Manual

AudioSpike

Spike Detection and Evaluation Software Package
from release 2.0.0.0

Contents

1 Prerequisites . . . 4
1.1 Technical requirements . . . 4
1.2 Getting started . . . 4
1.3 Definitions . . . 5
1.4 Abbreviations . . . 5
2 AudioSpike XML files . . . 6
2.1 Template files . . . 6
2.2 Result files . . . 10
3 Operation . . . 13
3.1 The main screen . . . 13
3.2 Settings . . . 14
3.2.1 Input and Output Filters . . . 18
3.3 Subscreens . . . 19
3.3.1 The stimulus window . . . 19
3.3.2 The epoche window . . . 19
3.3.3 Cluster windows . . . 21
3.3.4 The PSTH windows . . . 22
3.3.5 Parameter windows . . . 23
3.3.6 The spike windows . . . 26
3.4 The batch mode . . . 27
3.5 The manual search mode . . . 28
3.6 The free search mode . . . 29
4 Calibration and Equalisation of Loudspeakers and Microphones . . . 30
4.1 RMS values . . . 30
4.2 Calibration and Equalisation in Standard-Mode . . . 31
4.2.1 Spectral Equalisation . . . 31
4.2.2 Impulse Response . . . 33
4.2.3 Level Calibration . . . 34
4.3 Calibration and Equalisation in In-Situ-Mode . . . 35
4.3.1 Calibration of the Probe Microphone . . . 35
4.3.2 Calibration of the In-situ Loudspeaker . . . 36
5 Appendix . . . 37
5.1 Command line parameters . . . 37
5.2 Tools for MathWorks MATLAB® and GNU Octave . . . 38
5.2.1 Structure of MAT-files . . . 38
5.3 Equalisation filter file format . . . 39

1 Prerequisites

AudioSpike is a software package for electrophysiological measurements using acoustical stimulation for Microsoft® Windows®. Instead of using proprietary hardware to record electrode signals AudioSpike uses standard sound cards for acoustcial stimulation and recording electrode signals. The recorded spikes can be evaluated using multichannel cluster analysis and PSTH selection in realtime.

1.1 Technical requirements

The following requirements have to be fullfilled to use AudioSpike:

- AudioSpike requires a standard computer with recent hardware. The more power the computer has, the more channels and realtime evaluation windows can be handled simultaneously.

- AudioSpike requires Microsoft® Windows 7® or higher. Windows 10® or higher is recommended.

- AudioSpike must **never** be run from a network drive or other slow drives. AudioSpike does harddisk recording and thus requires fast drives that are always accessible.

- AudioSpike uses the ASIO sound driver interface by Steinberg. Only soundcards shipped with a corresponding driver can be used with AudioSpike.

- AudioSpike records electrode signals through sound card inputs. For recording electrode signals you will ususally need microphone pre-amplifiers (built-in or external) to reach sufficient signal levels or signal-to-noise ratios respectively. Additionally you might need impedance converters to adjust the electrodes impedance to the input impedance of sound cards.

- AudioSpike is shipped with tool scripts to interface between MATLAB® or GNU Octave and AudioSpike. To use these tool scripts MATLAB® or GNU Octave is required.

1.2 Getting started

AudioSpikeis shipped with a 32-bit version and a 64-bit version of all binaries. Two icons for the two versions are created on the desktop and in the start menu. The versions are 100% compatible concerning the data formats. The 64-bit version is especuially useful when sing huge measurement paradigms with many stimuli and detected spikes because it overcomes the 2GB limit of 32-bit applications.
Currently the 32bit version is only shipped as a fallback: the 64-bit version was introduced with version 2.0.0.0 and although it is thoroughly tested it may cause unexpected problems. The 32-bit version may be discontinued later without notice.
Important note: you may need to adjust your between MATLAB® or GNU Octave scripts respectively to use the 64-bit version: in calls to AudioSpike change the "bin" path to "bin64".

To get familiar with the multiple possibilities of AudioSpike it is recommended to do the following steps:

- Check out the example result file shipped with AudioSpike. This way you will get familiar with the multiple screens available in AudioSpikewithout the need to set up your hardware and collect own data in the beginning. The example is contained in the file

  ```
  results\example\result_0001.xml
  ```

  of the installation. Please load it from the main screen. It contains a complete (dummy) measurement for one channel (electrode) with the parameters "level" and "frequency". A cluster window is preconfigured. You can re-scan the measured epoches and change the cluster and/or PSTH selections, thresholds, set up new cluster windows and so on to get used to the operation within the different AudioSpike windows. Please refer to the chapter 3 for details.

- Then you may adjust all settings of AudioSpike according to your hardware. Please refer to the chapter 3.2 for a detailed description of all settings. Be sure to run the trigger test to ensure correct wiring and run the equalisation and calibration described in the same section for all used channels.

- Check out the free-search mode, where you don't have to set up an own measurement template (see chapter 3.5).

Afterwards you can take a look at the examples shipped with AudioSpike:

- If you want to use MATLAB® or GNU Octave to setup your measurements it is recommended to take a look at the M-files in the directory `examples\M-files`. Please refer also to the appendix (chapter 5.2). The scripts contain very detailed comments.

- If you want to set up own XML measurement templates with other tools take a look at the M-files in the directory `examples\XML`. Two very simple XML files are contained, one complete "measurement" and one file to demonstrate how to append more stimuli to an existing measurement. To test the append mode you have to load `example.xml` and start it. After stopping, save it, afterwards appending is possible. Then use the "Append" button or menu item to append the file `example_append.xml`. **Note**: note the special location of new levels to be appended within the "Settings" subnode rather than in the "Parameters" subnode (see also chapter 2).

1.3 Definitions

The following definitions are used in this manual:

Definition	**Description**
Measurement template	A "measurement template" is an XML file containing the definition (paradigm) of a complete measurement. It contains all variables, stimuli descriptions and presets for a measurement. A template can be created with an XML editor or by converting MATLAB® structures to XML using tools shipped with AudioSpike (see appendix, chapter 5.2).
Measurement result	A "measurement result" is an XML file containing complete measurement template and the complete results (spikes, evaluations...). A result can be loaded to MATLAB® structures using tools shipped with AudioSpike (see appendix, chapter 5.2).
AudioSpike batch file	A file containing a list of measurement templates to be measured subsequently .
Epoche	An "epoche" is one spike train recorded as a response to a single acoustical stimulation, i.e. the signal recorded on one electrode after acoustical stimulation. The "epoche length" is defined in the measurement template.

1.4 Abbreviations

The following abbreviations are used in this manual:

Abbreviation	**Description**
ASIO	Audio Stream Input/Output by Steinberg. Professional sound driver model for synchroneous input/output for Microsoft® Windows®.
PCM	Puls-Code-Modulation
PSTH	Peri-Stimulus Time Histogram. Histograms to visualize the rate and timing of neuronal spike discharges in relation to an external stimulus or event.

RMS	Root Mean Square. Square root of the arithmetic mean of the squares of a set of numbers (here: audio samples).
XML	Extensible Markup Language. Standard text file format used for templates and results in AudioSpike.

2 AudioSpike XML files

Mesurements are configured in AudioSpike using XML files containing the definition (paradigm) of a complete measurement. It contains all variables, stimuli descriptions and presets for a measurement. A template can be created with any XML editor or by setting up corresponding structures in MATLAB® or GNU Octave respectively and converting them to XML using tool scripts shipped with AudioSpike (see appendix). Tools for reading XML result files to MATLAB® or GNU Octave respectively are included as well.
AudioSpike uses a very simple XML scheme to read and write data that correspond to (nested) structures in MATLAB® or GNU Octave.
The root node must be named “AudioSpike”. Within this node fields are always represented by an XML node, values of fields are always represented by the text within a node, e.g.

```
<AudioSpike>
   <SubStruct>
     <ValueName1>Value1</ValueName1>
     <ValueName2>Value2</ValueName2>
     ....
   </SubStruct>
....
</AudioSpike>
```

2.1 Template files

A simple example for a measurement template file can be found in
`examples\XML\example.xml`
of the installtion. In is recommended to open this file in an XML editor or text editor while reading this description.
A measurement template file must contain at least the three mandatory nodes “Settings”, “Parameters” and “Stimuli”.
The node “Settings” contains general settings for a measurement, it does not contain subnodes. The following table shows all defined values, where an empty default value means, that the value is mandatory, “AudioSpike” means, that the default value is used from the AudioSpike settings (settings dialog or GUI respectively).

Table 3: XML “Settings” node

Name	Description	Default value
SaveSpikeParams	If set to 0 then the spike parameters (amplitudes, times, see table 15) are not saved for each spike separately	
InSitu	Switches AudioSpike to In-Situ-Mode, see also 4.3	0
SampleRate	Samplerate for soundcard to be used	AudioSpike
SampleRateDevider	Downsampling devider. The signals recorded from the inputs (electrodes) are downsampled by this factor. **Note**: if you have selected the option to save the probe microphone recordings, they are downsampled as well.	1

EpocheLength	Epoche length in seconds. EpocheLength is the length of a recorded repetition that is analyzed for spikes. It must be smaller or equal to RepetitionPeriod (see below)	
PreStimulus	Length of silence within an epoche before stimulus is played in seconds	
RepetitionPeriod	time between epoches in seconds. After this time the next stimulus starts.	
StimulusRepetition	number of times the stimulus list is presented. If randomization is activated the stimulus list is shuffled before each repetition	1
SpikeLength	Total length in milliseconds to extract/plot for each spike (Pre-Threshold + time from threshold crrossing to spike end)	AudioSpike
PreThreshold	Time in milliseconds to extract for each spike before thresold crossing	AudioSpike
PostThreshold	Time in milliseconds to wait AFTER threshold crossing, before next threshold crossing is interpreted as next spike. Please note, that the PostThreshold value determines the temporal resolution of successive spike-events. NOTE: if you are specifying a very short PostThreshold the same spike might be detected multiple times. If this value is 0 then the PostThreshold is calculated to PostThreshold = SpikeLength – PreThreshold.	AudioSpike
RandomMode	switch random order of stimuli on/off (1/0)	0
OutputChannels	Output channels to be used. **Note:** if more than one channel is specified, the channels have to be put into squared brackets, e.g. [1 2]. Indices are 1-based, really used hardware channels depend on the channels selected in the settings of AudioSpike	
InputChannels	Input channels to be used, see note for OutputChannels	
Thresholds	Threshold values for spike detection to be used. One value per input channel. If more than one channel is specified, the thresholds have to be put into squared brackets, e.g. [-0.12 -0.3]. The values are rel. fullscale i.e. between -1 and 1	AudioSpike
Level	**Important note:** if you are appending data to an existing measurement, then the parameter “Level” must **not** be added to the “Parameter” subnodes (see below) but to the settings node here. The format then must be identical to the already existing levels, e.g. for to add a new pair of levels to a two-channel measurement of 50 dB on the first and 60 dB on the second channel use `[50 60]`.	

The node “Parameters” contains subnodes “Parameter” for each variable stimulus parameter (such as level, frequency...), e.g.

```
<Paramaters>
  <Parameter>
```

```
      <ValueName1>Value1</ValueName1>
      <ValueName2>Value2</ValueName2>
      ....
    </Parameter>
    <Parameter>
      <ValueName1>Value1</ValueName1>
      <ValueName2>Value2</ValueName2>
      ....
    </Parameter>
....
</Parameters>
```

The following table shows all values defined for each "Parameter" subnode, where an empty default value means, that the value is mandatory, "empty" means that it is optional:

Table 4: XML "Parameter" subnodes

Name	Description	Default value
Name	Name to be displayed for a parameter. Must be unique within parameters.	AudioSpike
Unit	Unit to be displayed for a parameter	empty
Log	Flag, if a parameter should be displayed with logarithmic axis in plots (0 or 1)	0
String	Flag, if values of a parameter are strings rather than numbers. If this flag is set to 1, then the value for this parameter in each stimulus is interpreted as text, in plots these text values are spaced equally (Log is neglected)	0

Level	For the special parameter with name "Level" all used level values are specified here within the "Parameter" node rather than in the "Simulus" subnodes (see below). All level values for the same channel to be used have to be separated by a semicolon, channels are separated by a blank, the complete list has to be enclosed into squared brackets.
	Example 1: levels 50, 60 and 70 dB to be used for ALL output channels:
	`[50;60;70]`
	Example 2: levels 50, 60 and 70 dB to be used for channel 1 levels 90, 80 and 70 dB for channel 2 (fix "pairs" of levels):
	`[50 90;60 80;70 70]`
	Note: if you are setting up a template to be appended to an existing measurement, the new levels to be appended have to be specified in the "Settings" subnode (see above).	

The node "Stimuli" contains subnodes "Stimulus" for each stimulus, e.g. .

```
<Stimuli>
```

```
   <Stimulus>
     <ValueName1>Value1</ValueName1>
     <ValueName2>Value2</ValueName2>
     ....
   </Stimulus>
   <Stimulus>
     <ValueName1>Value1</ValueName1>
     <ValueName2>Value2</ValueName2>
     ....
   </Stimulus>
....
</Stimuli>
```

The following table shows all values defined for each "Stimulus" subnode, where an empty default value means, that the value is mandatory, "empty" means that it is optional:

Table 5: XML "Stimulus" subnodes

Name	**Description**	**Default value**
Name	Name to be displayed for the stimulus in the stimulus list window of AudioSpike.	AudioSpike
FileName	The filename of the audio file of this stimulus.	
RMS	RMS of the audio signel. If value is empty AudioSpike calculates the numerical RMS of the audio data. **Important note**: the RMS value is used for calibrated signal output in AudioSpike. The numerical RMS value does not necessarily represent a signal's energy accurately. Please read detailed descriptions in chapter 4 and especially in chapter 4.1 thoroughly!	empty
PARAMETERNAME	Value of PARAMETERNAME for this stimulus. This value is mandatory, i.e. a value for each parameter defined in subnode "Parameters" must be specified, except for paramater "Level" (see below) If the "String" flag of the corresponding parameter is set to 1, the value is interpreted as text, otherwise the value must be numerical.	

The parameter "Level", it is handled separately: the levels to be applied to all stimuli are specified directly within the corresponding "Parameter" subnode (see table 4).
On loading a template AudioSpike automatically generates a new subnode "AllStimuli" containing a copy of all stimuli defined in the "Stimuli" subnode for each configured level. The parameter names for levels for different channles are enumerated (Level_1, Level_2 ...). The different levels are generated within AudioSpike, no separate stimulus data (audio files) have to be generated for different levels.
In the example `as_simple_example.xml` this would lead to a total number of four stimuli within the "AllStimuli" subnode: 10kHz at 60 dB, 10 kHz at 70 dB, 20kHz at 60 dB and 20 kHz at 70 dB.
Optionally you can pre-define one or more cluster windows (see chapter 3.3.3) that are displayed directly on loading a template. Cluster windows can be pre-defined in the additional "Results" subnode, e.g.

```
<Result>
   <Clusters>
     <Cluster>
       <X>Peak1</X>
```

```
      <Y>Peak2</Y>
    </Cluster>
    ....
  </Clusters>
</Result>
```

Valid values for oth axis (“X” and “Y”) are all values listed in column “Abbreviation” of table 15.

2.2 Result files

Results are stored by AudioSpike in XML files in subfolders within the result path configured in the settings dialog of AudioSpike. A part of the result-subpath can be entered on each measurement start. AudioSpike enumerates this subpath starting with “0000” if subpaths with that name already exists. Result files are enumerated automatically as well, if the user decides on saving to create a new result file.
A example result file can be found in `results\result_0001.xml`.
The recorded epoches (electrode data, spiketrains) are stored as normalized raw 32-bit PCM data. Position information for the epoches within this file are contained in the result XML. The file `epoches.pcm` is always stored in the same directory as the result XML. If the corresponding option is selected in the settings, then recordings form the probe microphones are stored in in-situ mode in the same directory in file `probemics.pcm`. It contains the raw recordings during the epoches only, the positioning is identical to `epoches.pcm`. Note: the channels are sorted with ascending order of the audio output channels!
A result XML file will always contain a complete copy of the template file. On saving AudioSpike adds the following values to the “Settings” subnode:

Name	Description
EpocheLengthSamples	length of an epoche in recorded samples calculated from EpocheLength, SampleRate and SampleRateDevider.
SpikeLengthSamples	length of an epoche in recorded samples calculated from Spike-Length (settings dialog of AudioSpike), SampleRate and SampleRateDevider.

Note: it is strongly recommended to use these sample length values when accessing spikes from the result XML or epoches from epoches.pcm (see below) to avoid rounding errors when calculating these values with other tools!
On measurement startup AudioSpike creates the subnode “AllStimuli” containing all stimuli defined in subnode “Stimuli” for all levels defined in the level parameter as described above. When creating the “AllStimuli” subnode the parameter “Level” is split up into the enumerated channel levels (Level_1, Level_2 ...). The node “AllStimuli” contains all stimuli that are used in the stimulus list of AudioSpike.
All results are stored in the subnode “result”. The subnodes “Clusters”, “ParameterWindows” and “PSTH” contain corresponding window configuration and selection information for AudioSpike. These are only used when loading a result back into AudioSpike.
Information about all stored epoches are stored in separate “Epoche” subnodes within the subnode “Epoche”, e.g.

```
<Result>
  <Epoches>
    <Epoche>
      <Name>45kHz</Name>
      <FileName>../signals/45.wav</FileName>
      <Frequency>45</Frequency>
      <RMS>-3.02</RMS>
      <Level_1>60</Level_1>
      <Level>[60]</Level>
      <StimIndex>1</StimIndex>
      <RepetitionIndex>1</RepetitionIndex>
      <Done>1</Done>
    </Epoche>
    ...
```

All values that correspond to the stimulus that was played when recording an epoche are copied to the subnode, where the for parameter “Level” the original multichannel value is copied as well as the enumerated channel levels (Level_1, Level_2 ...). The following additional values are created for each epoche:

Name	**Description**
StimIndex	the stimulus index within the “AllStimuli” subnode that contains the corresponding stimulus.
RepetitionIndex	the index of the repetition, that this epoche belongs to
Thresholds	Threshold values for spike detection that were used within this epoche. Same value format as in “Thresholds” within “Settings” node.
Done	flag (1 or 0), if this epoche was finished and stored in epoches.pcm

The addtional field “StimulusSequence” in the “Result” subnode will contain the sequence of played stimuli indices for the complete measurement.
The file epoches.pcm will contain the recorded epoches in the following order (in this example for two recording channels):

```
epoche1_channel1 epoche1_channel2 epoche2_channel1 epoche2_channel2 .....
```

where the number of the epoche corresponds to the order of epoches within the “Epoches” subnode. Each channel epoche contains `EpocheLengthSamples` samples (normalized 32-bit floating point value). The byte position of epoche data for a particular epoche within the file `epoches.pcm` can be calculated by:

$$NumberOfChannels * EpocheLengthSamples * (NumberOfEpoches - 1) * SizeOfFloat$$

An example: if two channels were recorded, the fifth epoche of the first electrode starts at byte

$$2 * EpocheLengthSamples * 4 * 4$$

The same epoche position information hold for the optional probe microphone recordings in in-situ mode in file `probemics.pcm`. **Note:** the number of channels in `probemics.pcm`corresponds to the number of output (audio output)channels and **not** to the number of electrodes. The channels are sorted according to the order of the corresponding output channels.
Information about all detected spikes are stored are stored in separate “Spike” subnodes within the subnodes “Spikes” or “NonSelectedSpikes”, where the first contains all spikes that were selected in a cluster window on saving and the latter contains all other detected, but unselected spikes. The format within the “Spike” nodes is identical for both subnodes, e.g.

```
<Spikes>
  <Spike>
    <SpikeTime>0.13</SpikeTime>
    <SpikePosition>3037</SpikePosition>
    <StimIndex>1</StimIndex>
    <EpocheIndex>1</EpocheIndex>
    <Channel>1</Channel>
    <RepetitionIndex>1</RepetitionIndex>
    <Threshold>0.15</Threshold>
    <Frequency>45</Frequency>
    <Level>[60]</Level>
    <Data>........</Data>
  </Spike>
  ...
```

where :

Table 8: XML “Spikes” node

Name	Description
SpikeTime	time in seconds within epoche, where the threshold was reached for the current spike.
SpikePosition	same value in samples (use this value when accessing a spike within raw PCM data).
SpikeGroup	index of the “spike group”, i.e. the index of the selection rectangle in the cluster window (empty for NonSelectedSpikes), where 6 = blue rectangle 5 = red rectangle 4 = green rectangle 3 = turquois rectangle 2 = fuchsia rectangle 1 = lime green rectangle
StimIndex	fthe stimulus index within the “AllStimuli” subnode that contains the corresponding stimulus.
EpocheIndex	index of epoche within “Epoches” subnode that describes contains the corresponding epoche.
Channel	recording channel of the spike.
RepetitionIndex	the index of the repetition, that this epoche belongs to.
Threshold	threshold value that was used for the spike detection.
PARAMETER	values for all parameters of the corresponding stimulus (“frequency” and “level” in the example above).
Spike Parameters	all spike parameters described in table 15 named as the value from column “Abbreviation”. If value of “SaveSpikeParams” in settings node is set to enquote0 the spike parameters are NOT saved.
Data	the wave form (samples) of the spike as base64-encoded double values (64-bit floating point values).

3 Operation

3.1 The main screen

The top of the main screen of AudioSpike contains a main menu and a toolbar with the following functions:

Table 9: The main menu

Menu item	Description
File / Load Measurement Template	loads an XML file containing a measurement pardigm to be measured (see chapter 2).
File / Start Batch Mode	switches to batch mode and shows corresponding batch window (see chapter 3.4).
File / Load Measurement Result	loads a XML file containing a measurement result
File / Append To Measurement	Loads an XML file containing a measurement template to be appended to the current measurement (see chapter 2). **Note:** appending is only possible, if a measurement already contains data, i.e. it was started/stopped and saved, or if a result was loaded.
File / Save	Saves current measurement
Windows / Stimuli	Toggles visibilty of the stimulus list
Windows / Epoches	Toggles visibilty of the epoche (recorded electrode spike trains) window
Windows / Spikes	Toggles visibilty of spike window
Windows / PSTH	Toggles visibilty of PSTH (Peri-Stimulus-Time-Histogram) window
Windows / Stimulus-PSTH	Toggles visibilty of Stimulus-PSTH window. The plot contains a separate PSTH for each stimulus.
Windows / Cluster	Creates a new cluster window for a pair of spike parameters.
Windows / Parameter	Creates a new parameter window for a pair of stimulus parameters.
Tools / Settings	Shows the settings dialog (see chapter 3.2).
Tools / Adjust Spike-Length	Allows to change the spike length of a finished measurement result. After loading a result you can enter a different spike length here. After the change original data are rescanned automatically, all spikes are newly detected. Indiviual epoche thresholds are NOT cleared.
? / Help	Shows PDF manual.
? / About	Shows an information dialog.

Table 10: Main screen buttons

🗋	Same as menu item File / Load Measurement Template (see above)
🗐	Same as menu item File / Start Batch Mode (see above)
📂	Same as menu item File / Load Measurement Result (see above)
📂	Same as menu item File / Append To Measurement (see above)
💾	Same as menu item File / Save (see above)

	Same as menu item Windows / Cluster (see above)
	Same as menu item Windows / Parameter (see above)
▶	Starts current measurement
⏸	Pauses / unpauses current measurement
⏹	Stops current measurement
	Reloads recorded epoches.
	Reloads recorded epoches and repeats complete spike detection using the current threshold value(s). Individual epoche thresholds are cleared.
🔍	Starts the manual search mode (see chapter 3.5).
	Starts the free-search mode (see chapter 3.5).
Mode	The button right of the free-search button toggles between "Standard-Mode" and "In-Situ-Mode". It shows the current mode. The mode cannot be switched after loading a measurement (see chapter 4.3).

3.2 Settings

The settings of AudioSpike are adjusted in the "settings dialog" with multiple tabs. On the tab "General Settings" the following options are available:

Table 11: General Settings

Parameter	Description
Program settings/ Settings subdirectory	Here you can set the default settings subdirectory to be used for all AudioSpike settings, calibration values and filters. Using different settings is especially useful when using different hardware (settings) for different experiments on the same computer. If you select a new settings name here you have to restart AudioSpike for this change to be applied. You can create new settings variants, by creating a copy of the folder "default" in the "setttings" directory of your AudioSpike installation. Alternatively you can pass the settings name to be used on the command line (see chapter 5.1). The command line value superseeds the value selected here.
Program settings/ Allow Multiple Audiospike Instances	If this option is checked, then you may run multiple instances of AudioSpike. Otherwise each call to AudioSpike (double click or start command from MATLAB) will re-use an existing AudioSpike window. **Note:** do not start measurements from multiple AudioSpike windows at the same time: this may result in unpredictable audio stimulation and/or electroe recording signals! Use multiple windows only for evaluation purposes!

Program settings/ Always save additional MAT file	If this option is checked, AudioSpike saves every result additionally as MAT file for the direct import into MATLAB or Octave. The filename is the same as the result XML file with the extension ".mat".
Program settings/ Save probe microphone audio data (in-situ)	If this option is checked, AudioSpike saves the audio data from the probe microphones (in-situ-micophones) as well in file "probemics.pcm" in the result directory. **Note:** If you have specified a SamleRateDevider for downsampling the electrode recordings, thes downsampling is applied to the probe microphone recordings as well. For details on the format see 2.2.
Program settings/ Start in In-Situ-Mode	If this option is checked, AudioSpike always starts in In-Situ-Mode.
Program settings/ Free subwindow positions	If this option is checked, all subwindows (stimuli, epoches ...) can be placed freely on the desktop, otherwise they only can be moved within the main window.
Program settings/ Style	Here you can select different styles for the appearence of the software (applied after clicking "OK" in the settings. This might especially be useful when running AudioSpike with different configurations to distingish your different settings directly by different layout.
Directories/ Templatepath	Path were to load template XML files from. If you check the option "Auto" then the last used path is used as default and the fix selection is disabled.
Directories/ Resultpath	Root path were to store results
PSTH-Window/ Default binsize	Default binsize in milliseconds to use for PSTH window
Spikes/ Pre-Threshold (default)	Time in milliseconds to extract for each spike before threshold crossing. May be overwritten by measurement XML file.
Spikes/ Post-Threshold (default)	Time in milliseconds to wait AFTER threshold crossing, before next threshold crossing is interpreted as next spike. Please note, that the PostThreshold value determines the temporal resolution of successive spike-events. NOTE: if you are specifying a very short PostThreshold the same spike might be detected multiple times. If this value is 0 then the PostThreshold is calculated to PostThreshold = SpikeLength – PreThreshold. May be overwritten by measurement XML file.
Spikes/ Spike-Length (default)	Total length in milliseconds to extract/plot for each spike (Pre-Threshold + time from threshold crrossing to spike end). May be overwritten by measurement XML file.
Free Search Mode/ Stimulus Length	Length of stimulus in looped playback in free search mode in milliseconds
Free Search Mode/ Pre-Stimulus Length	Length of silence before stimulus in looped playback in free search mode in milliseconds
Free Search Mode/ Repetition Period	Repetition period for stimulus in free search mode in milliseconds
Free Search Mode/ Ramp Length	Length of hanning ramp in the beginning and at the end of the simulus in free search mode in milliseconds

The tab "Sound Settings" contains multiple settings to set up your audio system for the use with AudioSpike as well as access to calibration and equalisation interfaces. The available options depend on the current global mode AudioSpike is switched to ("Standard-Mode" or "In-Situ-Mode").

Table 12: Sound Settings

Parameter	**Description**
System/ Driver	Select the ASIO sound driver (sound card) to be used from the drop down list.
System/ Samplerate (default)	Here you may enter the samplerate to be used. This samplerate will be used as default samplerate (if no other samplerate is specified in a measurement template) and for calibration and internal equalisation. It is recommended to enter the highest used samplerate here. The selected sound card must support the specified samplerate.
System/ Downsampling (default)	Here you may enter the a factor for downsampling of recorded data. This factor is applied before extracting/storing the spikes
Special Audio Channels/ Trigger Output	Audio channel to be used for trigger output. This channel must be connected with a shortcut cable to the trigger input channel. The selected channel cannot be used for signal output below. If you click the maginfiying glass right to the selection a trigger test is started to check the wiring of your trigger channels: a dialog is displayed showing the names of the used trigger channels, and triggers are played and recorded. Two counters show the number of played and detected triggers (should increase together, if wiring is fine). Additionally the value of the last detected trigger (between 0 and 1) is displayed. It should not be smaller than 0.3, otherwise check your wiring and/or mixer settings. **NOTE:** Depending on internal delay of a soundcard there might be a lag between the counters, but they must increase their values in parallel.
Special Audio Channels/ Trigger Input	Audio channel to be used for trigger input. This channel must be connected with a shortcut cable to the trigger output channel. The selected channel cannot be used for signal input below.
Special Audio Channels/ Trigger Offset	The offset that is introduced between the playback of the trigger and the playback of a stimulus (in milliseconds). This value may be used to compensate for a delay of a user defined equalisation impulse response (see chapter 4.2.3) or for an acoustical or electrical delay in your setup.
Special Audio Channels/ Monitor Output	Audio channel use for monitoring (listen) to inputs. This setting is optional. If you select a channel here you can 'map' any eletrode input to this audio channel during measurements to listen to electrode responses. The selected channel cannot be used for signal output below. You can clear this setting by clicking the red 'x' next to the drop down list.
Special Audio Channels/ Reference Microphone	Here you can select an input channel for connecting a reference microphone to be used for "Spectral Equalisation" or "in-situ calibration" respectively , see below. For "Standard-Mode" this selection is optional, for "In-Situ-Mode" it is mandatory. You can clear this setting by clicking the red 'x' next to the drop down list. If the option "Use acoustic calibrator and reference microphone for level calibration" is checked, then the reference microphone is used for level calibratio (see chapter 4.2.3). In this case an acoustic calibrator is required for the calibration procedure, but no level meter.
Equalisation Method	Here you can select two different methods for the (optional) equalisation (spectral compensation) of your sound ouput channels. Please refer to chapter 4.2 for details. This section is only available in "Standard-Mode".

Sound Output	Select one ore more output channels to be used for acoustical stimulus playback. All available sound carf channels are displayed, the trigger and monitor channel selected above cannot be selected for stimulus output. The column “Index” shows the numbered indices of the channels (ascending numbers, starting with 1). The channels can be addressed in your measurement templates using these internal indices. The next two columns contain information about the calibration and equalisation or loaded impulse resonses respectively. Clicking with the right mouse button on a line invokes a context menu with commands for calibration and equalisation (refer to chapter 4 below) and for setting and removing a highpass filter for the output (see chapter 3.2.1). The edge frequency is displayed in column “Hi-Pass”. Additionally you can select an output to be a “raw output” channel. A “raw output” channel does not need any calibration: the signal(s) on that channel are played as they are without applying any level (calibration and /or stimulus level). This section is only available in “Standard-Mode”.
In-Situ Output/Input	Select one ore more output channels to be used for acoustical stimulus playback and corresponding probe microphone input for measurements with in-situ calibration. All available sound carf channels are displayed, the trigger and monitor channel selected above cannot be selected for stimulus output. The column “Index” shows the numbered indices of the channels (ascending numbers, starting with 1). The channels can be addressed in your measurement templates using these internal indices. Clicking with the right mouse button on a line invokes a context menu with commands for selecting/removing the input channel for the probe microphone (in-situ microphone), for perfoming/removing the calibration of the probe microphone (refer to chapter 4 below) and for setting and removing a highpass filter for the output (see chapter 3.2.1). The edge frequency is displayed in column “Hi-Pass”. If a selected probe-mirophone is already calibrated the column “Calibrated” displays the number “1”. Additionally you can select an output to be a “raw output” channel. A “raw output” channel does not require a probe microphone to be selected and does not need any calibration: the signal(s) on that channel are played as they are without applying any level (calibration and /or stimulus level). An in-situ caloibration for that output channel is never performed during a measurement. This section is only available in “In-Situ-Mode”.
Electrode Input	Select one ore more input channels to be used for spike recording. All available sound carf channels are displayed, the trigger input channel selected above cannot be selected for spike detection. The column “Index” shows the numbered indices of the channels (ascending numbers, starting with 1). The channels can be addressed in your measurement templates using these internal indices. Clicking with the right mouse button on a line invokes a context menu with commands setting and removing a bandpass filter on the input (see chapter 3.2.1). The edge frequencies are displayed in column “Band-Pass”.

3.2.1 Input and Output Filters

In the settings dialog you may specify a highpass filter for each output and a bandpass filter for each input.

Output Highpass Filters
A highpass is specified by setting the the edge frequency for the filter. At the edge frequency the filter's attenuation is still 0 dB, the ramp starts below the frequency. The filter steepness is 12 dB/octave. **Note**: this filter is ALWAYS activated (calibration, spectral calibration, (free) search and measurement. You have to take care not to use stimuli below the highpass frequency. A corresponding warning is displayed on measurement start.

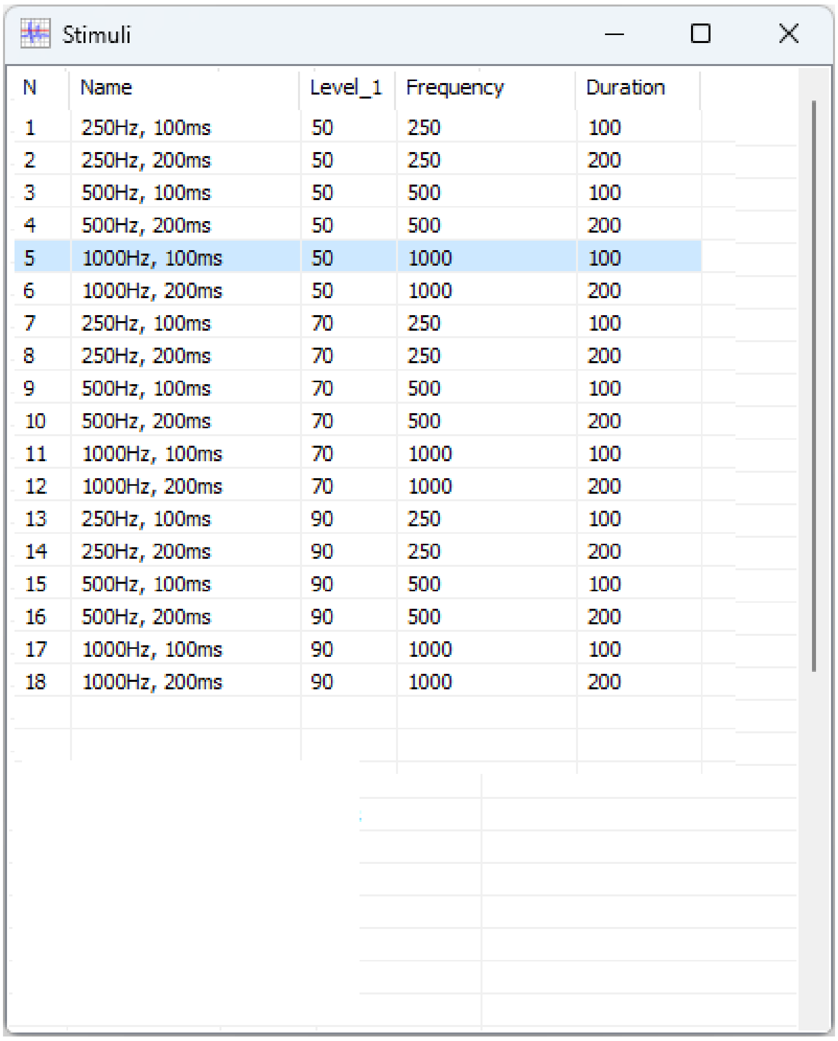
Input Bandpass Filters
A bandpass is specified by setting the two edge frequencies for the filter. At the edge frequencies the filter's attenuation is still 0 dB, the ramps start below the lower and above the higher frequency. The filter steepness is 12 dB/octave on both sides. **Note**: this bandpass filter is applied AFTER the analog signal is A/D converted. Thus it cannot compensate analog overdrive on the input! Additionally you have to be aware that this bandpass filter may affect the shape of your detected spikes, if the edge frequencies are not selected properly!

3.3 Subscreens

The following chapters describe the available windows/dialogs for displaying and selecting/analysing data. Many of the dialogs contain X-Y-plots. In most plots you can click on both axis to adjust the view (minimum and maximum of the axis).

3.3.1 The stimulus window

The "Stimulus window" shows a list of all configured stimuli of the current measurement. Figure 1 shows an example for a measurement template using three parameters (level, frequency and duration). The first column shows a running index, the second the stimulus name (as set in measurement template). The following columns show all parameters, where "Level_?" denotes level on channel with number ?. During a measurement the currently played stimulus is highlighted. In search mode you can select the stimulus to play by clicking it in the stimulus list (see chapter 3.5).

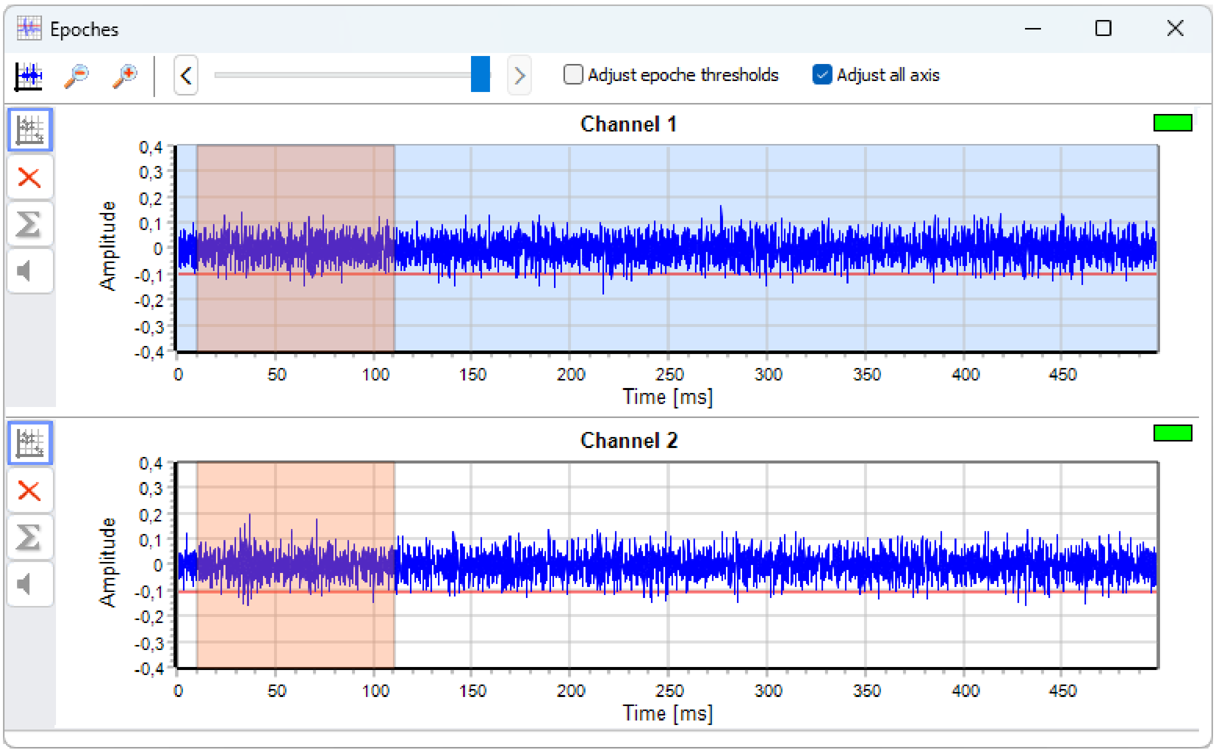


N	Name	Level_1	Frequency	Duration
1	250Hz, 100ms	50	250	100
2	250Hz, 200ms	50	250	200
3	500Hz, 100ms	50	500	100
4	500Hz, 200ms	50	500	200
5	1000Hz, 100ms	50	1000	100
6	1000Hz, 200ms	50	1000	200
7	250Hz, 100ms	70	250	100
8	250Hz, 200ms	70	250	200
9	500Hz, 100ms	70	500	100
10	500Hz, 200ms	70	500	200
11	1000Hz, 100ms	70	1000	100
12	1000Hz, 200ms	70	1000	200
13	250Hz, 100ms	90	250	100
14	250Hz, 200ms	90	250	200
15	500Hz, 100ms	90	500	100
16	500Hz, 200ms	90	500	200
17	1000Hz, 100ms	90	1000	100
18	1000Hz, 200ms	90	1000	200

Figure 1: The stimulus windows

3.3.2 The epoche window

The epoche window shows a plot of epoches (spike trains) for each used inut channel (electrode).
The window contains a toolbar at the top with the following buttons/controls:



☐ Adjust epoche thresholds ☑ Adjust all axis

Figure 2: The epoche windows

	Shows a dialog for selecting the electrode windows (channels) to be displayed.
	Decreases the height of the single electrode windows.
	Increases the height of the single electrode windows.
Scroll-bar	Use the scrollbar to scroll through recorded and loaded epoches. This scrollbar is only enabled, if the measurement is not running and epoches are loaded or rescanned respectively. The stimulus that corresponds to the currently displayed epoche is highlighted in the stimulus list.
☑	“Adjust epoche thresholds”. If this option is selected, then changing the spike detection threshold is only applied to the currently displayed epoche and the spike detection for this epoche is repeated automatically. It is recommended to select the option “Plot current epoche only” on the spikes window as well when adjusting individual epoche thresholds (see chapter 3.3.6). This option is only available post-hoc (after a measurement is stopped).
☑	“Adjust all axis”. If this option is selected, then changing minimum or maximum of X-axis or Y-axis respectively is applied to all electrode windows.

Each electrode window contains a vertcal toolbar with the following buttons:

	Selectets the currently active channel (electrode) for analysis (cluster, PSTH and parameter selection). The background of the plot of the active electrode is coloured light blue. The index of the currently selected channel is displayed in the title bar of all evaluation windows (cluster, spike, PSTH and parameter windows).

✕	Clears averaging buffer (see below)
Σ	Activates avareaging mode for this electrode. The plotted epoches are summed and averaged. Clikcing the button again switches back to regular mode (latest epoche is plotted).
🔊	Selectes the corresponding channel/electrode for monitor output. If a monitor output channel is configured input signal if the selected input channel is copied to the monitor output channel for acoustical monitoring purposes. The speaker button of the currently selected monitor channel is highlighted.

In the upper right corner a small LED indicates clipping for each input channel. If an overdrive is detected on the soundcard input the (A/D converter) of corresponding channel, the LED flickers red.

Important note: this overdrive is detected directly on the input: such an overdrive cannot be compensated by a bandpass filter on the input channel (see chapter 3.2), because the analog input signal causes the overdrive in the A/D converter!

The position in time of the current stimulus is indicated by a light red range within the electrode signal plots. The threshold value for a channel is displayed by a horizontal red line. You can adjust a theshold by performing a left mouse (or dragging) while pressing the Shift-key on the keyboard. Pressing thje Ctrl-key and performing a left mouse click shows a dialog for entering the threshold value manually. Threshold values are stored internally and are used as default (see note below) until they are changed again.

After changing threshold values you may rescan all epoches in stopped/saved measurements. All detected spikes are cleared and the spike detection is repeated using the new threshold value(s).

Important note: if threshold values are specified in a measurement template or a measurement result respectively, then these values are used and superseed values set in the GUI.

3.3.3 Cluster windows

A “cluster window” shows an X-Y-plot for two “spike parameters”. For each detected spike a dot is displayed in this plot.

Available spike parameters are:

Table 15: Spike parameters

Paramater	**Abbreviation**	**Description**
Total amplitude	TotalAmp	Total action potential amplitude.
Peak 1	Peak1	amplitude of the 1st component (“phase”).
Peak 2	Peak2	amplitude of the 2nd component (“phase”).
+ve Peak	PeakPos	amplitude of the positive peak.
-ve Peak	PeakNeg	amplitude of the negative peak.
Peak to Peak	PeakToPeak	time in μsec between the 1st and 2nd component
Trig to Peak	TrigToPeak	time in μsec between the “trigger” (threshold crossing) and 2nd component

Note: spike amplitudes are always scaled to rel. fullscale, i.e. to a range between -1 and 1 with respect to the full dynamic range of the sound card.

Figure 3 shows an example for a Peak1-Peak2-plot. Within a clusetr plot you can define cluster multiple cluster selections by drawing the mouse while pressing the left mouse button. A right mouse button click within a selection rectangle removes it. Pressing the button with the red “x” removes all selections.

You can define one or more cluster windows using the corresponding menu item or button respectively. A selection of spike parameters for X and Y axis is displayed. If you have opened more than one cluster window, then only one can be active for defining selections. In currently active window the selection button

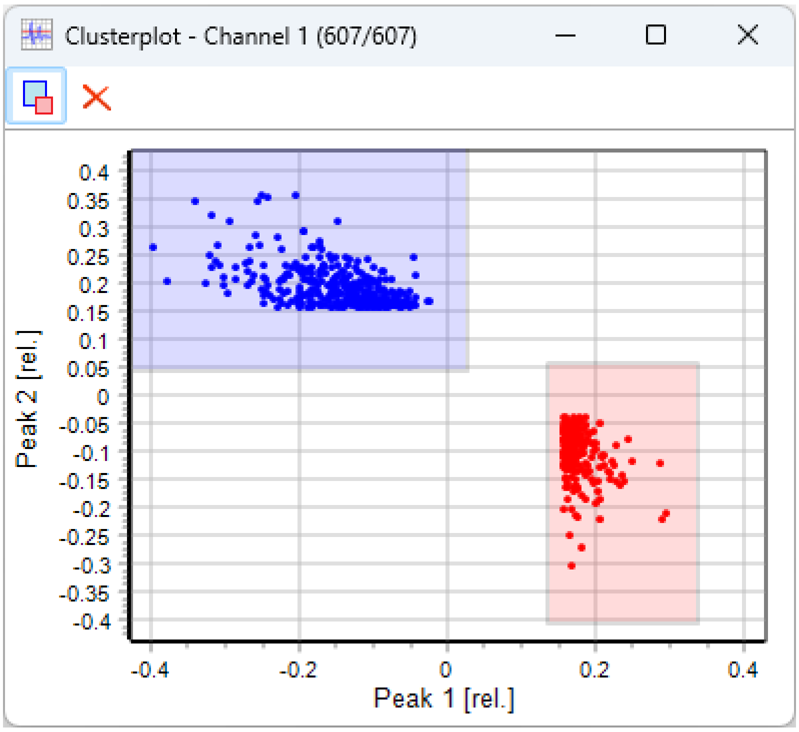


Figure 3: The cluster windows

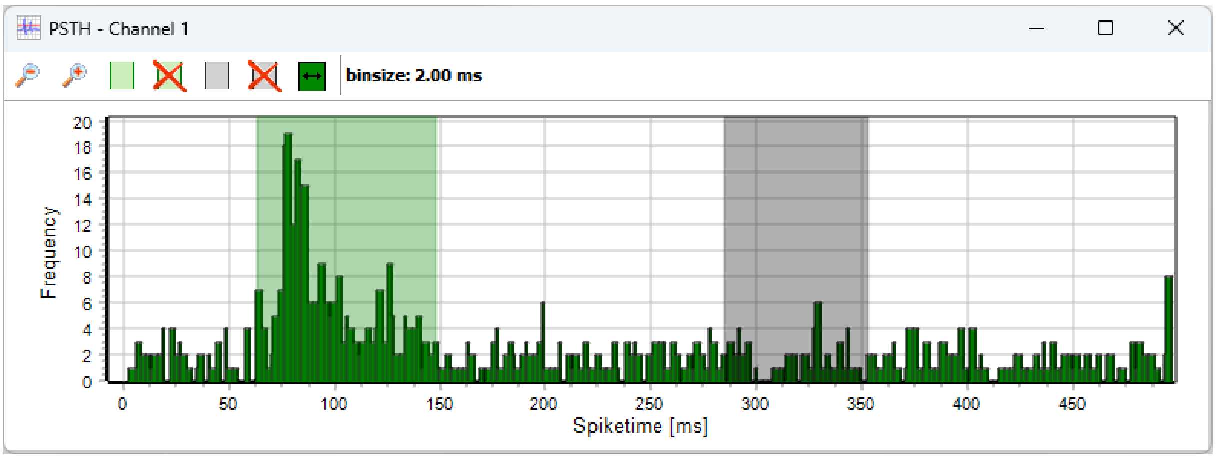
(top left) with the two selection rectangles is coloured, all other cluster windows show a grayed button. Click a grayed button to activate the corresponding cluster window for selections.
Different selections have different colours. The dots for the corresponding spikes and the spikes in the spike window are coloured in all cluster windows with the corresponding colour. The title of the cluster window shows the currently used channel (electrode) and the number of selected spikes (in brackets: selected/total).
Selections can only be displayed and/or changed for one recording (electrode) channel at the same time. The cluster, spike, PSTH and parameter windows always display the spikes of the channel selected in the epoche window only. To do selections/analyisi for different channels you may select the channels subsequently.

3.3.4 The PSTH windows

The PSTH window (figure 4) displays the Peri-Stimulus Time Histogram for the spikes currently selected in the active cluster window.
The window contains the following buttons:

🔎	Decreases the width of the histogram bars
🔎	Increases the width of the histogram bars
🟩	Shows a dialog to manually enter minimum and maximum spiketime for the signal (green) selection. With the signal selection you define the range of spike times for spikes to be used un the parameter plots. Alternatively you can “draw” a selection by dragging the mouse in the chart while pressing the left mouse button
❌	Clears the signal (green) selection

▯	Shows a dialog to manually enter minimum and maximum spiketime for the noise (gray) selection. With the optional noise selection you may define a noise ground. When calculating bubble sizes (number of occurrances) in the parameter windows the sum of all spikes within this selection is subtracted from each bubble in the plots. Alternatively you can "draw" a selection by dragging the mouse in the chart while pressing the right muse button
☒	Clears the noise (gray) selection
↔	opens dialog to set the binsize for the PSTH plot.



binsize: 2.00 ms

Figure 4: The PSTH windows

These selections are used to create the bubble plots in the parameter windows. **Note:** if no selection is defined in the PSTH windows, no bubbles will be displayed in the parameter windows.
The "Stimulus-PSTH" plot displays one separate PSTH for each configured stimulus. This plot is intended for displaying responses per stimulus only, no selection controls are available. The binsize can be adjusted as in the standard PSTH window.

3.3.5 Parameter windows

A "parameter window" shows an X-Y-bubble-plot for two "stimulus parameters". When creating a parameter window using the corresponding menu item or tool button a selection of available stimulu parameters is displayed for the X and Y axis. For the Y axis the additional selection "Response" is available. If you select this item, then the response "strength" as a function of the parameter on the X- axis.
You can define multiple parameter windows (if more than two stimulus parameters are available).
If you have selected "Response" for the Y axis the plot is a simple 2-d plot, see figure 5.
An two-parameter example is displayed in figure 6. The bubble plot shows an X-Y-plot of the selected parameters. The size of a bubble indicates the number of spikes within the selected spikes for this parameter combination. The bubble size is calculated from all spikes selected in the PSTH window. If a noise selcetion is specified there as well, the the sum of all spikes within the noise selection is subtracted from each bubble. Therfore negative values may occur, indicated by white bubbles rather than green. The title of the chart contains the total number of spikes in the largest bubble and the correspondiung spike rate. The second line of the title contains the same values devided by the number of repetitions.
If you click a bubble, then an additional "Bubble-PSTH" window is displayed showing a PSTH only for spikes contained in the clicked bubble. The binsize can be adjusted as in the standard PSTH window (see figure 7).

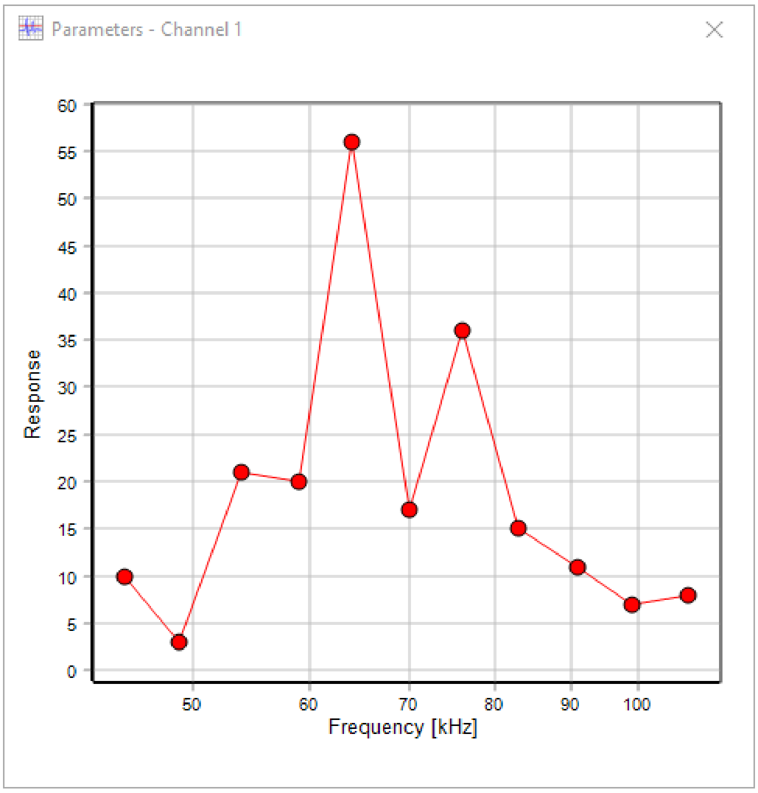


Figure 5: The parameter window with “Response” on the Y axis

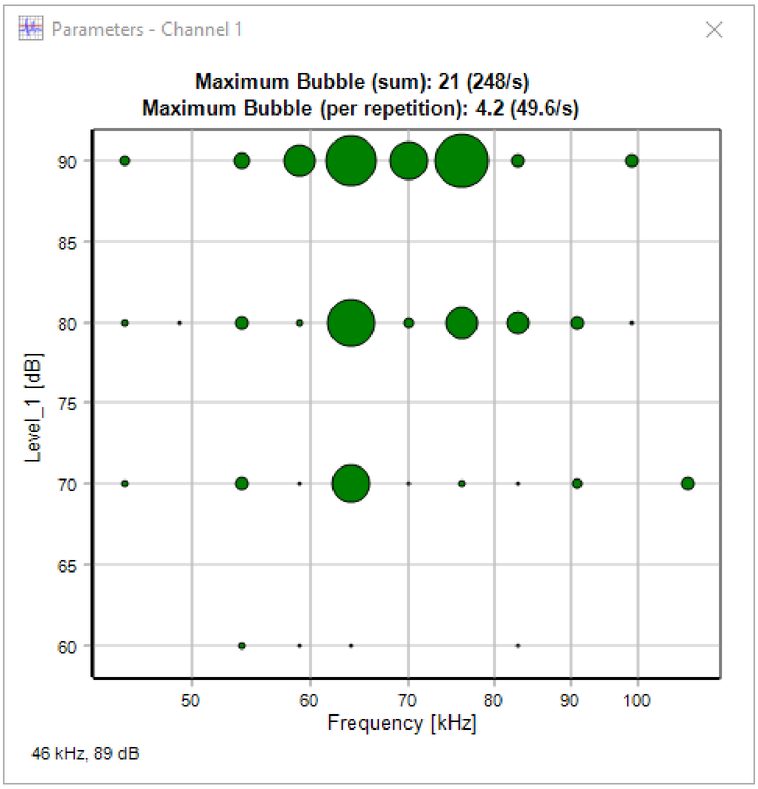


Figure 6: The parameter window

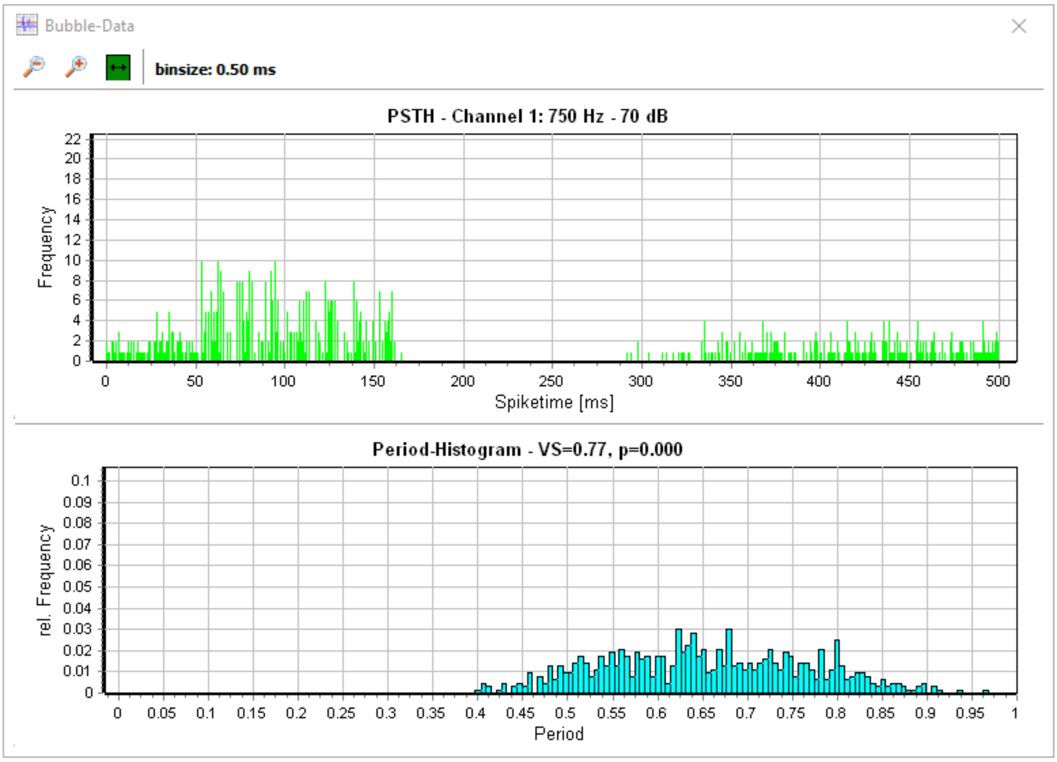


Figure 7: The bubble PSTH window

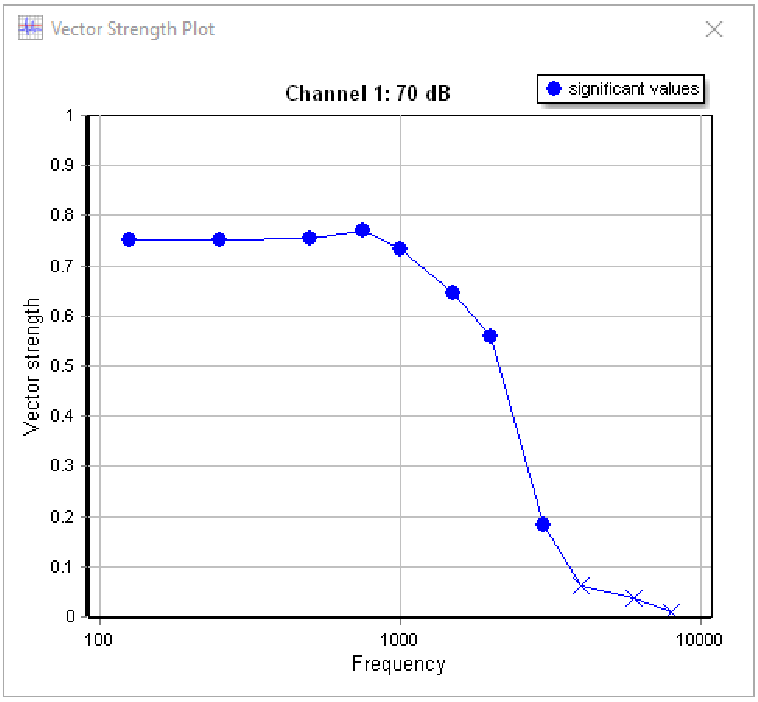


Figure 8: The vector strength window

If one of your parameters was “Frequency” then an additional period histogram is displayyed below the PSTH, where the title contains the vector strength (VS) and the probabilty, that the periods are ditributed uniformly (p). Additionally a vector strength plot is displayed where significant values ($p < 0.05$) are marked (see figure 8).

3.3.6 The spike windows

The spike window plots each spike that is currently selected in the active cluster window. The colours of the spikes correspond to the selections in the cluster windows. Additionally the current threshold is displayed as a horizontal read line. Pressing the small buttons with the rounded green arrows updates the plot. For performace reasons a maximum of 2000 spikes is plotted. You can scroll through all selected spikes using the arrow buttons at the top. If you check the option “Plot current epoche only” then only spikes of the epohce currently active in epoche window are displayed. This option is only available, when epoches were reloaded before.

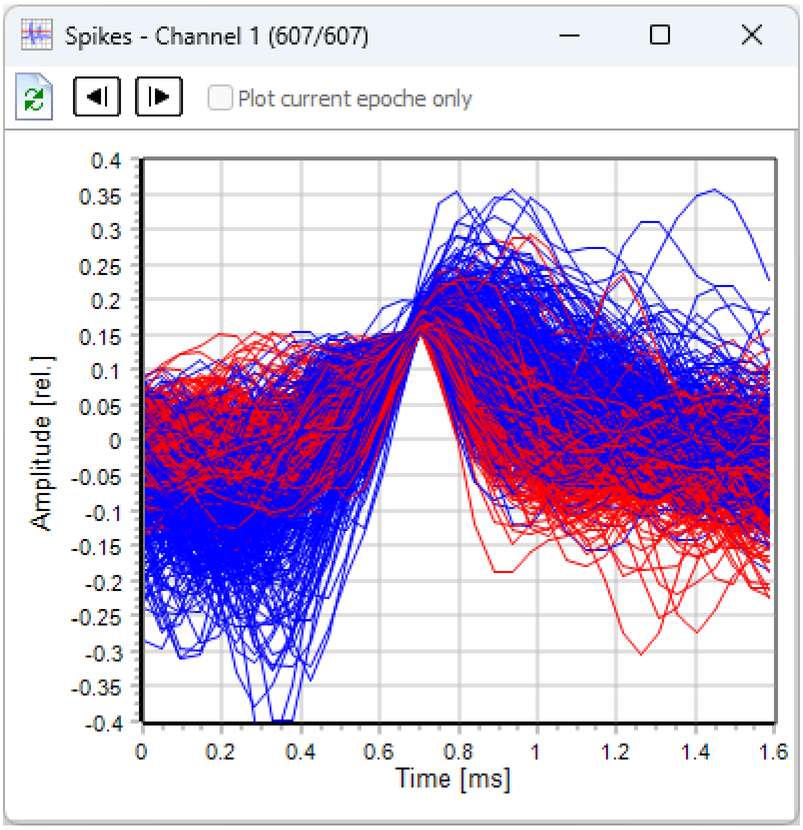
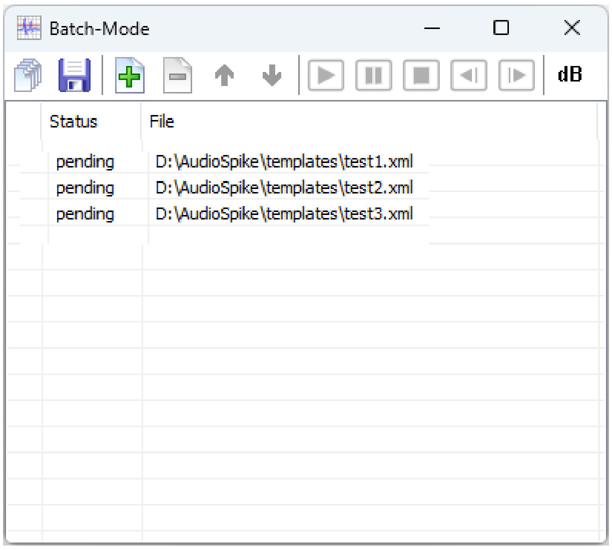


Figure 9: The spike windows

Note: during a measurement and in free search mode the latest 2000 detected spikes are plotted. After stopping you can scroll through all spikes as descripbed above.

3.4 The batch mode

In batch mode you can run a list of measurement templates subsequently in an automated way. In batch mode the standard controls for start/stop/pause on the main screen auf AudioSpike are disabled.



Status	File
pending	D:\AudioSpike\templates\test1.xml
pending	D:\AudioSpike\templates\test2.xml
pending	D:\AudioSpike\templates\test3.xml

Figure 10: The batch mode window

To edit your batch files (i.e. measurement template collections) you may use the following buttons:

🗐	Load a batch file
💾	Save a batch file
➕	Add item(s). You may drag-and-drop files to the window as well.
➖	Remove item(s)
🡅	Move item up
🡇	Move item down

A batch run is controlled by the following buttons:

▶	(Re)-starts batch run
⏸	Pauses / unpauses current measurement
■	Button stays pressed and can be released again: batch will stop and wait after current measurement is complete if the button is pressed.
◀|	Reject data of current measurement and wait. After restart the same measurement is repeated.

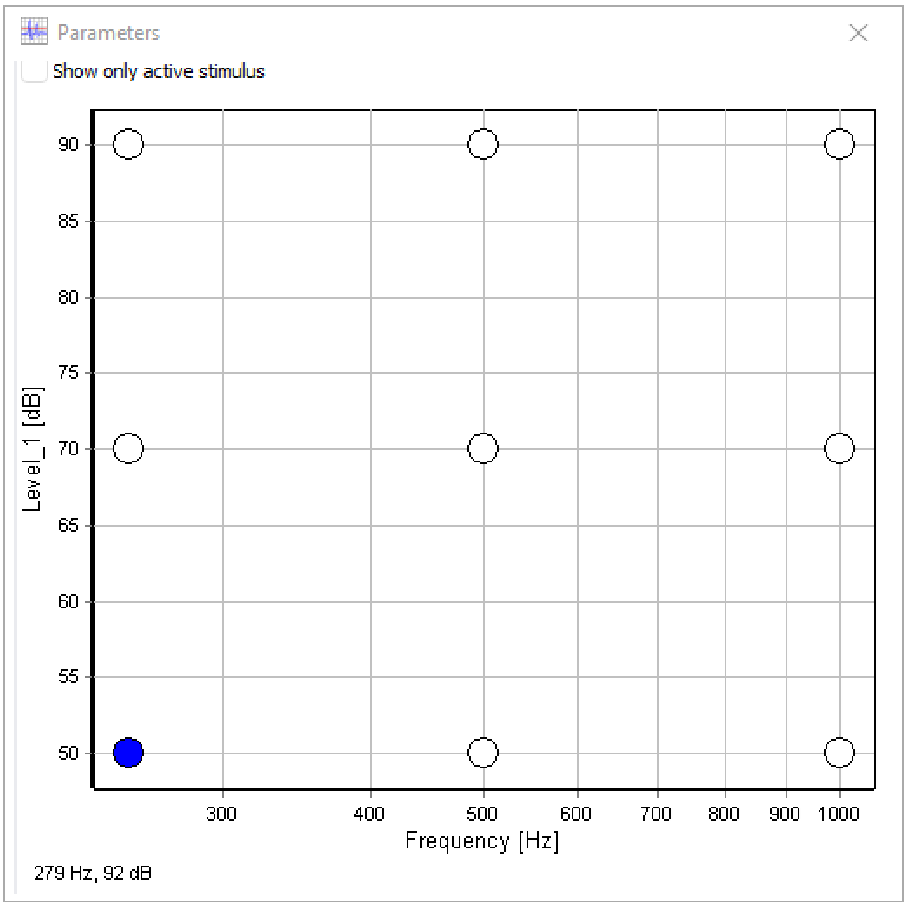
|▶	Store data of current measurement in current status and wait. After restart the next measurement is started.
dB	Button stays pressed and can be released again: for in-situ measurements the in-situ calibration will be repeated before next measurement if the button is pressed. Note: an insitu-calibration is only performed before the first measurement of a batch run. If the used output channels are differ between two subsequent measurements within a batch, then the in-situ calibration is performed automatically before a measurement again.

3.5 The manual search mode

After loading a measurement template you can start the measurement in the so called “manual search mode” by clicking the button with the magnifying glass. You will be queried to select two of your stimulus parameters for an X-Y-plot. After clicking Ok a corresponding plot will be displayed contaning a circle for each combination of the selected parameters (see figure 11). The playback of the first stimulus starts in an endless loop.

If you click into the chart playback continues with the stimulus next to the clicked coordinates. If you check the option “Show only active stimulus”, then only the currently played stimulus is displayed as blue bubble (no other “bubbles” displayed). Alternatively you can select a stimulus by clicking it in the simulus list window.

Clicking the “stop” button stops the manual search mode.


Parameters

□ Show only active stimulus

279 Hz, 92 dB

Figure 11: Manual search

3.6 The free search mode

The “free search mode” can be entered by clicking the button with the red sine and the magnifying glass. A separate window is displayed (figure 12).

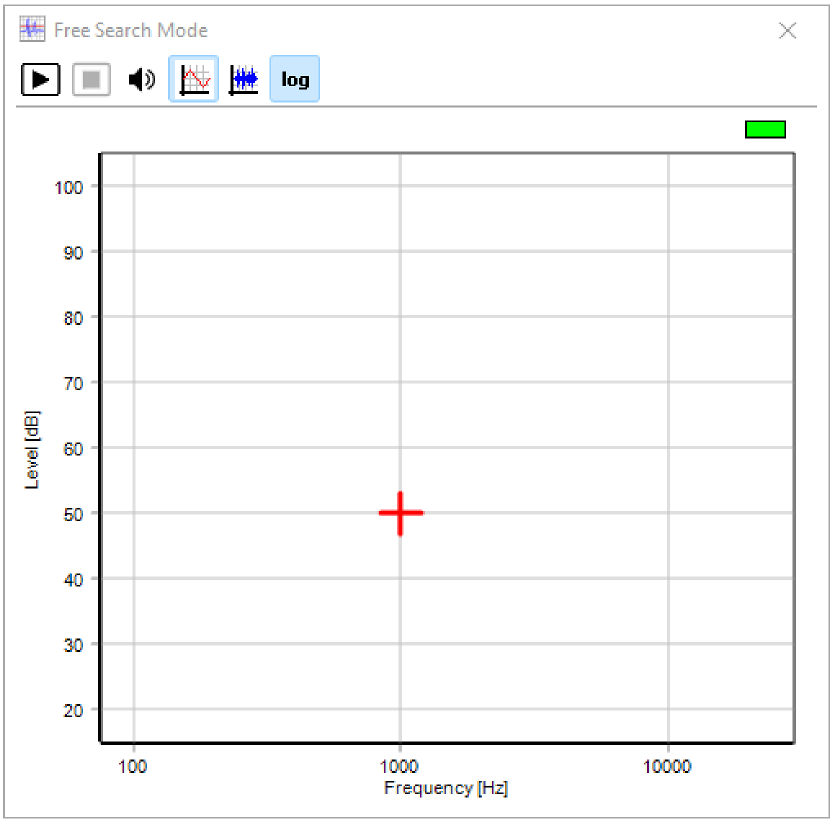


Figure 12: Free search

The window contains the following buttons:

▶	Starts free search
■	Stops free search
🔊	Shows a dialog to select the output channels to be used. All channels selected as output channels in the settings dialog are listed.
📈	Selects sine signals for output. You can click into the chart and a sine with the corresponding frequency and level is played.
📊	Selects a Schroeder-phase tone complex for output. The edge frequencies of this tone complex depend on frequency borders selected during the spetral equalisation or in-situ calibration respectively (see 4). You can click into the chart and the noise is played at the corresponding level.
log	This button toggles lin/log display of the frequency axis.

Closing the window exits the free search mode.

4 Calibration and Equalisation of Loudspeakers and Microphones

AudioSpike offers versatile methods for calibration and spectral equalisation (frequency response compensation) of loudspeakers and in-situ microphones.
Note: to understand how AudioSpike calculates output levels it is strongly recommended to read the following paragraph thoroughly!

The calibration process in AudioSpike is based on the following concept:

– Each used loudspeaker/transducer output is equalised to compensate for the technical frequency response of the used transducer/speaker using digital filters.

– Afterwards this equalisation is always applied to the corresponding output channel (during level calibration, (free) search and measurement). Thus AudioSpike can assume, that e.g. sine signals at different frequencies will always have the same level, if the signal amplitude is identical.

– During calibration of the absolute level a broadband noise is played and the user has to adjust the noise level in a special calibration procedure to a particluar level (to be measured with a levelmeter) by changing the digital attenuation of the signal .

– After this adjustment AudioSpike "knows" which digital attenuation corresponds to which sound pressure level. This so called "calibration value" is stored and used later on to calculate the digital attenuation to apply to a signal to play it with a requested level.

  The different steps of this procedure are perfomed differently for "Standard-Mode" and "In-Situ-Mode", but the basic concept remains identical (see below). Calibration and equalisation is described separately for the two AudioSpike-modes in the next paragraphs.

Important notes:

– For this calibration concept it is absolutely necessary that the equalisation of the loudspeaker/transducer has to be done properly. The "error" arising from a missing/inaccurate equalisation strongly depends on the resulting frequency response of the transducer: the less flat it is, the larger the mismatch will be.

– Furthermore AudioSpike has to know something about the "intrinsic" level/energy of the particular stimulus to play, please read the following paragraph about the "RMS problem" thoroughly.

– Usually sound transducer show non-linearities especially for broadband signals at higher levels. Thus the measured output level for such stimuli may deviate from the calibrated level by a few dB!

4.1 RMS values

Audio signals to be used are passed to AudioSpike by filename in the measurement template – AudioSpike does not know anything about the "nature" or the recording level of the signal. If AudioSpike would simply play a stimulus "as it is" (applying the calibrated attenuation for a particular level described above), then the real output level would depend on the recording level of the signal within the audio file. E.g. a sine signal with a peak in the file of 0.5 would be 3.01 dB softer than a full driven sine with a peak of 1.
Therefore AudioSpike compensates for the intrinsic signal energy by calculating it's RMS value and compensate the level by this value. In the example of the two sine signals this would work perfectly, since the RMS value of the softer sine will exatcly be 3.01 dB lower.
But this "automatic RMS compensation" will not always be what you want to happen – or what you would expect.
The following example tries to demonstrate the problem: if you e.g. want to set up a measurement to compare the influence of a pulsed signal versus a stationary signal, then you may set up a sine signal and

a pulsed sine signal containing pauses. In your comparison experiment you might want to play both sine signals at the "same level", where "same level" is expected to hold for the sine parts of both signals: for the pulsed signal those parts of the signal containing the sineshould have the same level as the stationary signal. But if you would simply calculate the signal energy (RMS value) of both signals, then the pauses in the pulsed signal will contribute to the RMS and the value will be lower than of the stationary signal.
When compensating the levels using the pure calculated RMS, this would lead to higher (sine–)levels for the pulsed signal.
In this simple example this could be solved by using peak values for compensation rather than RMS, but for many other signals, the peak is not a good estimatior for a signal's energy.
In general, only the user who generated the signals knows, which parts of the signal (peak, RMS, plateu, signal parts without pauses...) are the best "estimator" for the energy of a stimulus. For this reason you can specify an RMS value for each stimulus in a measurement template. This way you can "scale" your stimuli among each other.
Important note: this RMS value is optional. If you do not specify RMS values, then AudioSpike calculates the RMS from the pure signal and compensates differences in RMS values. This means that simply scaling your stimuli (audio files) themselves will **not** result in different output levels: the two differently scaled sines from the example above will be played at the same level.

4.2 Calibration and Equalisation in Standard-Mode

In "Standard-Mode" AudioSpike offers two different methods for the equalisation of the sound output of the used combinations of sound card channels and loudspeakers.

4.2.1 Spectral Equalisation

The equalisation method "Spectral Equalisation" uses a built-in overlapped-add FFT filter to apply a frequency response to the sound output. You can select the used FFT length in the drop down list at the right. The resulting resolution (bin-size) of the filter can be calculated by

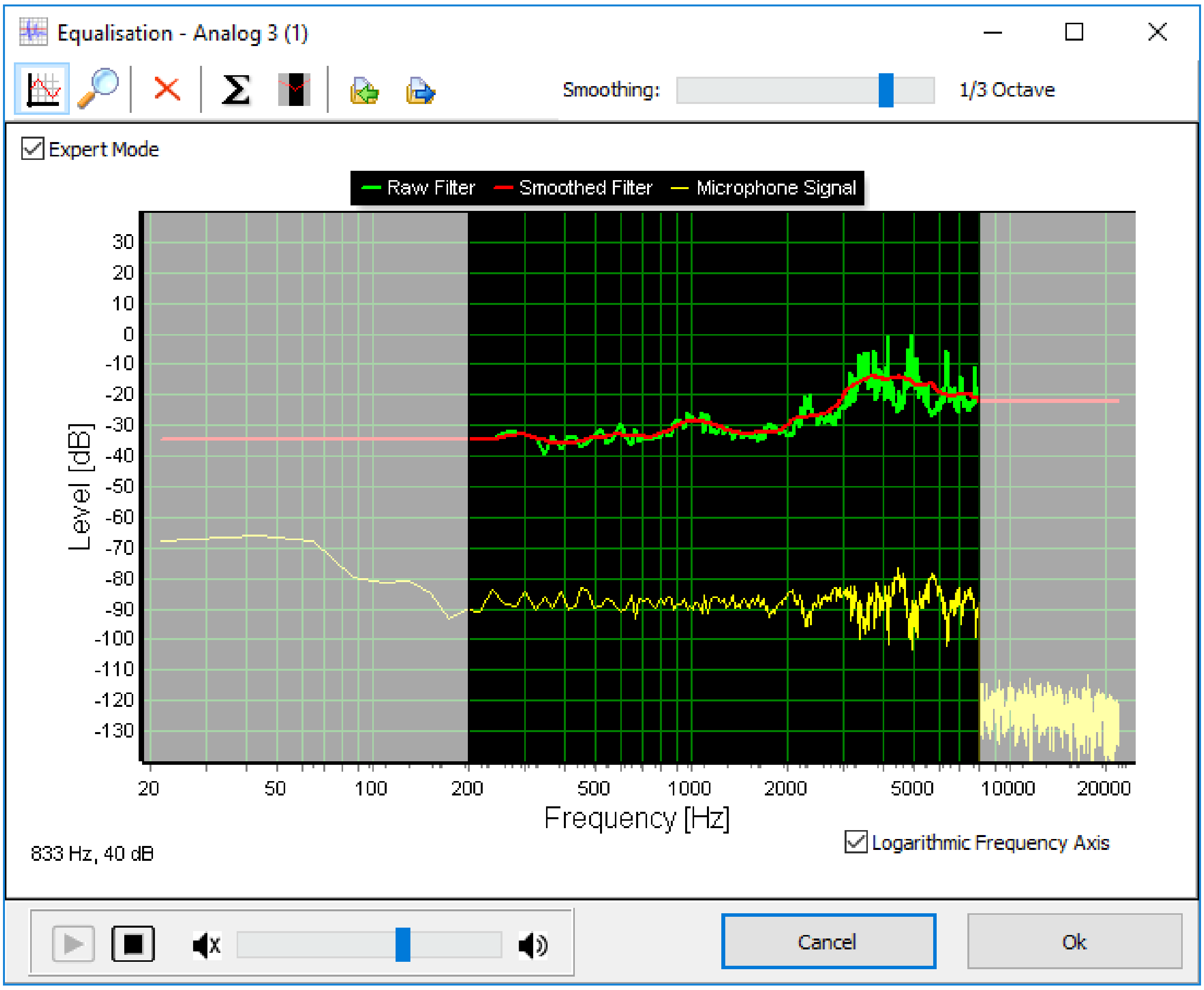
$$\delta f = \frac{s}{l}$$

where s is the samplerate and l is the used FFT length.
The frequency response can be adjusted with a graphical user interface: perform a right mouse button click on the corresponding channel in the list of output channels and select "Edit Spectral Equalisation" from the context menu. The equalisation dialog is displayed (see figure 13).
The current filter is displayed in a level-frequency plot as a red line. The frequency axis can be plotted linear or logarithmic (toggled with the checkbox at the bottm right of the chart). If a microphone input channel is configured in the sound settings, the current microphone signal is plotted as a yellow line.
The toolbar contains the following buttons:

Table 20: Equalisation dialog buttons

📈	When this button is pressed you can "draw" a filter in the chart by pressing the left mouse button and moving the mouse (toggles zoom button).
🔍	When this button is pressed you can zoom into the chart by pressing the left mouse button and drawing a zoom rectangle. Clicking the right mouse button or drawing a zoom rectangle from right to left will un-zoom again (toggles draw button).
✗	Clears the current filter. After clearing and pressing Ok an existing filter for a channel is removed.

Σ	Starts recording, averaging and generating the filter automatically (see below). This button is only enabled if the playback is currently running and a microphone channel is configured in the sound settings.
	Opens a dialog for adjusting the minimum and maximum frequency for calculating and smoothing a filter (grayed areas). **Note:** if you have configured a highpass filter frequency for this channel, the minimum frequency is adjusted to this highpass frequency at lest after clicking OK.
	Imports a filter from a file (see below)
	Exports current filter to a file (see below)


Equalisation - Analog 3 (1)

Smoothing: 1/3 Octave

☑ Expert Mode

833 Hz, 40 dB

☑ Logarithmic Frequency Axis

Figure 13: The equalisation dialog

At the bottom left a playback, a stop button and a volume slider are displayed. After pressing the start button, a Schroeder-phase tone complex containing only the frequencies between selected minimum and maximum frequency is played on the output channel selected when invoking the equalisation dialog (displayed in the caption of the dialog). If you have configured a microphone channel in the settings dialog spectrum of the current microphone signal is displayed in the chart as a yellow line.

A filter can be “drawn” with the mouse into the chart while pressing the left mouse button. The current mouse position (frequency and level) are displayed on the lower left of the chart. After releasing the mouse button the filter will be applied immediately.

Alternatively you can use the $\sum$–button (only enabled with configured microphone channel) to calculate the filter automatically. The input signal will be averaged for 10 seconds and compensation filter will be calculated from the division of the output by the input signal.
After drawing or calculating a filter automatically you can smooth the filter using the slider in the upper toolbar. When smoothing is applied the original (drawn or calculated) filter (green line) is smoothed. In linear viewthe smoothing is applied by averaging the magnitude in each FFT bin with it's neighbouring bins. The number of bins and the corresponding spectral width is displayed at the right of the slider. In log view smoothing is done by averaging the magnitude in FFT bin's within fractions of an octave.
Note: when the smooting slider is at the very left (no smoothing) the smoothed filter and the raw filter are identical, thus only the red line is visible.
A filter can be exported to and imported from files using the corresponding buttons. This might be usefule to copy a filter to another output channel or to generate a filter for a known frequency response (e.g. from the loudspeaker's techincal specifications). The file format is described in the appendix in chapter 5.3.
Press OK to store and apply the filter. If you want to remove an existing filter, click the clear button (red X) and click OK.
Important note: if you have changed the frequency response for a channel be sure to repeat the calibration afterwards!

4.2.2 Impulse Response

If you select the equalisation method "Impulse Response" in the settings, you can load a user defined impulse response (wave file) to each output channel. The sound output will be convoluted with the specified impulse response during calibration, free search and measurement. You may record your own user defined impulse responses and load them directkly into AudioSpike. To load an impulse response to an output channel, perform a right mouse button click on the corresponding channel in the list of output channels and select "Edit Spectral Equalisation" from the context menu.
The following restrictions for impulse responses apply (otherwise a corresponding error message will be displayed):

– The sample rate of an impulse response must match the sampling rate used for sound ouput (calibration, free search and measurement).

– If you use multiple output channels, the length in samples of all used impulse responses must be identical

Additionally you have to keep in mind, that an impulse response causes a delay. This should be taken into account when calculating spike times. Thus you have to enter the delay of your impulse response into the "Trigger Offset" field in the settings dialog (or add it to the already entered acoustical delay).
If you are using linear–phase impulse responses, then the delay can be calculated by

$$\delta t = \frac{N-1}{2s}$$

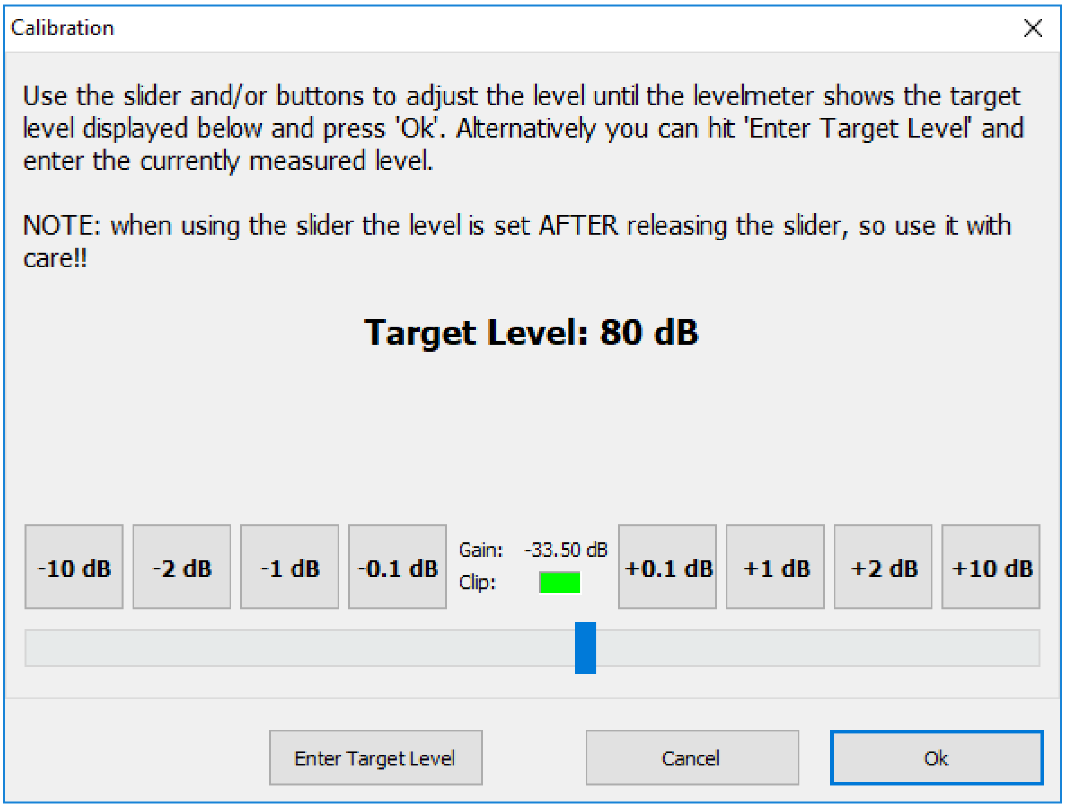
where N is the length of the impulse response and s is the samplerate.
The scaling of the impulse response may be set arbitrary. The calibration procedure will compensate for the absolute amplitude, and the meaurement procedure will take care, that no digital output will occur during a measurement.

4.2.3 Level Calibration

A level calibration is needed to calibrate an output channel (speaker) and for the probe microphone calibration in in-situ-mode.
To calibrate an output channel perform a right mouse button click on the corresponding channel in the list of outputs on the settings dialog. Select the menu item "Calibrate" from the context menu to show the calibration dialog (see figure **??**).
The probe microphone calibration is described in chapter 4.3.1.
A level calibration can be either performed using a level meter or using a reference microphone and acoustic calibrator. To use the acoustic calibrator method please select the option "Use acoustic calibrator and reference microphone for level calibration" in the settings dialog.

4.2.3.1 Level Calibration Using A Level Meter

When starting the level calibration the playback of a Schroeder-phase tone complex is started on the channel to calibrate at an attenuation of 100 dB and a calibration dialog is displayed (figure 14). Place a levelmeter at the desired position and use the dB–buttons or the slider below the buttons to adjust the measured level to 80 dB. **Caution**: when using the slider the new level is set after releasing the slider (mouse button), so use it with care! In the middle between the buttons the current digital attenuation is displayed and an LED indicates, if the output is clipping (LED turns red).



Calibration

Use the slider and/or buttons to adjust the level until the levelmeter shows the target level displayed below and press 'Ok'. Alternatively you can hit 'Enter Target Level' and enter the currently measured level.

NOTE: when using the slider the level is set AFTER releasing the slider, so use it with care!!

Target Level: 80 dB

Gain: -33.50 dB
Clip: 🟩

Figure 14: Level calibration dialog (level meter)

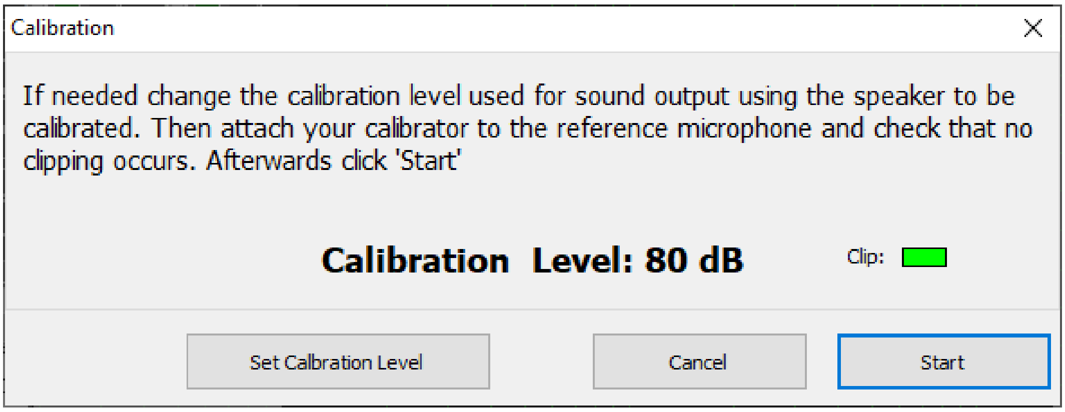
If you cannot reach 80 dB or if you want to use a calibration level other than 80 dB please use the button "Enter Target Level" to change the target level.
After clicking OK the following actions are performed depending on the current calibration step:

– Output channel calibration: The calibration level is stored. Please note, that calibration values are stored separately for different optional impulse responses, thus when changing the impulse response for a channel, the calibration has to be repeated. **Note:** If you are using the spectral equalisation method the calibration has to be repeated after changing the frequency response for a channel!

- Probe microphone calibration: the calibration procedure continues with the next step, please follow the instructions.

4.2.3.2 Level Calibration Using An Acoustic Calibrator

When starting the level calibration a calibration dialog is displayed (figure 15).



If needed change the calibration level used for sound output using the speaker to be calibrated. Then attach your calibrator to the reference microphone and check that no clipping occurs. Afterwards click 'Start'

Calibration Level: 80 dB

Clip: 🟩

Figure 15: Level calibration dialog (acoustic calibrator)

The displayed so called “calibration level” is the level, that will be applied when playing a Schroeder-phase tone complex using the corresponding output channel in the second step. Change the default calibration value of 80 dB if needed before starting the calibration.
Afterwards follow the instructions on the dialog and from additional queries/messages.

4.3 Calibration and Equalisation in In-Situ-Mode

In “In-Situ-Mode” AudioSpike always uses spectral FFT based filters, for properties of these filters see chapter 4.2.1.
The calibration and equalisation process in “In-Situ-Mode” offers the possibility to perform the loudspeaker equalisation and calibration before each measurement “in-situ” without the need of a reference mirophone and levelmeter each time. To achieve this the calibration process in devided into two steps:

1. Equalisation and calibration of the probe microphone (in-situ microphone). For this part a reference mirophone and levelmeter is required.

2. Equalisation and calibration of the in-situ speaker using the calibrated probe microphone at measurement start.

4.3.1 Calibration of the Probe Microphone

The calibration of a probe microphone is invoked from context menu of the “In-Situ Output/Input” section in the settings dialog. After selecting “Calibrate Microphone” you are queried for the maximum and minimum frequency for the calibration. **Important note:** The frequencies entered here are used for the complete in-situ calibration process which means:

- The filter generated for the equalisation of the probe microphone is calculated only within these borders. The identical filter values calculated for the edge frequencies are used for frequencies below and above respectively.

- The same holds for filter generated before each measurement for the equalisation of the in-situ loudspeaker.

Thus it is very important to select these frequencies carefully with respect to

- the sensitivity of the used reference microphone and probe microphone. If frequencies are used where on of the microphones has a very low sensitivity, then you might generate a filter with "artifically" huge compensation values at frequencies, where simply the microphone is not suitable.
- the in-situ loudspeaker: if frequencies are used where the used loudspeaker has a very low efficiency, then this may result in a filter that tries to boost those frequencies extremely.
- "room acoustics": the acoustical in-situ situation may limit the range, where a spectral equalisation is reasonable. For example high frequencies in a closed volume (e.g. an in-situ loudspeaker within the ear canal) will cause standing waves, and this an equalisation might lead to huge errors, when measuring the level at the microphone position, while the situation at the ear drum may be completely different.

After confirming the edge frequencies you are prompted to connect and place the reference microphone. Afterwards the level calibration for the reference microphone is performed, please refer to chapter 4.2.3 for this procedure. After the level was calibrated you are queried to connect/place the probe microphone. **Note:** before clicking "Ok" you may move the query aside and check the displayed recorded spectrum to adjust the probe microphone position.
After clicking Ok the equalisation and calibration of the probe microphone continues automatically.

4.3.2 Calibration of the In-situ Loudspeaker

At the start of a measurement or (free-)search all used in-situ loudspeakers are equalised and calibrated subsequently. For each loudspeaker the following procedure is perfomed:
The spectral equalisation dialog is displayed (see figure 13) and you are queried to connect/place the loudspeaker and probe microphone and adjust the level. Confirm the query and use the volume slider to adjust to a reasonable level. Checking the displayed spectrum recorded with the probe microphone (yellow line) you may adjust the position of the loudspeaker and microphone until everything looks reasonable. After clicking "Start" the equalisation and calibration of the in-situ loudspeaker continues automatically.
After all used speakers are calibrated the measurement or (free-)search respectively starts.

5 Appendix

5.1 Command line parameters

AudioSpike supports command line parameters in format `parameter=value` to load a measurement template, append stimuli to an existing measurement or load an existing measurement result. If a value contains spaces, then the complete parameter and value must be quoted, e.g.

```
"file=C:\Program Files\my file.xml"
```

If AudioSpike is already running then the command line is passed to this running instance. If a command is currently not allowed (e.g. if a measurement is currenlty running) then this is indicated by a red error message in AudioSpike.

Passing settings (directory) to be used

You can specify a subdirectory, where the settings should be read by passing the name with:

```
AudioSpike settings=mysettings
```

These settings have to exist, otherwise the default settings are used (see chapter 3.2).

Loading a template

A fully configured measurement template file (XML) can be loaded to AudioSpike using the following command line

```
AudioSpike command=measure file=mytemplate.xml subpath=mypath
```

where "myfile.xml" is the name of the XML file to load. This command is identical to loading an XML file to AudioSpike using the corresponding menu entry or button respectively.
The optional parameter "subpath" may contain the name of the subpath to use for the result files. If this parameter is specified it will be used as default value when you are prompted for a result subpath on measurement start.

Appending stimuli

You can append stimuli from an XML file to an existing measurement using the following command line

```
AudioSpike command=append file=myappendifle.xml
```

where "myappendifle.xml" is the name of the XML file to be appended. This command is identical to appending an XML file using the corresponding menu entry or button respectively.

Loading a result

An existing result can be loaded to AudioSpike using the following command line

```
AudioSpike command=result file=Results\test_0001\result_0001.xml
```

where the value of "file" is the name of the XML result to load. This command is identical to loading a result XML file using the corresponding menu entry or button respectively.

Loading a result and re-save it with new name

An existing result can be loaded to AudioSpike using the following command line

```
AudioSpike command=result_save file=Results\test_0001\result_0001.xml
```

where the value of "file" is the name of the XML result to load. The file is save immediately with a new name (index increased) and AudioSpike is closed again. This might be useful to convert files to a newer version of AudioSpike.

5.2 Tools for MathWorks MATLAB® and GNU Octave

AudioSpike is shipped with a collection of script files for MATLAB® or GNU Octave for creating measurement template XML-files from structures and loading AudioSpike results back into the workspace.
To use these tools in GNU Octave the Java environment has to be installed and has to be configured properly (AudioSpike uses xerces to read and write XML files).
The tool scripts are located in the `tools` subdirectory of the AudioSpike installation. Usually there is no need to change these files. The files convert MATLAB structs to XML and vice versa and additionally decodes spike waveforms that are stored bas64-encoded in AudioSpike-XML results into vectors. Optionally the raw recorded epoche data are read back to the workspace.
If you have selected the option to save additional MAT-files in AudioSpike you may use the tool script

```
audiospike_loadmat.m
```

to load the MAT file and the addional raw epoche waveforms and the optional probe microphone recording waveforms in in-situ mode (see comments in the script files).
Otherwise you can load the XML files of AudioSpike to MATLAB using the tool script

```
audiospike_loadresult.m
```

This script uses a tool executable to convert an AudioSpike-XML-file to a MAT-file. You may use this tool

```
AudioSpike2MAT.exe
```

located in the BIN subdirectory of the installation manually as well. Simply pass the name of an XML-file, a MAT-file with the same name and MAT-extension is created.
The directory `examples\M-files` contains some example scripts demonstrating the usage of the tools. These examples are well commented and might be a good starting point for creating own measurement templates.

5.2.1 Structure of MAT-files

The MAT-files created by AudioSpike contain a structure with the following values/substructures:

Table 21: MAT-file structure

Name	**Description**
Settings	Contains all values from settings. see table 3, e.g. `Settings.SampleRate`
Parameters	Contains a substructure for each parameter, see table 4, e.g. `Parameters(1).Name`
StimulusSequence	Contains a sequence of indices of stimuli in the order they were presented during the measurement (indices as in Stimuli structure, see below)
Stimuli	Contains a substructure for each stimulus, see table 5, including the value for each used stimulus parameter e.g. `Stimulus(1).Name`
Spikes	Contains a substructure for each stimulus, see table 8, including the value for each used stimulus parameter e.g. `Spike(1).SpikeTime`. The field “Data” contains the waveform of the spike. **Note:** this “Spikes”-structure contains only spikes, that were selected in a cluster plot in AudioSpike.
NonSelectedSpikes	Identical to “Spikes”, but the structure contains all spikes that where NOT selected in a cluster plot in AudioSpike. Thus the structures “Spikes” and “NonSelectedSpikes” together contain ALL spikes detected in AudioSpike.
Epoches	Contains a substructure for each epoche, see discription on page 10, including the value for each used stimulus parameter e.g. `Stimulus(1).Name`. If you have used the tool script `audiospike_loadresult.m` with the flag to load the raw epoche data, then the field “Data” contains the waveform of the epoche.

5.3 Equalisation filter file format

Equalisation filters can be imported to and exported from AudioSpike. The files are stored in ini-style:

```
[Filter]
frequency1=level1
frequency2=level2
....
```

where frequencies must be sorted ascending. The decimal separator is the dot. As the defaut the levels are specified in dB fullscale, i.e. 0 is no amplification/attenuation, -10 is 10 dB attenuation and so on. Optionally you can set levels to be specified as linear factors by adding lin=1 to the first line after [Filter]:

```
[Filter]
lin=1
....
```

Then 1 stands for no amplification/attenuation, 0 is silence.
You can specify as many frequencies as you like, levels between these frequencies are interpolated on a logarithmic frequency scale. The first and last level are used for all frequencies outside the specified range.
The following example specifies a filter that attenuates the signal output between 1000 Hz and 5000Hz by 20 dB

```
[Filter]
999=0
1000=-20
5000=-20
5001=0
```