution window</b>	<b>10</b>
<b>6</b>	<b>The editing window</b>	<b>11</b>
<b>7</b>	<b>Importing and exporting</b>	<b>13</b>
7.1	Importing sound files . . . . .	13
7.2	Exporting .ald files . . . . .	13



## 1 Introduction

This is the user manual for the first (alpha) release of Aliki.

This release is basically the same program as demonstrated at LAC2006, just cleaned up a bit. Many features are still missing, as is for example any decent form of error reporting.

As I'm still searching for the right 'model' that will encompass all planned features in an understandable and easy to use structure, future versions of Aliki may be quite different from this one. But it works and it can be used for practical work.

### 1.1 Installation

Compiling and installing Aliki should be simple enough. Unpack the source tarball, *cd* to it, then *make* and as root *make install*. The default install directory is **/usr/bin**. To change this, just edit the **PREFIX** variable in the Makefile.

Alike depends on a number of shared libraries. These are:

- **libclthreads** version 2.2.0-1 or later,
- **libclalsadrv** version 1.1.0-1 or later,
- **libclxclient** version 3.3.1 or later,
- **libsndfile**, any recent version,
- **libfftw3f**, any recent version.

The first three are available from the same place as the Aliki sources. The last two come with all Linux distributions. It may be worth the effort to compile the `fftw3f` library locally using optimisation options (e.g. enabling SSE) adapted to your platform.

### 1.2 Running Aliki

Running `aliki -h` will show the command line options:

```
Aliki-0.0.1
(C) Fons Adriaensen
<fons.adriaensen@skynet.be>

-h          Display this text
-J          Use JACK
-A          Use ALSA, with options:
  -d <device>   Alsa device [hw:0.0]
  -r <rate>     Sample frequency [48000]
  -p <period>   Period size [1024]
  -n <nfrags>   Number of fragments [2]
-E <prefix>   Prefix for audio process, e.g. sudo
```

Without either `-J` or `-A` only the processing functions are available.



Most of this will be clear enough, but the **-E** option may need some comments.

The audio interface part of Aliko is a separate executable (`aliko-rt`) which is started by the main program if an audio interface is requested on the the command line (**-A** or **-J**). The two parts communicate via shared memory. The main reason for this separation is to avoid JACK locking *all* the memory blocks used by the processing code. These could be quite large, there is no need to lock them, and doing so could easily deplete physical memory on some systems.

It is preferable to run `aliko-rt` with real-time privileges. How this is done depends on your system. On some systems nothing special is required. On others, `aliko-rt` needs superuser status in order to be able to run in real time mode, and this may require a prefix command such as `sudo` or `rtlimits`. The **-E** option allows to specify such a prefix.

Aliko does not depend on low latency (latency can be compensated for later), so you may use larger period sizes than would be required for interactive audio work.

Note that it is possible to run Aliko without any audio interface at all, for example to process previously captured files.

### 1.3 The Aliko file format

Aliko uses its own file format with the filename extension **.ald**. The same file format will be used by some other applications (in development) that use impulse responses.

An **.ald** file is a multi-channel, multi-section audio file using single precision floating point as the sample format. Apart from the audio signals some other information is stored. For example, special formats such as Ambisonics B-format or MS stereo can be identified and treated correctly, even if they have the same number of channels as some other formats.

All files used by Aliko can be exported to WAV format for external use. Future versions will also allow conversion of external files to the **.ald** format.

### 1.4 The session directory

The current version of Aliko does not have a real 'session' concept, but it does depend on a particular directory structure. A session directory should contain at least the following subdirectories:

- **sweep**: sweep and equalisation files.
- **capture**: recorded room responses.
- **imprsp**: raw impulse responses.
- **edited**: modified impulse responses.

It is absolutely essential for the current version (which lacks decent error reporting) that the session directory is set correctly. Failing to do so may lead to mysterious failures. A new session directory can be initialised from the file window (see below).

To set the session directory, click **Session** in the main window. A filer window will open for the current directory. To create a new session directory, navigate to the place where you want it. Use the **New dir** function to create the directory. Then click **Session** to create the required subdirectories. To select an existing session directory, navigate to it (so you see the required subdirs), then click **Session**.

## 2 Viewing waveforms

The main window normally displays the waveform(s) in the currently loaded file, if any.

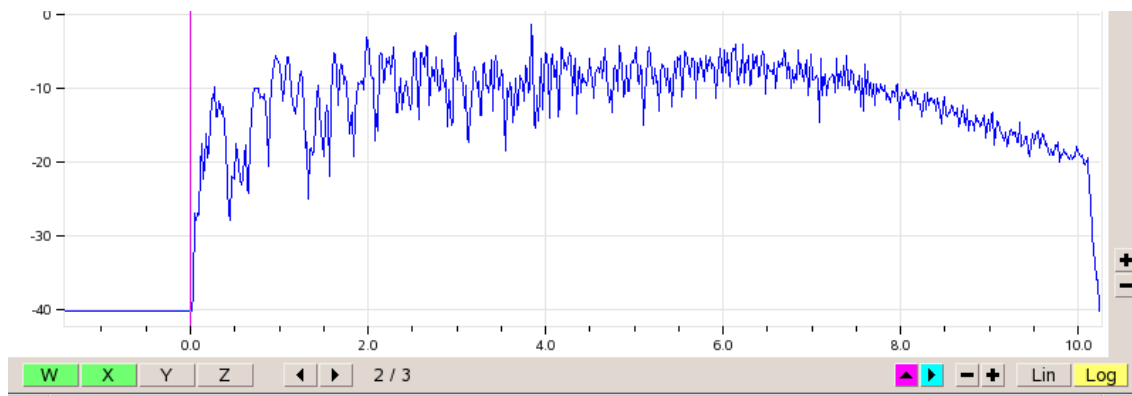


Figure 1: Bottom of main window

At the bottom (fig.1) are some controls for this waveform display.

To scroll the display horizontally, just L-drag the time scale.

The controls at the left control which part of a file is shown.

If the file is a multichannel file, there will be up to four channel selection buttons. The labels depend on the file type - in the figure an Ambisonics B-format file is shown. L-click selects a single channel, R-click adds a channel to the current selection. These buttons are not shown if the file only has a single channel.

If the file has more than one section, there will be left and right arrow buttons selecting resp. the previous and next section. Again these buttons will not be shown if there is only a single section. The index of the current section and the number of sections are always shown.

At the right are a set of buttons that control the markers, zooming and the vertical scale.

Up to two markers can be put on the waveform display. A marker is a vertical line corresponding to a particular sample offset. It is used for various purposes, for example when editing an impulse response.

To set a marker, R-click in the display area at the place where you want to put a marker. To move a marker, L-drag it by the square near the top.

When a marker is set, this is indicated by one of two buttons lighting up in the corresponding color and showing an arrow instead of a square. The arrow indicates where the marker is: up if it is on screen, left or right if a marker is set but not visible.

When a marker is set, then an L-click on the corresponding button scrolls the display to the marker position so that it is put at the centre.

An R-click on a marker button will erase the marker if it is on-screen, or bring it on-screen otherwise. The previous position is lost in both cases. This can be handy when you want to re-use an existing marker that has gone off-screen.

The + and - buttons to the right control the horizontal scale factor, i.e. they zoom in or out. L-click moves one level up or down, R-click goes faster.



The zoom focus is the most recently referenced marker if still set and visible, or the centre of the display area otherwise. In other words, when you go to a marker clicking its button, or when you move it, it also becomes the zoom focus.

The **Lin** and **Log** buttons select either a linear or logarithmic vertical scale. The **+** and **-** buttons on the right margin modify the vertical scale. For the linear format they just change the scale factor. For the logarithmic one they modify the lower limit of the displayed range.

### 3 The sweep generation window

A sweep file is a single-section, two-channel .ald file. The first channel contains the 'forward' sweep, with rising frequency, used by the capture function. The second channel is the 'inverse' sweep used for deconvolution. The convolution of these two will always be a Dirac pulse bandlimited to the frequency range of the sweep.

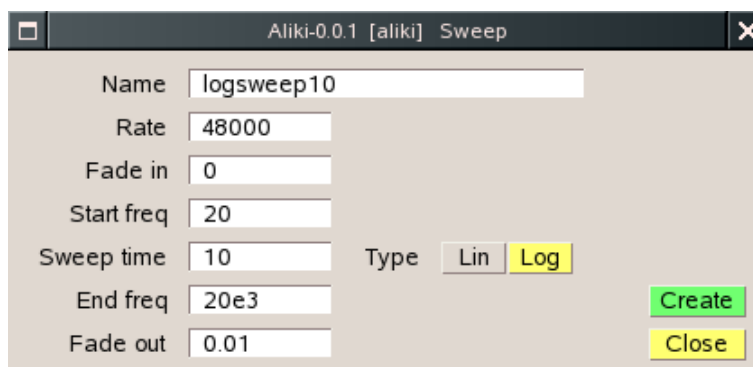


Figure 2: The sweep generator window

The following parameters (fig.2) should be filled in to generate a sweep:

- **Name:** filename without the .ald extension. The file will be put into the SWEEP subdirectory of the session.
- **Rate:** the sample rate. This will be filled in if the audio interface is active, otherwise type in the required value.
- **Fade in:** a soft fade-in can be specified. The parameter is the time in seconds. It is normally not required as the generated sweep will always start at a zero-crossing.
- **Start freq:** the lower limit of the frequency range, in Hz.
- **Sweep time:** the sweep time in seconds.
- **Type:** either linear or logarithmic.
- **End freq:** the upper limit of the sweep range, in Hz.
- **Fade out:** a fade-out time in seconds after the upper frequency limit has been reached. This is used to avoid a discontinuity at the end. Note that the frequency will continue to move up during the fade-out, so this should be short to avoid going above half the sample frequency in some cases.

When all parameters are filled in, click the Create button to generate the sweep.

## 4 The capture window

This window will be available only if Alik was started with an audio interface option (-A or -J). It controls all aspects of the actual IR measurement operation.

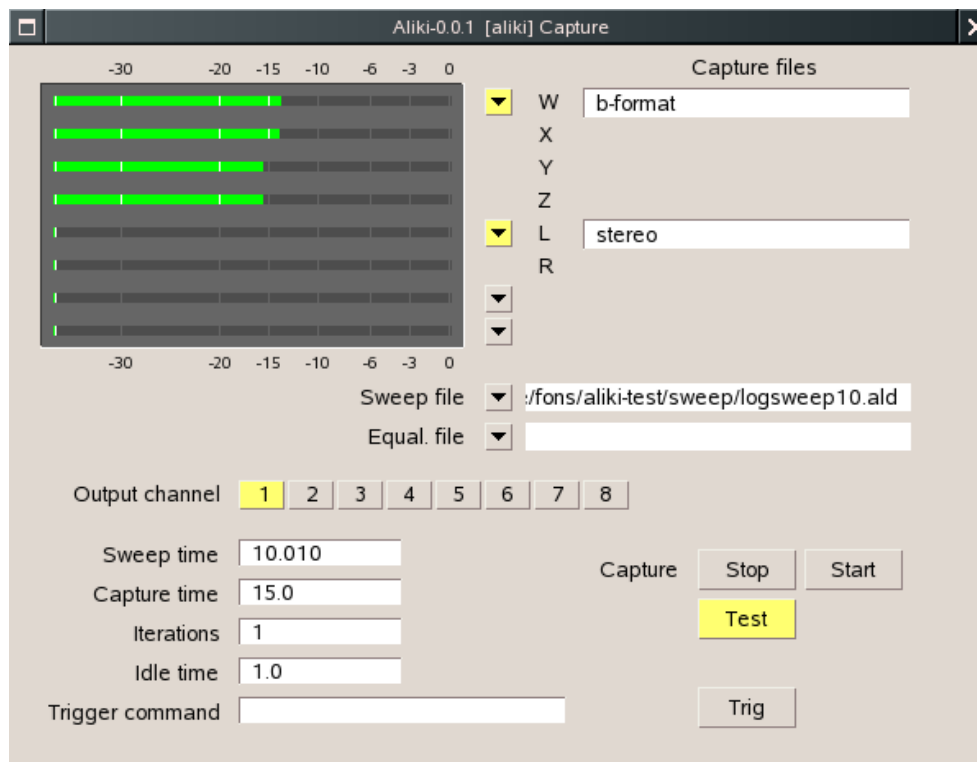


Figure 3: The capture window

### 4.1 Inputs and outputs

The top left part of this window (fig.3) shows the level meters for each input channel. To the right of these are the controls to set file names and types for the recorded sweeps.

An L-click on any of the arrow buttons besides the level meters will bring up a menu of available file types. Select the number of channels and type you need. The input channels will be labeled accordingly and a text input field for the file name will appear. Recorded files will be stored in the CAPTURE subdirectory of the session. To remove an input file from the list, use the Remove option in the file type menu.

The sweep file to be used can be selected from the SWEEP directory. Make sure to select one with the correct sample rate - this is not checked in the current release. Using the wrong sample rate will work, but the frequency range will be shifted.

Future versions will allow a loudspeaker equalisation file to be selected from the SWEEP directory, and also provide a function to compute such files. This is currently not implemented.

Any mix of up to eight output channels can be selected by the buttons 1...8. In most cases a single source will be used for IR measurement.



## 4.2 Measurement parameters

Below the output selection buttons are the fields that control the actual capture operation.

- **Sweep time** is filled in from the sweep file, and can not be changed.
- **Capture time** sets the length of the recorded signals in seconds. It should be at least equal to the sum of sweep time, audio latency, expected IR length time and some safety margin.

Aliko allows to automate sequences of measurements using the same parameters, for example when measuring the polar diagram of a speaker. This feature can also be used to record a set of IR using a sequence of different sources. In this case the output switching must be done manually (this will be automated as well in later versions).

- **Iterations** is the number of automatically sequenced measurements. Each of them will correspond to a new section in the capture files.
- **Idle time** is the time in seconds between two measurements, i.e. the time required to rotate a speaker, move a microphone, or whatever. Nothing will be recorded during this time. At the end of the idle time the next sweep playback and the recording of the next section of the capture files will start simultaneously.
- **Trigger command** can be any external program. It is executed at the start of the idle time after each measurement and can be used for example to trigger a rotating table to move to the next position.

## 4.3 Performing the measurements

The buttons in the lower right corner control the actual measurement.

A **Test** signal (an FM modulated sine) can be output to adjust levels before a measurement. The test output is at -15 dB, and should result in about the same level returning on the inputs. The actual sweeps will be at -6 dB, so this should be taken into account when setting up the levels using the test tone.

The measurement is started using the **Start** button. The **Trig** button will flash at the start of the idle time after each measurement. If you perform an automatic sequence using multiple sources this is the right time to switch them.

The measurement sequence is finished when the **Start** button lights again. It can be terminated at any time using **Stop**. In that case any partially recorded capture files will be deleted.

## 5 The convolution window

This is used to deconvolve the recorded sweeps and obtain the impulse response(s). Use **Load** in the main window to load a capture file before selecting the convolution function.

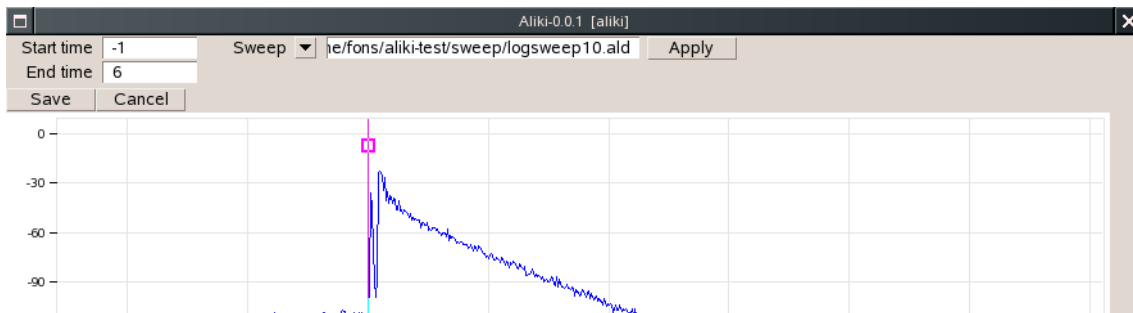


Figure 4: Convolution controls

**Start time** and **End time** are relative to the position where the impulse response would start without any audio latency and delay. A negative **Start time** can be used when using a log sweep to see the harmonic distortion of the system being tested - each harmonic will show up as a separate IR at some place in negative time depending on the sweep rate. The **End time** should be set in function of the expected impulse length. A good value is the expected reverb time plus two or three seconds.

The **Sweep file** should of course be the same one as used during the measurement.

When all parameters are set, click **Apply** to start the deconvolution on the current section. This may take a few seconds (there is no progress bar yet, but it will be added in future releases). The result will be displayed (and you can zoom in, scroll etc.) but not yet stored. For a multi-section file, moving between sections will start the deconvolution of the next or previous section. Again this may take some time.

When the parameters seem OK, the deconvolution can be done for all sections and the results saved by clicking **Save**. It is not necessary to have moved through all sections before as everything will be recomputed anyway. The resulting file is stored in the IMPRESP subdirectory of the session, and also becomes the currently loaded file.

Clicking **Cancel** will return to the main window without storing anything. The capture file remains the current file in that case.

## 6 The editing window

This function is used to prepare IR for practical use. Enter this window with an IR file from the IMPRESP directory loaded.

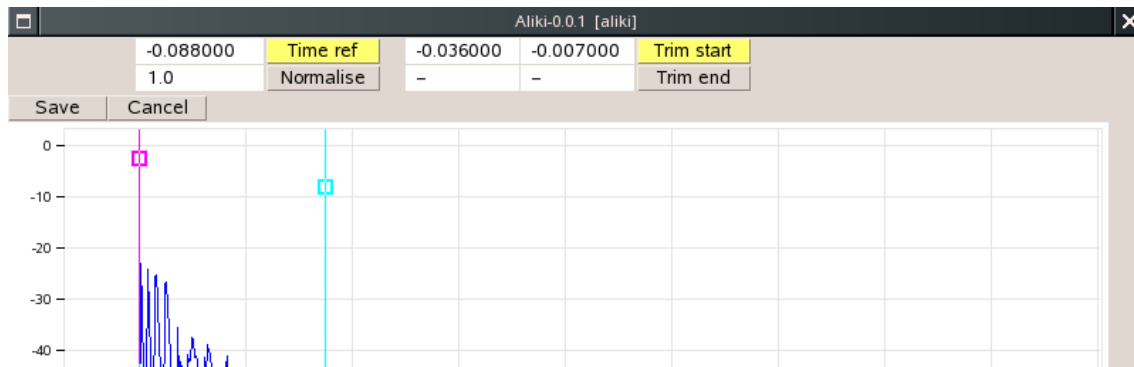


Figure 5: Editing controls

Four operations are provided. All of them are non-destructive until the result is saved. They can be also be repeated any number of times and be can be undone by R-clicking the corresponding button.

- **Setting the reference time.** This will not normally be necessary, but is provided to be used while preparing files for another application which needs this. It can also be useful just to align the time axis while examining an IR.  
Put a single marker at the place that should correspond to 'zero' time (zoom in to set it precisely). L-click Time Ref to move the time axis so it becomes zero at the marker position.
- **Amplitude normalisation.** Put a single marker on the direct sound peak of the IR (zoom in to set it precisely). L-click Normalise to normalise to unit energy in a small region around this point.
- **Trimming the start.** This normally used to remove any debris before the start of the IR, or to remove the direct sound or even some early reflections. Put two markers at the times where you want a raised cosine fade-in to start and reach unity gain respectively, then L-click **Trim start**.
- **Trimming the end.** This normally used to remove the noise tail at the end of an IR. Put two markers at the times where you want a raised cosine fade-out to start and reach zero gain respectively, then L-click **Trim end**.

When moving between sections of a multi-section file the currently active settings are applied to each newly displayed section.

When the result is saved, the current values are applied to all sections of the file, so relative timing and gain are preserved. The signal parts before the fade-in and after the fade-out are removed. The output file will be put into the EDITED subdirectory of the session, and also



---

become the currently loaded file. Cancel will terminate the editing without storing anything, and keep the current file.

## 7 Importing and exporting

### 7.1 Importing sound files

This is not implemented in the current version.

### 7.2 Exporting .ald files

The currently loaded file can be exported to floating point WAV format. Each section should be exported separately, and multi-channel .ald files are split into separate WAV files. Click **Export** and fill in a name without the extension. A channel suffix and the extension will be added automatically. The option to export all sections in a single operation will be added in a future version.