

Ares Acoustical System

Frequency Response Module

User Manual

Acoustical and Audio Measurement and Design Tools

McIntosh Applied Engineering, LLC (MAE)
MAELLC.COM
info@MAELLC.COM
support@MAELLC.COM

Copyright 2012-2015 by McIntosh Applied Engineering, LLC
Eden Prairie, MN
USA

Notice

Ares is copyrighted and licensed by McIntosh Applied Engineering, LLC (MAE). Ares and all of its hardware and software components are provided "as is". MAE makes no representations or warranties concerning the compatibility of Ares to the user's computer system, or any potential damage caused to any computer, peripherals, digital storage systems, or personal physical safety. Further, the accuracies of the measurement, modeling and design components are not warranted and should not be used as the sole source of evaluating an engineered system or component for commercial suitability or physical safety.

The user agrees not to distribute the Ares software to any non-licensed third party, attempt to disable the licensing system, or reverse engineer the operation of the program or its hardware components.

Table of Contents

Notice.....	2
Table of Contents.....	2
Document Revision History.....	3
Frequency Response Module.....	4
Recording device	4
sound card	4
Playback device	4
sound card	4
PC.....	4
Playback device	5
sound card	5
Recording device	5
sound card	5
PC.....	5
Interface	6
Setup Controls.....	7
Recording device	7
Playback device	10
RS232 COM port for B&K NEXUS	11
Acquisition Controls	11
soundcard calibration	11
stepped sine measurement.....	18
Chirp Measurement.....	29
WAV file measurement	30
Plotting Controls	32
Files/Clipboard controls.....	34
Tools menu.....	35
RLR Calculator	36
SLR Calculator.....	37
TCLw Calculator	40
STMR Calculator.....	43
Channel Noise Measurement	48

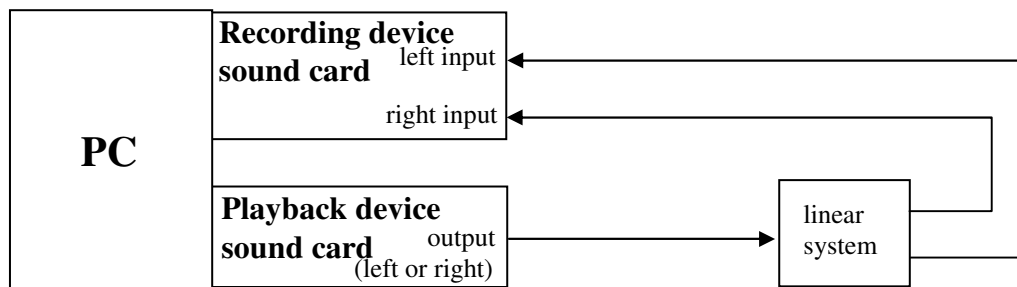
Document Revision History

Version	Date	Comments
1.00	October, 2012	Initial documentation release
1.01	April, 2015	Separated manual into individual documents

Frequency Response Module

The Frequency Response Module uses two channel sound cards to perform simple frequency response measurements of a linear system. The module has a series of tools to compute telephony metrics from the measured responses. These metrics include SLR (Send Loudness Rating), RLR (Receive Loudness Rating), TCLw (weighted Terminal Coupling Loss), STMR (Side Tone Masking Rating), and channel noise.

The basic system configuration is shown below:



Three transfer functions are measured:

$$\frac{\text{left input}}{\text{right input}},$$

$$\frac{\text{left input}}{\text{output}},$$

and $\frac{\text{right input}}{\text{output}},$

where input and output are relative to the PC sound cards and not to the linear system. This will be confusing for users who are used to studying linear systems, but it's consistent with the other Ares measurement modules and with the PC sound card I/O.

The three magnitude spectrums are also measured

left input spectrum
right input spectrum
output spectrum

Two metrics are also computed for the sine wave based measurement methods:

and SNR (Signal to Noise Ratio)
and TND+N (Total Harmonic Distortion plus Noise).

Note: The Frequency Response Measurement module does not have a spectrum analyzer feature. It measures the response of a system to a known input.

Three types of measurement methods are provided:

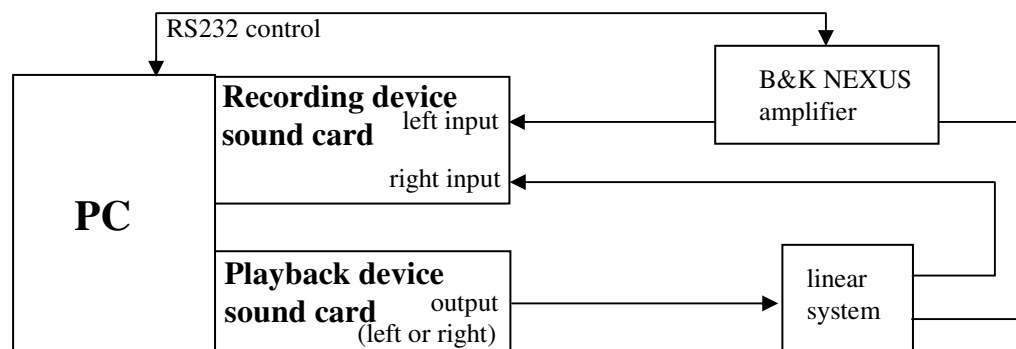
- Stepped Sine
- Chirp
- WAV File

Stepped Sine is the slowest, but provides the most accurate THD+N measurement. Chirp is the quickest and will be preferred for most measurements. WAV File doesn't provide phase measurements, but is useful for measuring the response of telephone systems using real speech. Each is discussed in more detail in their own sections.

The module supports three types of units:

- voltage
- Pascals
- digital or digital full scale (DFS)

To simplify making acoustic measurements with microphones, the module supports using an RS232 interface to a B&K NEXUS amplifier. Ares will read the amplifier's Pa/V sensitivity through the RS232 connection and automatically apply it to the input signals to obtain calibrated Pascal measurements. (Note that this requires that the microphone sensitivity be programmed into the NEXUS.) The block diagram below shows an example of using a NEXUS to amplify the left input channel, but it can also be used for the right input channel as well.

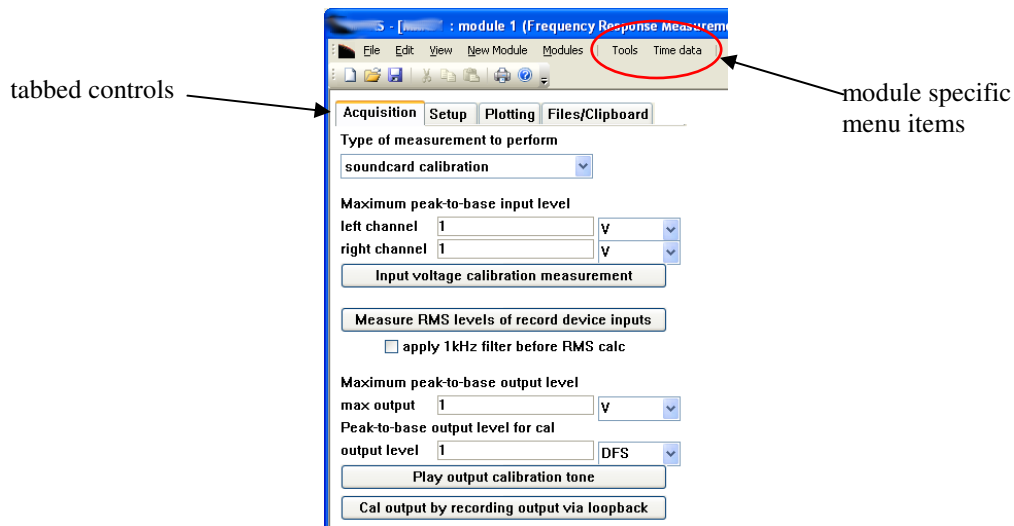


Unlike the Flow Impedance Measurement Module, the Frequency Response Measurement Module will not automatically adjust the NEXUS gains to optimize the input signal level. The user must observe the input signal levels to ensure good SNR levels and that no clipping is occurring. The module will place the text "CLIPPING" in the graph legend if clipping occurred on the inputs for a specific measurement.

The module will not produce a clipped sine wave output. It will automatically reduce the output to prevent a clipped sine wave out of the sound card. However, Ares will allow the output of a WAV file to be clipped, but will warn the user if this has occurred during a measurement.

Interface

The interface for this module consists of a series of tabbed controls on the left side of the Ares window and some module specific menu items.



The *Acquisition* controls will most likely be the most frequently used, and so they appear first on the tab bar. However, before an acquisition can be performed, the controls in the *Setup* tab must be properly set for the desired measurement.

Ares will save the last settings in these controls to the ARES.cfg file and recall them the next time Ares runs.

The *Acquisition* controls are dynamic in that as the user selects different features, the controls that are displayed will change to reflect the required settings for the various features. For instance, if the right input channel is disabled in the *Setup* section, the *Maximum peak-to-base input level* for the right input channel will disappear in the *sound card calibration* section. The benefit of dynamic controls is that the user only needs to enter the information requested on the screen, and the screen won't be cluttered with unused or disabled input controls. The downside of dynamic controls is that there isn't just one display layout to get used to, which can lead to some confusion. Please be patient. Chances are you'll only use a few different setup configurations and you'll quickly get acquainted with the control interface.

In general, to perform an acquisition you will follow the steps given below:

- 1) Use the *Setup* controls to choose the record and playback sound card interfaces
- 2) Use the *Acquisition/soundcard calibration* controls to specify calibration values
- 3) Use the *Acquisition/stepped sine* or *chirp* or *WAV file measurement* controls to perform a measurement

- 4) Use the *Plotting* controls to compare multiple measurements
- 5) Use the *Files/Clipboard* controls to export or input different measurements

Each of these sections will be discussed in detail below.

Setup Controls

The *Setup* tab controls are where the input and output sound cards are selected, as well as the behavior of the left and right channels for those sound cards. The controls are shown below.

The screenshot shows the 'Setup' tab of a software interface. It contains several sections: 'Sound cards for measurement' with 'Recording device' (M-Audio Transit USB), 'left chan' (mic: NEXUS ch 2), and 'right chan' (voltage measurement); a checkbox for 'release sound card resource'; a 'Playback device' section with 'M-Audio Transit USB', 'Output channels' (left & right), and 'Output units' (voltage); and an 'RS232 COM port for B&K NEXUS' section with 'COM1'. Annotations with arrows point to these controls: 'select sound card for input signals' points to the Recording device; 'select behavior of left & right channels' points to the right chan; 'select sound card for output signals' points to the Playback device; 'select which channel the output is played onto' points to the Output channels; 'select units for output' points to the Output units; and 'select COM port for NEXUS control' points to the COM1 port. A note at the bottom states: 'Note: the NEXUS must be properly with the microphone calibration Also, set baud to 2400 and echo on.'

Recording device

All of the sound cards that are available for recording will be displayed in the recording sound card list box. Ares doesn't differentiate between sound cards with stereo or mono inputs. A mono recording sound card can be used for single input channel acquisition, but a stereo input device is needed for a left/right transfer function. It's the author's experience that most laptops will have a built in sound card that only has a mono input channel.

The input options for each left & right channel are shown in the list box below.

The screenshot shows a dropdown list box with the following options: 'mic: NEXUS ch 2' (selected), 'not used', 'voltage measurement', 'mic: NEXUS ch 1', 'mic: NEXUS ch 2', 'mic: NEXUS ch 3', 'mic: NEXUS ch 4', 'mic: manual cal value', and 'digital level [0-1]'.

not used

voltage measurement

No data will be recorded from this channel.

Treats the channel as a simple AC voltage input. This will require that the input channel be calibrated. The calibration is given in terms of the maximum peak-to-base sine wave input that can be recorded on the

	channel before the input is clipped. See the <i>soundcard calibration</i> selection in the <i>Acquisition</i> section for more details.
<i>mic: NEXUS ch 1-4</i>	The input for this channel is to be from a NEXUS amplifier that's connected to a RS232 port. These inputs are still treated as voltage inputs for calibration purposes, but the output sensitivity in V/Pa will be read from the specified NEXUS channel to provide the voltage-to-Pascal conversion for the channel. That conversion will be applied to the signal and the data will be reported with units of Pa. To use this option, the NEXUS must be connected and properly calibrated for the microphone being used.
<i>mic: manual cal value</i>	As with the <i>mic:NEXUS ch 1-4</i> option, the input will be treated as a voltage input for calibration, but the V/Pa conversion value for this channel is manually entered by the user and a NEXUS amplifier is not required. The calibration value is entered in <i>soundcard calibration</i> selection in the <i>Acquisition</i> section.
<i>digital level (0-1)</i>	The input will be treated as a digital signal whose levels are normalized to have a maximum value of 1 and a minimum value of -1. (The "(0-1)" part of the descriptor is meant to indicate this normalization.) Many communications networks deal with digital levels rather than voltages. For a digital signal, the units will be specified as DFS for Digital Full Scale. In DFS units, a full scale sine wave will have an RMS level of 0.707 DFSrms, or -3dB DFS.
	If one input is configured as digital, both inputs must be configured as digital. A mixture of voltage and digital inputs for the left and right input channels isn't supported.
	A digital input level calibration will be presented in the <i>soundcard calibration</i> selection in the <i>Acquisition</i> section. If there are differences in digital levels between the PC and the system being measured, this digital calibration value can be used to compensate for it.

release sound card resource

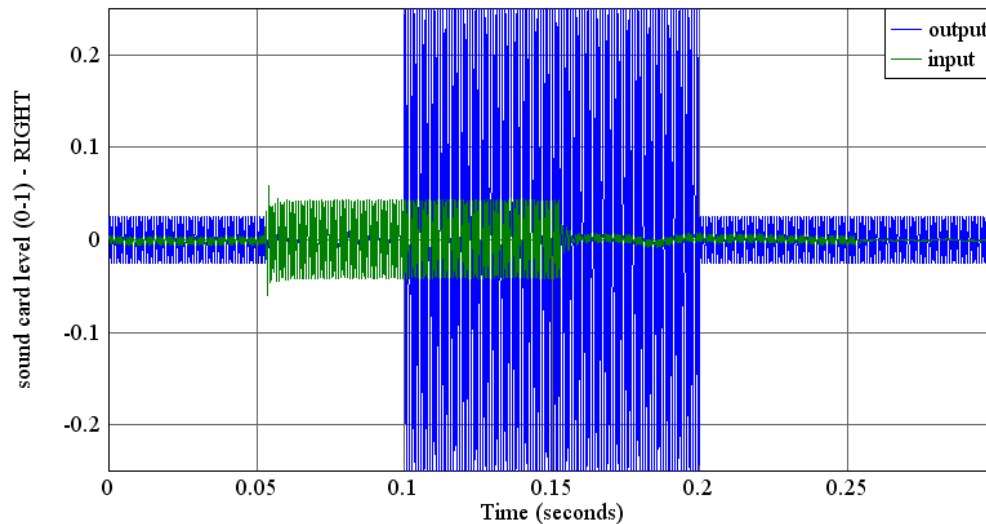
It has been a challenge to get the Ares acquisition software to work across all versions of Windows and sound card drivers. On some systems, not releasing the sound card resources after they have been used has made some sound card drivers usable. An option has been provided to allow the user to select whether the sound card resources are released or not. The user will have to experiment with this setting and their sound cards to see if it improves performance with a given sound card device. If the problem persists, contact McIntosh Applied Engineering (MAE) for assistance.

Play/Record pulse from soundcards

This button can be used to test the sound card configuration for proper operation. It plays a short tone burst out of the sound card (both the left and right channels), while simultaneously recording the input signals. The signal consists of three sections of a 1kHz tone:

100ms 1kHz tone at $1/40^{\text{th}}$ of the maximum output level
100ms 1kHz tone at $1/4^{\text{th}}$ of the maximum output level
100ms 1kHz tone at $1/40^{\text{th}}$ of the maximum output level.

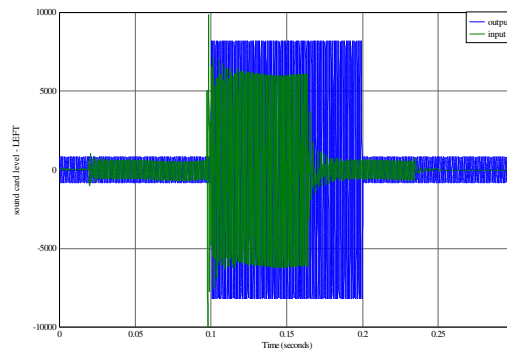
The output of this sequence must be measurable by the sound card input channels. This can be done by connecting the output to a speaker and measuring the speaker pulse with microphones connected to the left and right inputs, or by simply connecting the left and right inputs directly to the sound card output. If the sound cards are working properly, the result should be something like that seen below. There will be two graphs, one for the left channel, and one for the right channel.



The blue curve is the output signal, and the green curve is the input signal. There are two very important aspects of the recorded signal that must be observed to verify that the sound cards are working properly.

First, the input signal (green curve) must have some portion of its loudest central 100ms segment overlap the output signal's (blue curve) loudest central part. In the graph above, the input (green) signal has the last ~50 ms overlap with the beginning of the output (blue) curve from time index 0.1 to 0.15. Ares tried to start output playback and input signal's recording at the same time, but as is seen in the above graph, the sound card didn't start capturing the input signal until ~50 ms after the output signal began playing out. If there is too much delay between the start of the playback and the recording, the data acquisition routines won't properly synchronize the signals and there will be a problem. By ensuring that the central section of the input (green) curve overlaps with the central section of the output (blue) curve, there will be less than a 100ms delay between playback and recording, which the data acquisition software will properly handle.

The second important property is that the length of the central section of the recorded signal should be the same length as the central section of the playback signal. With some USB sound cards, it has been observed that they will lose samples and so the recorded signal will actually contain less data than the signal that was played out. This can be seen in the graph below, where the recorded central section (green curve) is only about 70 ms long instead of 100ms. If this happens, the sound card configuration cannot be used.



The above behavior has been seen when using a single USB sound card for both the playback and recording. When a second USB sound card was purchased, and one was used for playback and the other for recording, the problem disappeared.

Playback device

This section is where you select what sound card is to be used for the output signal, which channels (left or right) the signal should be played out of, and the units to assume for the output signal. The playback device does not have to be the same sound card as the recording device.

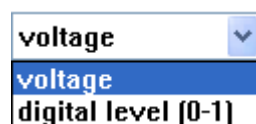
Output channels

This option controls whether the output signal is played out of the left, right or left and right channels simultaneously.



Output units

This option specifies the units to be used for the output channel. Two units are available, voltage or digital. A calibration value for both can be specified in the *soundcard calibration* selection in the *Acquisition* section.



Play fullscale chirp out L&R chans

This button causes a full scale (-1 to 1 normalized digital units) chirp signal to be played out of the playback soundcard's outputs. This is useful for setting amplifier gains for an external power amplifier that's being used to drive an excitation speaker. It is recommended that you:

- turn the power amplifier gain to its minimum value
- press the *Play fullscale chirp out L&R chans* button
- slowly turn the gain on the amplifier to the highest value without excessive distortion of the speaker (alternatively you can place a volt meter on the speaker and turn the gain up until the maximum drive voltage is achieved)

This way, you can ensure that the measurement module won't overdrive the speaker during a measurement.

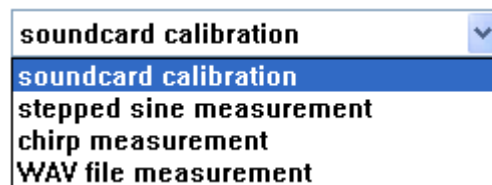
The frequency range that's used for the chirp is taken from the starting and ending frequencies specified in the *Acquisition/stepped sine or chirp measurement* sections. The chirp is played until the button is pressed again.

RS232 COM port for B&K NEXUS

This specifies the COM port that's being used to control the B&K NEXUS amplifier. If calibrated microphone levels are required, it's extremely convenient to simply have Ares read the calibration levels directly from a B&K NEXUS amplifier. While it's not necessary to use a NEXUS amplifier, using a NEXUS is highly recommended. It allows for the microphone gain to be changed without having to manually change the microphone's sensitivity in the Ares interface, preventing the accidental loss of proper calibration during measurements.

Acquisition Controls

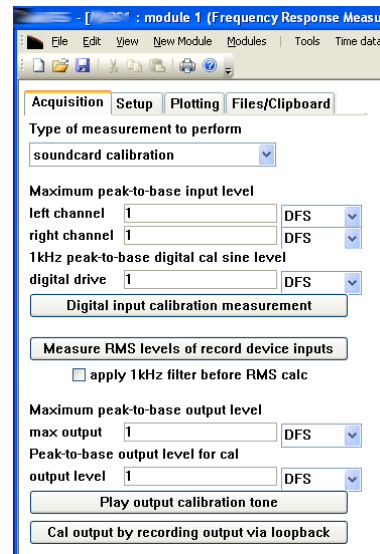
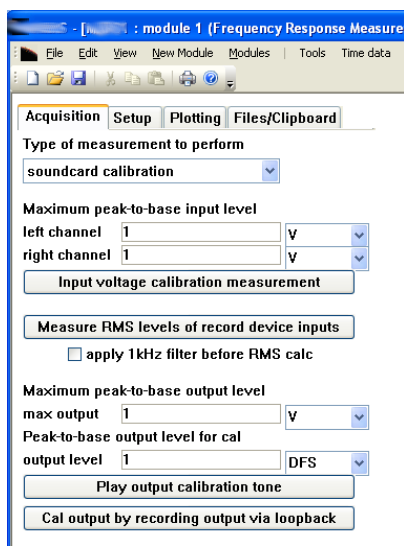
The *Acquisition* tab is where most of the measurements are performed. It consists of several sets of controls selected with the *Type of measurement to perform* list box. The list box is shown below.



soundcard calibration

The sound card input and output calibration values are specified in this section. The controls change slightly depending on whether the inputs or outputs are specified to be analog voltages,

microphone inputs, or digital levels. The screen capture below and to the left is for when input and outputs are both voltages, and the capture on the right is for when they are both digital.



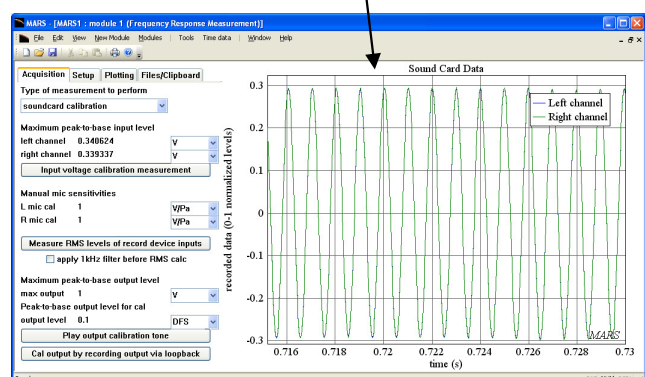
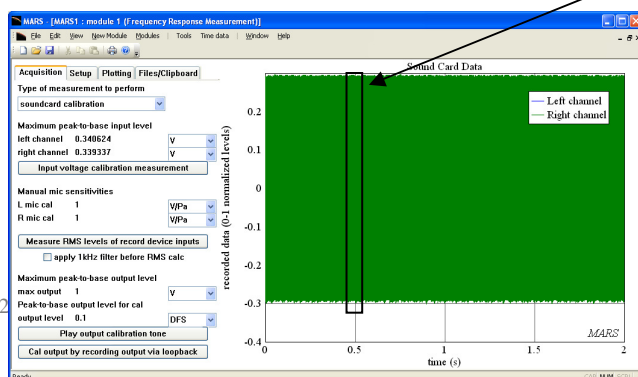
Calibration for voltage inputs

When the inputs are voltages or microphones, calibration is provided specifying the maximum peak-to-base input levels. This is the maximum voltage that can be sampled by the sound card inputs without clipping. The *Input voltage calibration measurement* button can be used to perform a calibration of the input levels by recording a known voltage signal on the inputs. It requires that a 1 kHz, 100 mV peak-to-base signal be connected to the sound card inputs. The MAE200-ACVC1-A generates such a signal and is available from MAE.

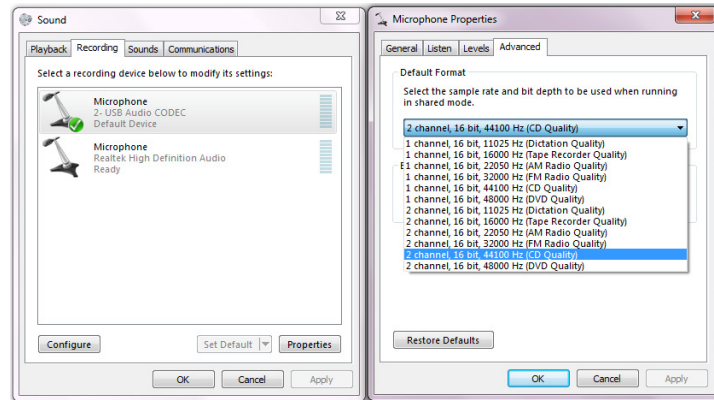
A photograph of the MAE200-ACVC1-A connected to the inputs of a USB sound card is shown below along with screen captures of the resultant calibration. Note that the left channel's maximum input was 0.340624 and the right was 0.339337. While these values are very close, it's important that they're not identical.



right click on graph, select Zoom, draw a rectangular zoom box around the data to better view the 1kHz waveform and verify that it's not clipping or distorting.



If the calibration of the left and right channels results in **identical** calibration values for both the left and right input channels, the sound card is probably configured as a mono input. When this occurs, look for a setting for using the sound card in shared mode and set it to a 2 channel 44.1kHz configuration. Go to the control panel, select *sound*, then *recording*, right click on the sound card and select *properties*, select *advanced*, and then select *2 channel, 16 bit 44100 Hz (CD Quality)* from the drop down box. The dialog boxes for doing this in Windows 7 are shown below.

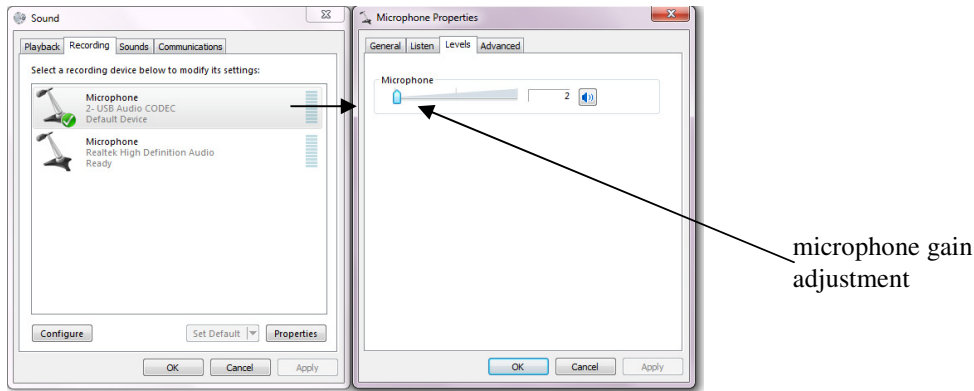


Also, look at the recording settings and turn off any advanced features such as bass boost or dynamic gain adjustments. The sound card should be just a simple input device with no special processing features enabled.

Many sound cards are configured for their analog inputs to be compatible with external microphones, providing both phantom power and a considerable amount of gain. In this case, the large gain will likely cause the *Maximum peak-to-base input level* to be very small, perhaps less than 100 mV. Since most signals from an amplified microphone or voltage across a speaker will be much higher than 100 mV, the gain on the sound card inputs may need to be reduced. The user will need to adjust the input gains to achieve a desired maximum input level. Typically, a maximum input level between 1 and 10 V is desirable for most applications. (However, if non-amplified microphones are being measured, perhaps a high input gain is exactly what's needed.)

If the maximum peak-to-base input level is less than 100 mV, the MAE200-ACVC1-A cannot be used to calibrate the input voltages.

Since every sound card driver interface is different due to various drivers and Windows versions, the user must figure out how to adjust the input gains for their particular sound card. However, in general, you want to go to the Control Panel, select *Sound*, select *Recording*, right click on the sound card being used for recording, and then go through the controls to adjust the microphone gain. The dialog boxes for doing this in Windows 7 are shown below.



Calibration for digital inputs

For digital input signals, the calibration button changes to *Digital input calibration measurement* and a *digital drive* level is required to be specified. A similar calibration is performed, but in this case the user is required to place the specified digital drive level on the soundcard's inputs. How this is done is left to the user. However, in almost all circumstances, the maximum peak-to-base input level for digital systems will be 1 (i.e. unity). The option to calibrate digital levels is only provided in the unlikely event that there is a digital level mismatch between the sound card and the digital system it's connected to.

The recorded signal is multiplied by the calibration value. So if a full scale digital signal from the system being measured only registers as a 0.5 DFS signal on the PC sound cards, the calibration constant should be set to 2.0. This way, a 0.5 DFS signal ready from the input will be scaled up to a 1.0 DFS signal.

As an example of when being able to set the DFS calibration value is useful, consider a connection with a Bluetooth headset made using a Rohde Schwarz CBT. The CBT has analog outputs (for the uplink signal to the headset) and analog inputs (for the downlink signal to the headset). Thus, an analog signal is being used to represent a digital Bluetooth signal.

The following was found for the system:

maximum input level for the CBT for the downlink signal to the headset = $1.50 V_{\text{peak-to-base}}$
 maximum output level for the CBT for the uplink signal from the headset = $1.06 V_{\text{peak-to-base}}$

maximum input level for the recording analog sound card = $1.98 V_{\text{peak-to-base}}$
 maximum output level for the playback analog sound card = $1.90 V_{\text{peak-to-base}}$

If the recording sound card input is configured as a digital signal and connected to the output of the CBT, its effective digital calibration level will be:

$$\frac{1.98 \text{ max input voltage}}{1.06 \text{ max output voltage from CBT}} = 1.87$$

So the DFS input calibration will be 1.87, meaning a full scale input on a recording channel will equal a 1.87 DFS value.

If the playback sound card output is configured as a digital signal and connected to the input of the CBT, its effective digital calibration level will be

$$\frac{1.90 \text{ max output voltage}}{1.50 \text{ max input voltage}} = 1.27$$

So the DFS output calibration will be 1.27, meaning a full scale output on the playback channel will equal a 1.27 DFS value on the CBT.

The screen captures below show how the analog sound cards could be configured as digital devices and the above digital calibration values manually entered into the *soundcard calibration* section of the *Acquisition* controls.

The left screenshot shows the 'Setup' tab with the following settings:

- Sound cards for measurement
- Recording device: M-Audio Transit USB
- left chan: digital level (0-1)
- right chan: not used
- ☐ release sound card resource
- Play/Record pulse from soundcards
- Playback device: M-Audio Transit USB
- Output channels: left & right
- Output units: digital level (0-1)

The right screenshot shows the 'Acquisition' tab with the following settings:

- Type of measurement to perform: soundcard calibration
- Maximum peak-to-base input level
- left channel: 1.87 DFS
- 1kHz peak-to-base digital cal sine level
- digital drive: 1 DFS
- Digital input calibration measurement
- Measure RMS levels of record device inputs
- ☒ apply 1kHz filter before RMS calc
- Maximum peak-to-base output level
- max output: 1.27 DFS

Manual mic sensitivities

If the left or right input channel is specified to have a *mic: manual cal value* input, a section will appear between the *Input voltage calibration measurement* and the *Measure RMS levels of record device inputs* button. This section, shown below, will require that the user manually enter the microphone sensitivity for each channel that's configured with the *mic: manual cal value* option.

The screenshot shows the 'Input voltage calibration measurement' section with the following settings:

- Manual mic sensitivities
- L mic cal: 1 V/Pa
- R mic cal: 1 V/Pa
- Measure RMS levels of record device inputs

The module doesn't have a direct means to calibrate a microphone, but the *Measure RMS levels of record device inputs* button can be used to read the signal level from a microphone when a calibrator is placed on the microphone.

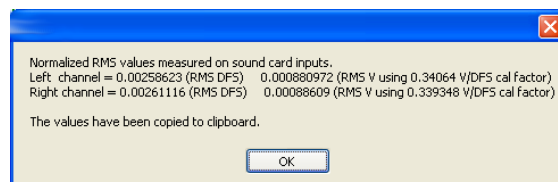
To calibrate a microphone, insert it into a calibration device that generates a known SPL level and record the voltage from the device using the *Measure RMS levels of record device inputs* button. Divide the reported voltage by the known pressure in Pascals to come up with the volts/Pa calibration value. If the calibration source is a 1 kHz signal, the *1kHz filter before RMS calc* option can be selected to apply a 1 kHz band pass filter to the recorded data, improving the accuracy of the calibration.

A B&K Type 4231 is a typical calibrator that generates a 1kHz tone at 94 dB SPL or 1 Pa. If the Type 4231 is used to calibrate a microphone and a value of 50 mV is measured, the microphone's sensitivity will be $50 \text{ mV} / 1 \text{ Pa} = 50 \text{ mV/Pa}$.

Note that before this technique can be used, the sound card inputs must be calibrated with accurate *Maximum peak-to-base input levels* specified.

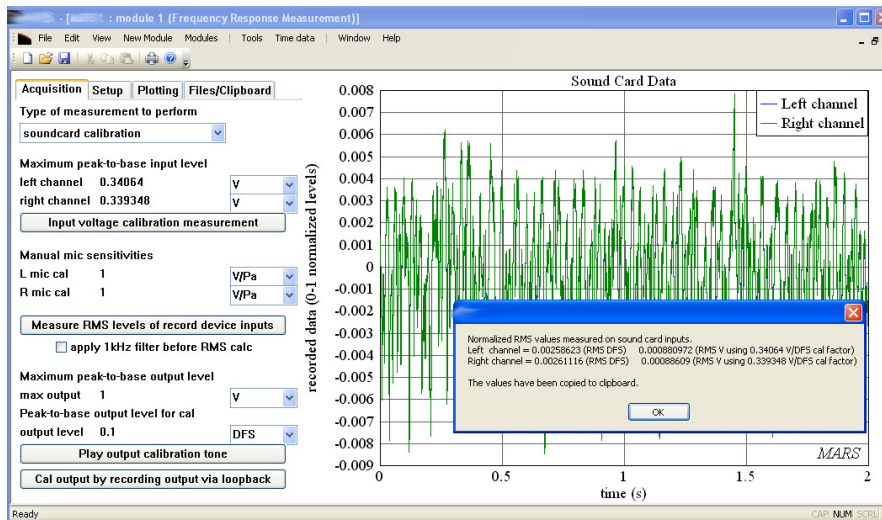
Measure RMS levels of record device inputs and 1kHz filter before RMS calc

This button will cause the RMS level of the sound card inputs to be measured and reported to a dialog box. The reported values include the DFS units read directly from the normalized sound card sample values, along with the voltages (obtained by applying the channel's voltage calibration values to the DFS values).



The *1kHz filter before RMS calc* option can be selected to apply a 1 kHz bandpass filter to the data before the RMS level is computed. As discussed above, this can increase the calibration accuracy of a manually calibrated microphone.

The time domain data that's acquired during the RMS measurement will be plotted in the right side of the Ares window. An example of this is shown below. When performing a calibration, the quality of the calibration signal should be observed by using the right click graph menu and zooming in on the data. However, in the case shown below, the inputs only had noise on them so a nice 1kHz calibration tone is not present. You would not want to use such a signal for calibration!



Maximum peak-to-base output level

The output calibration is specified by the maximum peak-to-base signal that can be generated by the playback device's output. This value is entered into the *max value* parameter. Two buttons are provided to provide two means for calculating the output calibration value. While either technique can be used, if the sound card inputs are voltages and have already been calculated, the second technique, *Cal output by recording output via loopback*, is probably the preferred technique.

Play output calibration tone

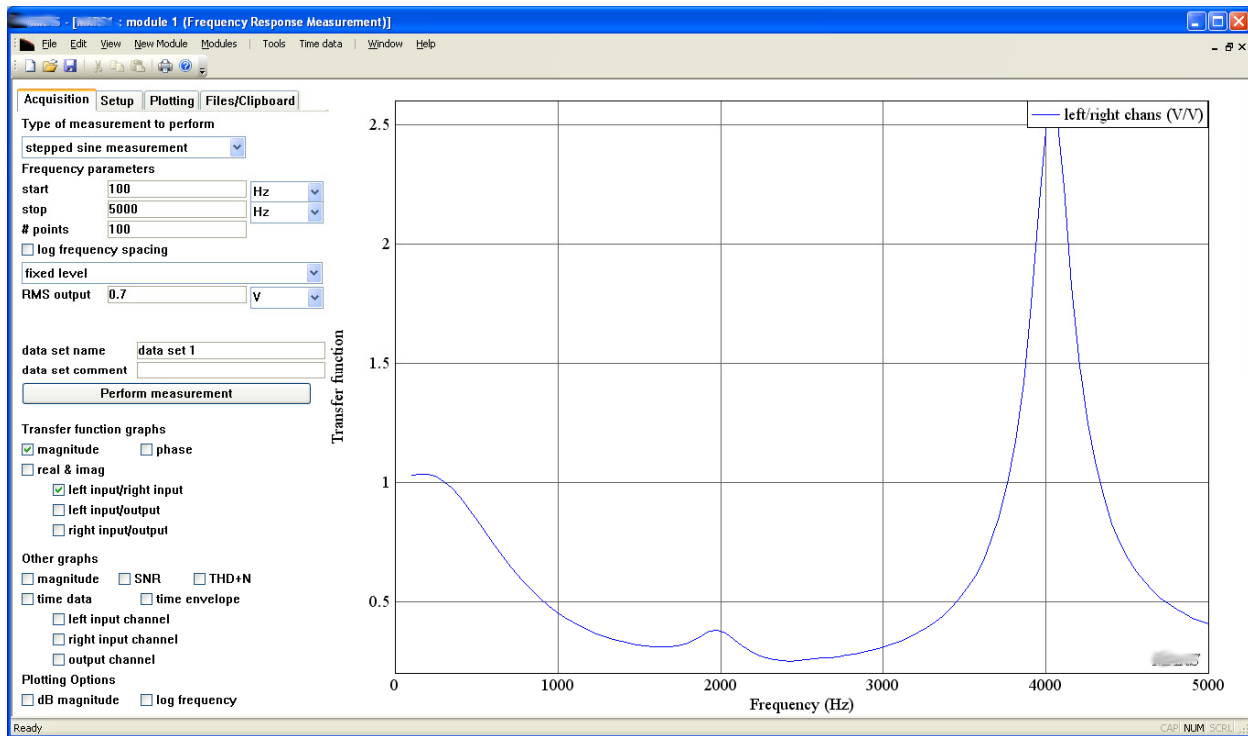
This button will play a 1 kHz tone out of the playback device. The level is specified in the output level parameter. By measuring the AC level generated by the tone with a volt meter, the output calibration value can be computed. Before the tone is played, a dialog box is presented to the user giving details on how to compute the calibration value.

Cal output by recording output via loopback

If the sound card inputs have been calibrated, the sound card's output can be calibrated by connecting it to the left channel sound card input. When this button is pressed, a dialog box will appear giving the user instructions on how to configure the system for calibration. Care must be taken to ensure that the sound card output doesn't clip the sound card input. If it does, the user will be prompted with a clipping warning. You should reduce the *output level* value and try the calibration again.

stepped sine measurement

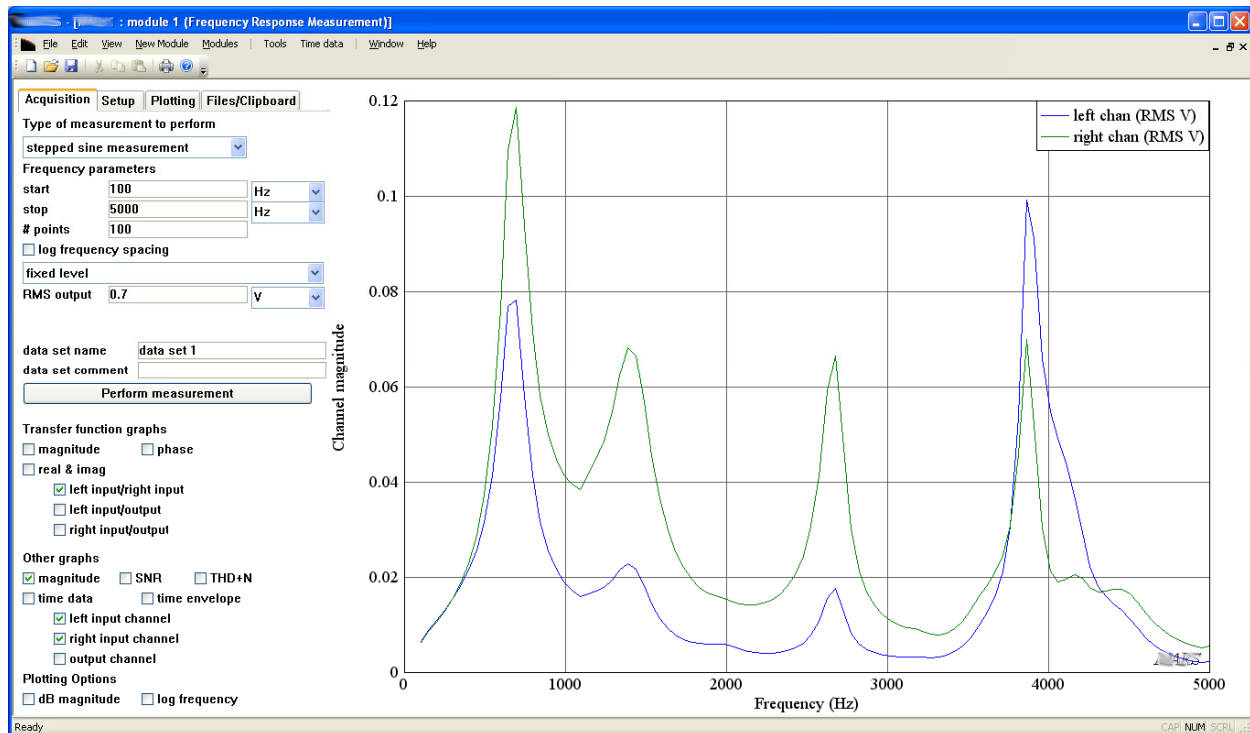
The stepped sine measurement acquisition section provides an interface to measure a response using pure tones. A series of tones from the starting frequency to the stopping frequency are generated. The number of frequencies are specified along with whether they are linearly or logarithmically distributed. A screen capture of the controls is shown below along with a graph of the left input/right input transfer function magnitude that was last measured.



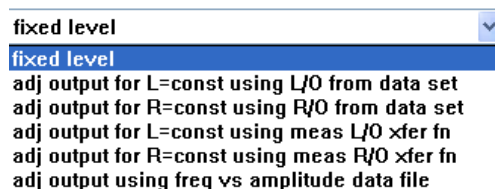
Once a measurement is successfully completed, the measured results will be copied into a data set buffer with the given data set name and comment. By default, the name will be “data set 1” and the comment will be blank. The data sets can be plotted in the **Plotting** section, or saved to individual files in the *Files/Clipboard* section.

The data for the last successful measurement can be plotted in the *Acquisition* section using the plotting options in the lower left corner of the screen.

The screen capture below shows the magnitude of the left and right input channels plotted for the last measurement. Notice that their levels vary substantially. For this measurement, the output level was held constant at an RMS level of 0.7 V for all frequencies. It is sometimes desirable to vary the output voltage to obtain a uniform input voltage level. This can be done using the drop down list box that allows you to choose how the output should be controlled.



The output control list box is shown below. Each option will be discussed.



fixed level

A *fixed level* keeps the output voltage constant for all frequencies. The output level is specified in the *RMS output* control that will appear. This control is shown below. Note that the level is in RMS units and the output calibration value entered into the *Acquisition/soundcard calibration* section will be used to convert the desired voltage level to a sound card output level that will achieve the desired voltage.



If Ares determines that the RMS output level is too large to achieve, a dialog box will alert the user that the output is too large and offer to reduce the level to an achievable value.

If the output is specified to be a digital value, the controls will ask for a digital RMS value rather than an RMS voltage. Note that a full scale sine wave will have a value of 0.7071 RMS DFS or -3 dB DFS.

adj output for L=const using L/O from data set and
adj output for R=const using R/O from data set options

These options should be read as:

“adjust the output to achieve a constant left channel value using a left input/output transfer function from a data set”

or

“adjust the output to achieve a constant right channel value using a right input/output transfer function from a data set”

When one of these options is selected, you will be asked to specify a left/output or right/output transfer function from an existing data set to use in computing an output voltage as function of frequency that will achieve a desired left or right input value. The controls that appear for this option are shown below.

adj output for L=const using L/O from data set ▼		
RMS target	0.7	Pa ▼
data set EQ	1: data set 1 ▼	

The *RMS target* is the desired value to be achieved on the left or right channel. The **data set EQ** list box will list all of the data sets held in the module’s buffer. Once a data set is selected, the left input/output or the right input/output transfer function will be used to compute the output voltage for each frequency. If H is the selected transfer function so that

$$H = \frac{\text{input channel}}{\text{output channel}}$$

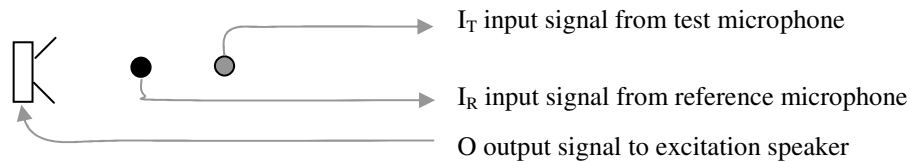
and T_{RMS} is the target RMS value for the desired input channel, then the output channel level will be

$$\begin{aligned} \text{RMS output} &= \left| \frac{T_{\text{RMS}}}{H} \right| \\ &= \left| \frac{T_{\text{RMS}}}{\text{input/output}} \right| = \left| \text{output} \cdot \frac{T_{\text{RMS}}}{\text{input}} \right| \end{aligned}$$

Note that the units of T_{RMS} will be equal to the units of the input channel from the data set that H is taken from, and not the units that the input channel are presently set to for the current measurement. So in the controls above, the **data set 1** left/output transfer function was measured with the left input channel being set to measure a microphone response (Pascals).

It is important to understand that it is not required for the current measurement to have the same input units as the input units when H was measured.

Consider the following situation. A common audio measurement is to measure the response of a microphone relative to the response of a reference microphone as shown below.



To see the response of the test microphone, one would typically connect the test microphone signal I_T to the left input channel, and the reference microphone signal I_R to the right input channel, and then plot the left input/right input transfer function.

However, if the signal from the test microphone is digital, and the reference microphone is an analog voltage, then the two microphone signals can't be measured simultaneously because sound cards do not support mixing analog and digital inputs.

In this case, one can measure the transfer function $H1$ of the reference microphone over the output signal

$$H1 = \frac{I_R}{O} \left(\frac{Pa}{V} \right)$$

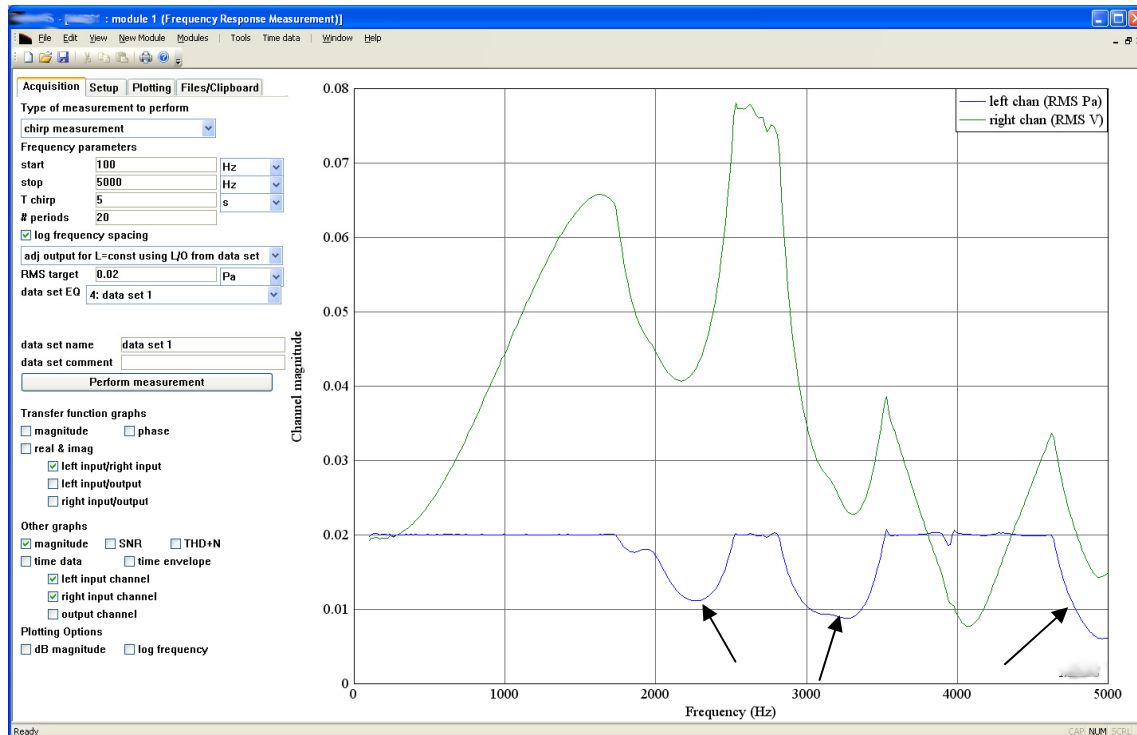
which will have units of Pa/V. Then select one of the *adj output for L=const using L/O from data set* or *adj output for R=const using R/O from data set* options so the $H1$ transfer function can be selected to adjust the output voltage, and set the RMS Target value to be 1 Pa. This will drive the speaker at a level that will achieve a 1 Pa pressure at the reference microphone, even without the reference microphone being measured. We will use this to measure the test microphone so that its magnitude response will be relative to this 1 Pa pressure reference level.

When the test microphone signal I_T is measured and the magnitude of the input channel is displayed, the curve will be relative to a 1 Pa reference microphone level, which is essentially what a I_T/I_R transfer function would look like. Note that the I_T magnitude plot will have units of DFS, and not DFS/Pa, however, the curve will essentially be DFS/Pa since it was measured for an input level of 1 Pa.

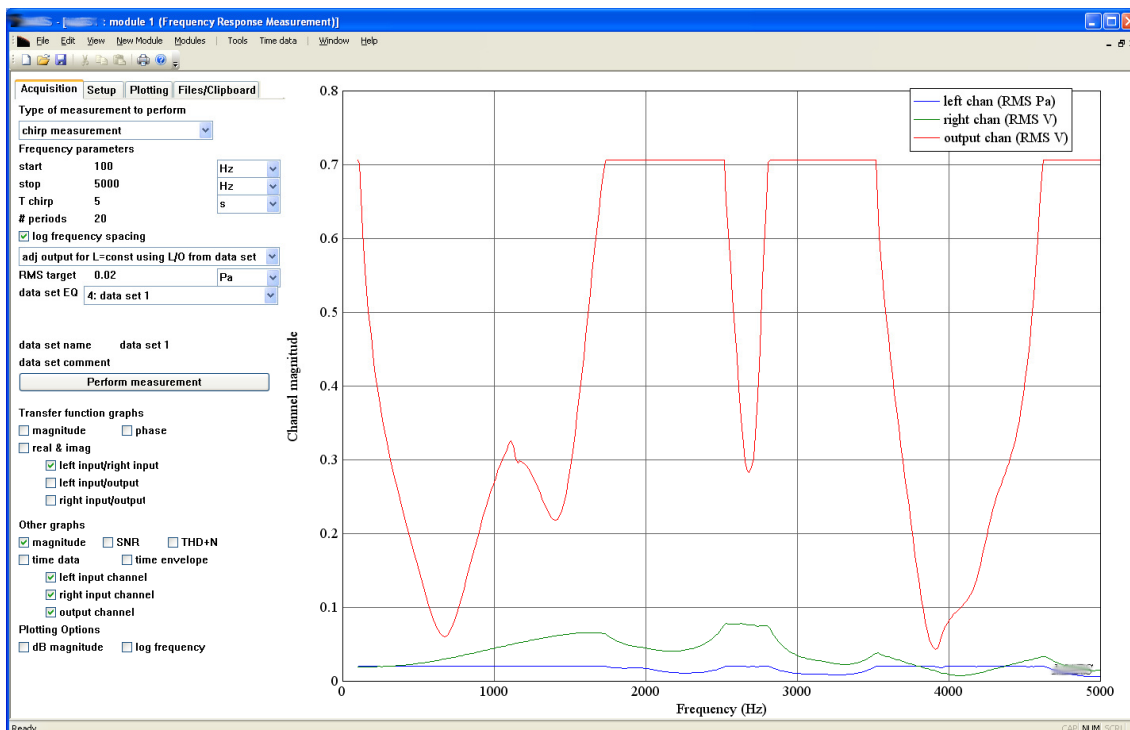
Note that if the $H1$ measurement was performed with a different set of measurement frequencies, the data will be interpolated for the frequency values for the current measurement. So it is not necessary for the $H1$ measurement to have the same frequency settings as the current measurement.

It's possible that the output level computed using this method may exceed the output capability of the playback soundcard device for some frequencies. In this case, Ares will warn you that the required output level exceeds the systems capability and will ask if you want to continue or abort. If you continue, Ares will choose the maximum output possible for those frequencies.

For the measurement shown below, a target level of 0.02 was selected for the left channel input, but the system couldn't achieve that level for all frequencies. The arrows point to the frequencies where the 0.02 level couldn't be achieved.



The output voltage has been added to the magnitude plot shown below. The frequencies where the desired left input level couldn't be achieved are the frequencies where the RMS output level was limited to $0.707 V_{RMS}$.



As will be seen with the *adj output for L=const using meas L/O xfer fn* and *adj output for R=const using meas R/O xfer fn* options, the transfer function H1 can be improved through iteration to achieve a L or R = constant value with less deviation from the target value. Usually one measurement will achieve a tolerance of about +-1 dB about the target value. If a tighter tolerance is required, try using a *adj output for L=const using meas L/O xfer fn* or *adj output for R=const using meas R/O xfer fn* with several iterations for the H1 measurement.

adj output for L=const using meas L/O xfer fn and adj output for R=const using meas R/O xfer fn options

These options should be read as:

“adjust the output to achieve a constant left channel value using a measured left input/output transfer function”

or

“adjust the output to achieve a constant right channel value using a measured right input/output transfer function”

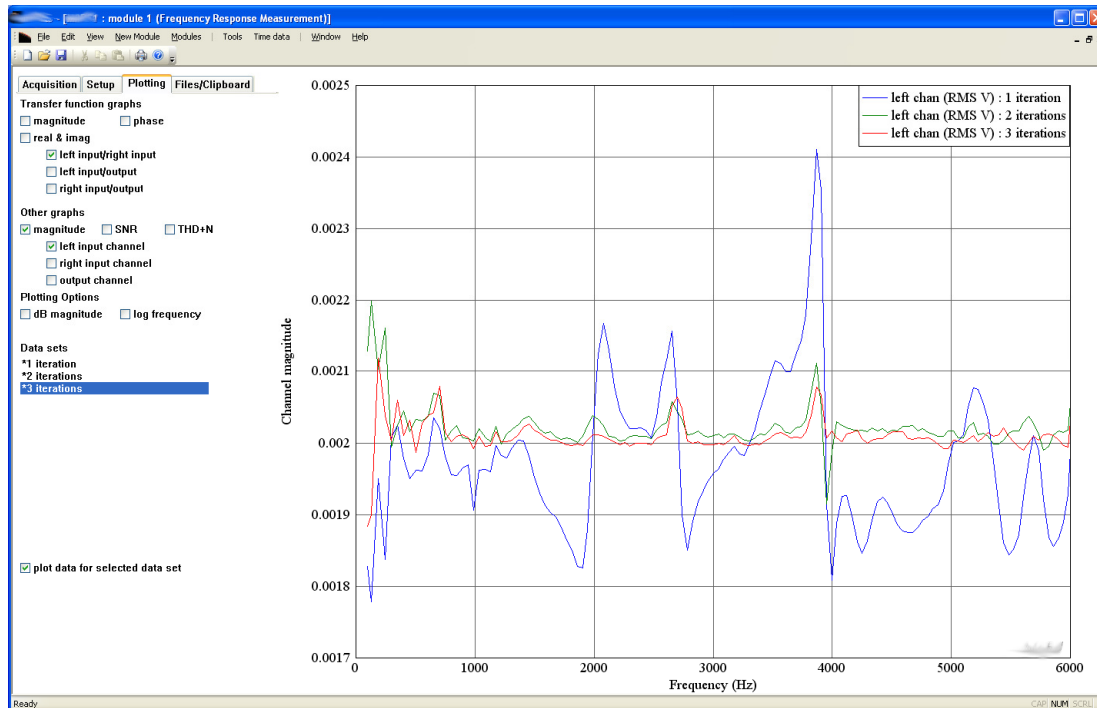
These options are similar to the *adj output for L=const using L/O from data set* options described earlier, however, the transfer function that’s used to compute the desired output is measured when the *Perform measurement button* is pressed. So there will be at least two measurements made. The first measurement is made to measure the input/output transfer function H1. H1 is then used to compute the output necessary to achieve the desired *RMS target* level. Then this adjusted output is used to perform a second measurement, which should achieve the desired *RMS target level*.

In this case, the *RMS target* units will be the units of the associated input channel.

adj output for L=const using meas L/O xfer fn		
RMS target	0.02	Pa
# of iterations	2	
initial output	0.7	V

There is an option to perform more than one measurement to estimate H1. This is controlled by the *# of iterations* parameter. One iteration means that only one measurement is used to estimate H before the final measurement is made. Increasing this to 2 means that two estimates for H1 are made, each one improving the estimate for the output that will achieve a constant value for the input spectrum. So the total number of measurements made will be the *# of iterations* plus one.

The screen capture below shows the left input channel magnitude for measurements with three different values for the *# of iterations*, 1, 2 and 3. As can be seen, the target value 0.002 for the left channel improves as the *# of iterations* is increased.



The *initial output* parameter determines the constant output level that's used for the first measurement. After the first measurement, the output will vary with frequency to achieve the *RMS target* value.

adj output using freq vs amplitude data file

This option takes the output drive level from a text data file. For this option, a single extra control is created that allows you to browse for the data file that's to be used.

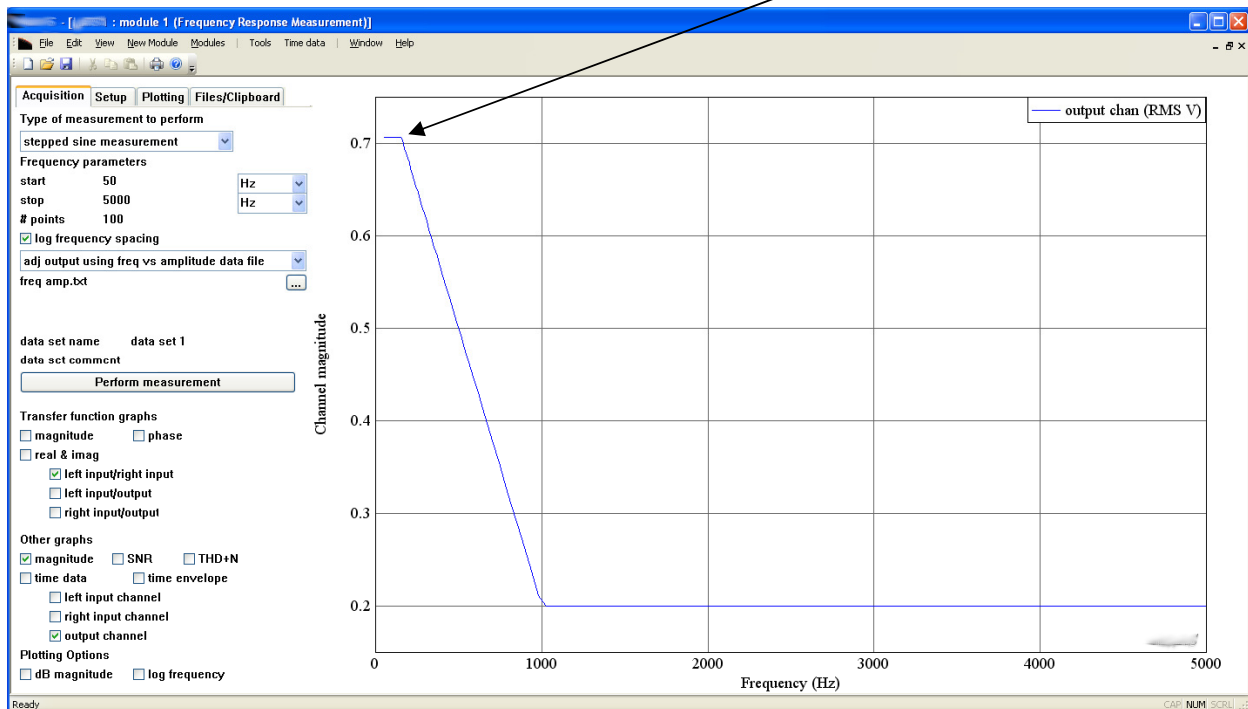


The format of the data file is to be two numerical text columns, separated by spaces. The first column is frequency and the second column is the RMS output level for the corresponding frequency. An example of the contents of a data file is given below. It consists of three lines, specifying the amplitude at 500, 1000, and 2000 Hz.

```
500 .5
1000 .2
2000 .2
```

For frequencies not specified, the data will be linearly interpolated and/or extrapolated to obtain levels for other frequencies. For the data above, the output level will be a constant 0.2 for all frequencies above 1000 Hz. For frequencies between 500 and 1000 Hz, the output will be interpolated between 0.5 and 0.2. For frequencies below 500 Hz, the output will be extrapolated from the 500 and 1000 Hz data points. Note that the extrapolation operation can cause the output to easily go beyond the limits of the system. Ares will detect these overdrive points and limit the output.

The output magnitude for the “freq amp.txt” data for a measurement from 100 to 5000 Hz is shown below. Note that for frequencies below 150 Hz, the output was limited to a value of 0.7071.



data set name ***data set comment***

Each data set is given a name and contains comment text. Once a measurement is successfully completed, the data is copied to the last position in the module’s data set buffer. The data set will be given the name specified in the *data set name field*, and will be given the comment specified in the comment field. If you don’t change the data set name, they’ll all be given the default value of “data set 1”, which isn’t very informative, so it’s a good idea to update the name for each measurement.

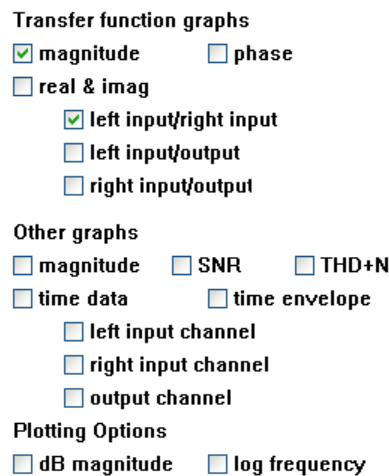
If you want to rename or delete a data set, you can do so in the *Files/Clipboard* section.

Perform measurement

Pressing the *Perform measurement* button will start a frequency response measurement. Once the measurement begins, the button will change to *Stop measurement* which will allow you to terminate the measurement before it finishes. If the SNR levels are too low, the *Stepped Sine Measurement* can sometimes take a long time trying to find the signal buried in the noise. In this case, terminating the measurement and checking the input levels may be the best course of action.

Graphing options

The graphing options appear in the lower left corner of the Ares window.



The options are divided into three sections:

Transfer function graphs
Other graphs
and *Plotting options*

Transfer function graphs

The *Transfer function graphs* options allow the user to specify how and which transfer functions are to be plotted. The options are magnitude, phase, and real & imaginary parts. Once the type of graph is selected, which of the three transfer functions are to appear in the graphs can then be selected. The three transfer functions are:

left input / right input
left input / output
right input / output

Other Graphs

The *Other Graphs* options allow the user to select what quantity is to be plotted for the three signals (left input, right input, and output). These options are magnitude, SNR, THD+N, time data, and a time envelope.

magnitude

For stepped sine and chirp measurements, the magnitude graph shows the RMS level of the sine wave at the specific frequency. For WAV file measurements, the magnitude plot shows the RMS level in the octave frequency band. Usually a magnitude spectrum of a wide band signal such as a WAV file will be plotted as a power spectrum in units of V/ $\sqrt{\text{Hz}}$, but Ares plots the magnitude as a linear Volts (or Pascals, or DFS, depending on the appropriate units). The total RMS level of the WAV file will be the square root of the sum of the individual voltage values squared, that is:

if $V(i)$ is the voltage of the WAV magnitude curve at frequency i , then

$$\text{RMS of WAV file signal} = \sqrt{\sum_{i \text{ all freq}} V(i)^2}$$

where i is an index that ranges over all frequency values.

SNR

The SNR (signal to noise ratio) is displayed in units of dB and is computed according to the formula

$$\text{SNR} = 20 \log_{10} \left(\sqrt{\frac{\text{fundamental power}}{\text{total power}}} \right)$$

SNR is only available for stepped sine and chirp measurements. It is not computed for WAV file measurements.

THD+N

THD+N (total harmonic distortion plus noise) uses the formula

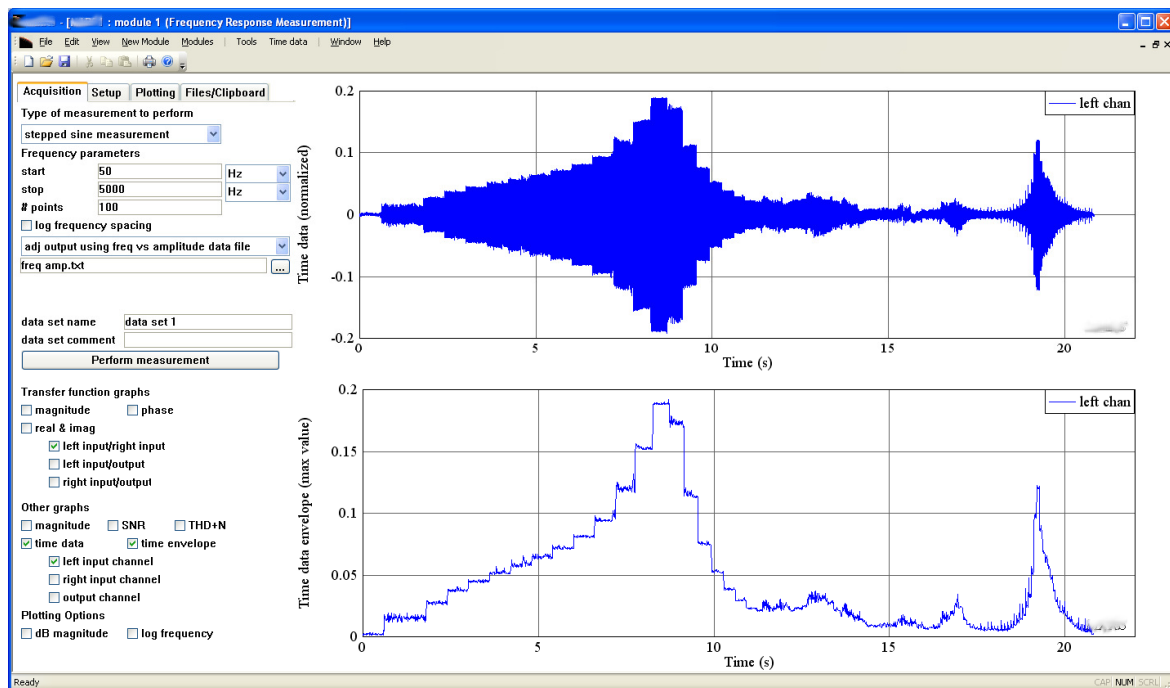
$$\text{THD} + \text{N} = \frac{\sqrt{\text{total power} - \text{fundamental power}}}{\sqrt{\text{total power}}} \times 100\%$$

Only the THD+N from the stepped sine measurement should be used. The chirp measurement attempts to return a THD+N estimate, but it presently isn't a good estimate. THD+N is not computed for WAV file measurements.

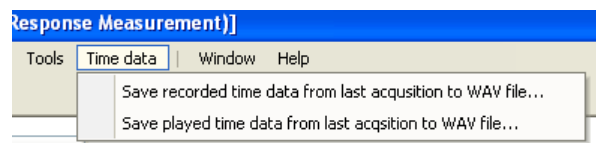
time data and time envelope

The time data graphs show the normalized (0-1) digital waveforms sent to the playback sound card device (for the output channel), or read from the recording sound card device (for the left and right input channels). These represent the exact time domain signals that Ares sent or recorded, and so it is important to verify that they generally look like good clean signals that don't have excessive noise or are clipping the sound card inputs. The time envelope graphs plot a maximum absolute value of the time domain signal taken over small windowed time sections. Since the time data graphs plot EVERY data point in the time domain signals, Ares can become very slow when it draws them. The envelope graphs will have a fraction of the data points as the full time data graphs, and so the envelope graphs plot much quicker.

These graphs are shown for the left input channel for a stepped sine measurement. Note that each of the stepped sine frequencies is clearly visible, but there is a small amount of noise on the signals. The input level is well scaled, reaching a peak value of 0.2 (of the 1.0 maximum digital value).



The time data played out and recorded with the sound card inputs can be exported to a WAV file using the *Time data* menu as shown below. This allows the waveforms to be examined more closely with a WAV dedicated file editor.



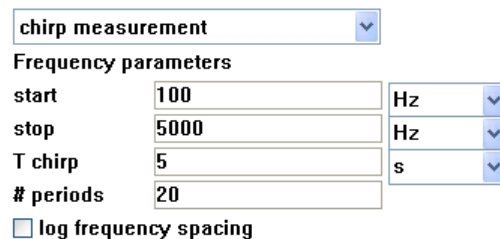
Plotting Options

There are two plotting options, *dB magnitude* and *log frequency*. These will only be applied to the appropriate graphs and the legends will indicate that the value is in dB.

Chirp Measurement

The chirp measurement plays a swept sine signal out of the playback device. This measurement is much quicker than a stepped sine measurement and will likely be the preferred measurement for systems that don't require a WAV file measurement. (WAV file measurements will likely use real speech signals that are required for digital telephony systems.)

The parameters for the chirp measurement are shown below. Rather than specifying a number of frequency points, the chirp measurement requires the length of time the chirp is to last and the number of sine wave periods that are to be averaged to generate a single frequency measurement.



The image shows a software interface for configuring a chirp measurement. At the top, there is a dropdown menu set to 'chirp measurement'. Below it, the section 'Frequency parameters' contains four input fields: 'start' with the value '100', 'stop' with '5000', 'T chirp' with '5', and '# periods' with '20'. To the right of these fields are three unit dropdown menus: 'Hz' for start, 'Hz' for stop, and 's' for T chirp. At the bottom, there is a checkbox labeled 'log frequency spacing' which is currently unchecked.

chirp measurement			
Frequency parameters			
start	100	Hz	
stop	5000	Hz	
T chirp	5	s	
# periods	20		
☐ log frequency spacing			

The *T chirp* parameter specifies how long the chirp signal will be. The longer the chirp signal, the more frequencies that will be resolved for the measurement.

The *# periods* parameter specifies how many sine wave periods are used to generate a single frequency.

For the parameters shown above, Ares will generate 641 discrete frequencies for the chirp measurement in $1/5^{\text{th}}$ the time it would take the stepped sine measurement to obtain 100 frequency points.

The chirp measurement has all the same features as the stepped sine measurement, but it doesn't provide as good of a THD+N measurement. Stepped sine should be used to obtain accurate THD+N measurements. Otherwise, a chirp measurement will likely be preferred.

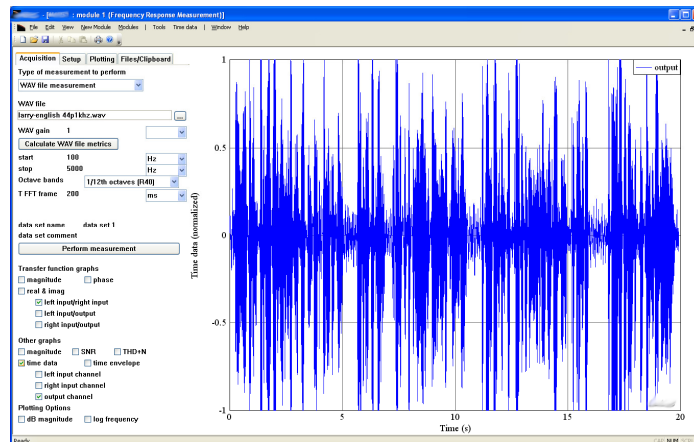
The *log frequency spacing* option will cause the low frequency portion of the chirp signal to last longer, giving greater low frequency resolution than what can be achieved with a linear chirp signal. For most cases, a log frequency spacing will likely be preferred.

WAV file measurement

A WAV file measurement uses a Windows audio WAV file as the output signal from the playback sound card device. It is intended to be used for measuring the transfer function of telephony systems that are designed to efficiently reproduce speech signals.

It is important for this WAV file to be a speech signal because the input and output signals are time aligned by using the WAV file envelope. The peaks and valleys of a real speech are critical for achieving good time alignment of these signals. If the WAV file doesn't have significant amplitude modulation, the time alignment won't work properly and the measurement's accuracy will suffer. (Pink noise or a constant amplitude chirp signal do not have sufficient amplitude modulations and may not work well with this measurement technique.)

The screen capture below shows the time domain output signal for a WAV file that contained speech. As can be seen, there is significant amplitude modulation that will allow for an accurate time alignment of the signals.



The WAV file measurement only uses power spectrums of the output and input signals. The advantage of this is that it's largely unaffected by any nonlinear processing features of modern speech vocoders. The disadvantage is that this measurement technique doesn't generate complex transfer functions, SNR or THD+N measurements.

WAV file measurement

WAV file
larry-english 44p1khz.wav

WAV gain 1

Calculate WAV file metrics

start 100 Hz

stop 5000 Hz

Octave bands 1/12th octaves (R40)

T FFT frame 200 ms

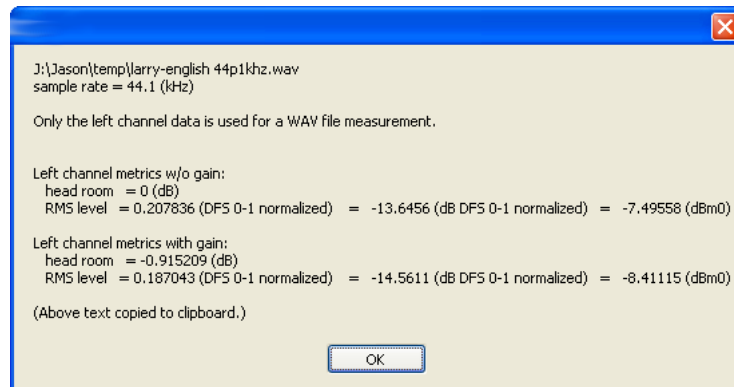
WAV gain

The WAV gain is used to change the WAV file level that's played out of the playback soundcard device by multiplying every sample in the WAV file by this value. If the gain causes the samples to clip (i.e. exceed the digital output range), you'll be alerted to that condition after the WAV measurement is completed. To see how much gain the WAV file can be given without clipping it during playback, use the *Calculate WAV file metrics* button and observe the "head room" without gain. This gives the maximum gain before clipping.

Note that even if the WAV file output is clipped, Ares will use the clipped waveform to compute the necessary FFT spectrums, so the measurement results will still be accurate.

Calculate WAV file metrics

Pressing this button will cause the WAV file metrics to be computed and displayed in a dialog box. An example is shown below. These metrics are computed without and with the *WAV gain* applied to the WAV file data.



head room this is the maximum gain that can be applied to the WAV file without clipping
RMS level the RMS level of the WAV file data

start and stop frequencies

The WAV file measurement uses a series of short time FFT's to process the output and input signals. The start and stop frequencies specify the lower and upper bound of the FFT results that are kept. The frequencies outside of this range are simply discarded.

Octave bands

The short time FFT power spectrums are summed into octave bands to produce the spectrum data. 1/3rd (R10) to 1/24th (R80) bands are supported.

T FFT frame

This parameter specifies the minimum time frame to use for the short time FFTs. The FFTs will be rounded up to the closest power of two samples. Due to this power of two rounding up, the actually FFT time used will likely be larger than that specified by *T FFT frame*.

The tables below give a short list of the FFT time and # of FFT points for four common sample rates. The *T FFT frame* time will be rounded up to these values. The sample rate that's used is taken from the WAV file. For example, a 44.1 kHz sample rate WAV file, the 200 ms FFT frame time will be rounded up to 372 ms and will result in an FFT size of 16384 points.

sample rate	time (s)	# FFT points	# FFT points
		8 kHz	16 kHz
	0.064	512	1024
	0.128	1024	2048
	0.256	2048	4096
	0.512	4096	8192
	1.024	8192	16384
	2.048	16384	32768

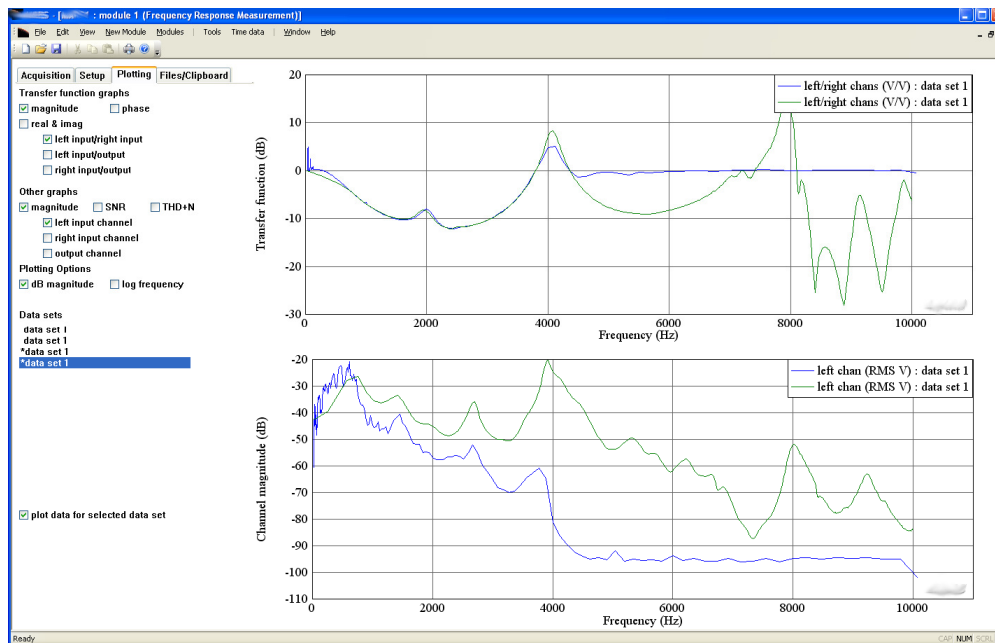
time (s)	# FFT points	# FFT points
	22.05 kHz	44.1 kHz
0.092	2048	4096
0.186	4096	8192
0.372	8192	16384
0.743	16384	32768
1.486	32768	65536
2.972	65536	131072

The entire WAV file is played once. Using a shorter WAV file will result in a shorter measurement time, but less averaging. The number of short time FFT averages will roughly be the length of the WAV file in seconds, divided by the length of time of each FFT frame.

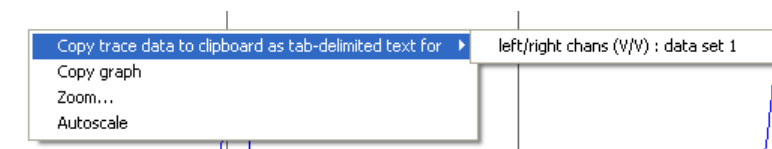
Plotting Controls

The *Plotting* tab displays a list of all of the data sets in the current module and allows for their data to be plotted. To plot the data for a given set, click on the set name in the *Data set* list and then select the *plot data for selected data set* check box that appears below the data set list. Uncheck this selection to stop plotting the data for that set.

The plotting controls are similar to what was already covered in the *Acquisition* section. The time data options are no longer available because the time data is not stored in the data sets. Time data is only available immediately after a successful measurement in the *Acquisition* section.



By right clicking on a graph, a menu will pop up with some useful features.



Copy trace data to clipboard as tab-delimited text for

All of the curve will appear in the sub-menu listed by their legend text. If one of the curve names is selected, the data for that curve will be copied to the clipboard in text format. It can then be pasted into a text editor, Excel, or any other text based program.

Copy graph

This option will cause the graph to be copied to the clipboard in a metafile format. It can then be pasted into applications that support graphical displays, such as Word, PowerPoint, Excel, etc.

Zoom...

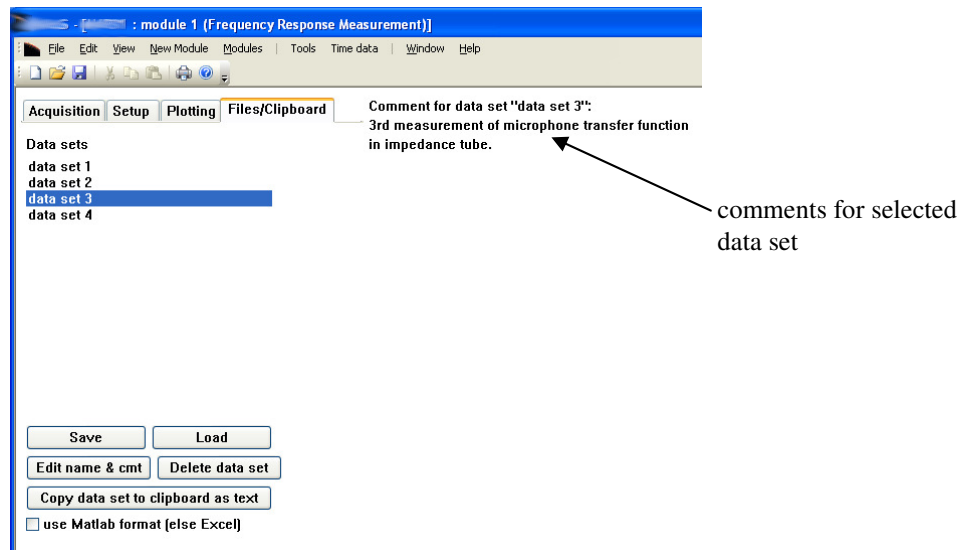
After the *Zoom...* option is selected, the mouse's left button can be used to click and drag a zoom box over the graph data. After the left button is released, the area in the zoom box will be the new display area for the graph.

Autoscale

This option allows you to zoom out and autoscale all of the graph data. It will return the graph to its default x and y axis scales.

Files/Clipboard controls

The *Files/Clipboard* controls allow for individual data sets to be saved to or loaded from *.ARES_FR files. Features to edit the name and comment, or delete the data set are also available. When the data set is selected in the *Data sets* list, the comments for that data set are shown to the right of the controls.



The *Copy data set to clipboard as text* button is a convenient way to export all of a data set's data, including the name and comment fields, to the clipboard. By default, the values are exported as tab delimited values which can easily be pasted into Excel. A format convenient for Matlab is also available by selecting the *use Matlab format (else Excel)* option.

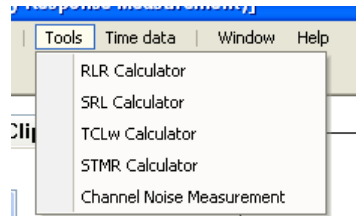
Note that using the right click menu on the graphs to copy the data for a single curve is a good alternative to using this feature to export all of the data set's data. Using the right click menu allows you to just export the data you're interested in, as opposed to the entire data set.

When the *Save* or *Save as* commands in the main *File* menu are used, it saves all of the modules to a single *.ARES file, including all of the data in the modules. So rather than exporting or importing data sets one at a time, all of the data from the Frequency Response Module can be saved to one *.ARES file. Saving all of the data to a single *.ARES file is the preferred method for saving data.

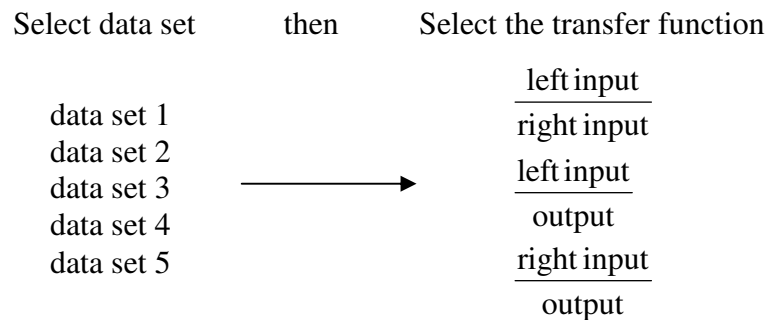
However, saving too much data in a single file is an unwise practice because if that file gets accidentally deleted or corrupted, you could lose a day's or week's worth of work.

Tools menu

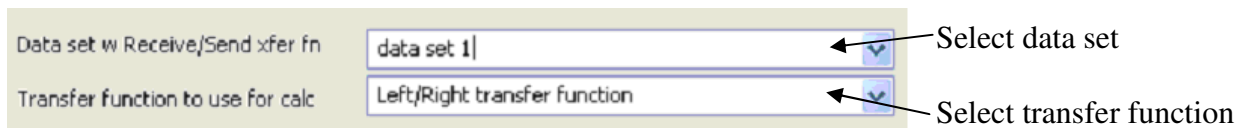
The *Tools* menu has a series of utilities for computing metrics from the data sets. These are mostly focused on telephony metrics such as RLR (Receive Loudness Rating) and SLR (Send Loudness Rating).



The tools require that you specify the transfer function from the data sets for the analysis. Since each data set holds three transfer functions, specifying a transfer set requires two list boxes, one to specify the data set, and the second to specify the transfer function.



The two list boxes for this selection are shown below for the RLR Calculator.



This “data set transfer function” method is used throughout the tools.

Some terminology for the tools:

DUT	Device Under Test
Uplink	The audio signal sent from the DUT to the network.
Downlink	The audio signal sent from the network to the DUT.
Network	The system that the DUT is communicating with, could be a closed system created with a PC computer and a network simulator.
Side Tone	The uplink signal that’s played back out the DUT’s earpiece speaker to provide the user some feedback as to the quality of the uplink audio.
Near or Far End	A telephone conversation consists of a “near end” talker and a “far end” talker. The near end consists of the phone and user that you’re involved with.
B&K	Brüel & Kjær, world leading acoustic testing equipment manufacturer
HATS	Head And Torso Simulator such as a B&K Type 4128

ERP
DRP

Ear Reference Point, a theoretical location at the outer surface of an ear's pinna
Drum Reference Point, the ear drum location in an ear

RLR Calculator

The RLR calculator allows for the Receiver Loudness Rating to be computed from a transfer function measurement. It consists of a dialog box where you specify the transfer function that's to be used for the calculation, the type of correction to be applied for various artificial ears typically used in such measurements, and whether or not the calculation should be done for a narrow band or wide band telephony network. (As of this writing, almost all telephony networks are narrow band. Internet communications services such as Skype are wideband.)

RLR Calculator

Compute RLR(Receive Loudness Rating) based on Receive/Send sensitivity transfer function as specified in the ITU P.79 specification.

This transfer function will be either Pa/V for an analog system, or Pa/DFS for a digital system.

The pressure is that produced by the downlink audio measured with an ear simulator and corrected back to ERP.

The voltage or digital level is the signal sent to the telephony system to produce the downlink audio.

For normal telephony bandwidth, the data should at least cover 176 Hz to 4590 Hz.
For wideband telephony bandwidth, the data should at least cover 88 Hz to 9000 Hz.

Data set w Receive/Send xfer fn: data set 1

Transfer function to use for calc: Left/Right transfer function

Correction: Type 3.3 B&K HATS Simulator DRP to ERP

☐ wide band audio

Calculate RLR RLR: 0

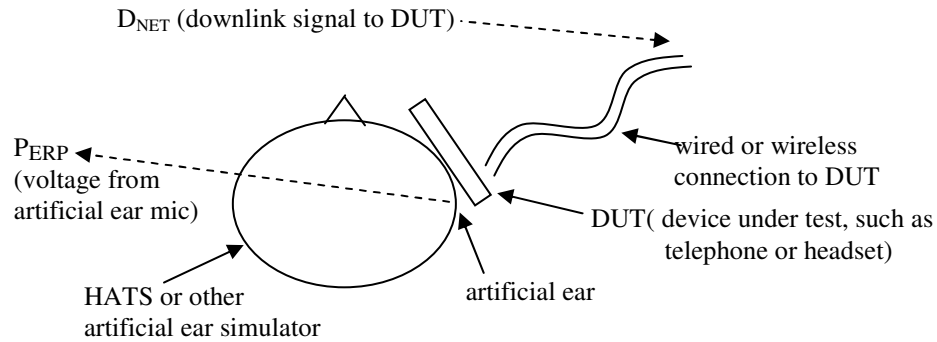
Data set comment:

Append RLR to data set comment Close

The RLR calculator uses the specification given in ITU P.79. To compute RLR, the transfer function H should be the pressure response from an artificial ear's Ear Reference Point P_{ERP} over the signal from the telephone network D_{NET} . Usually the signal from the network will be a digital value, but a voltage signal is also possible.

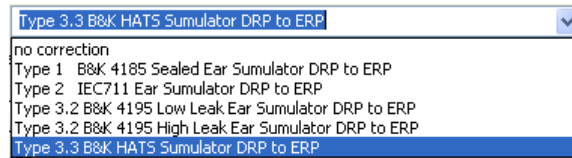
$$H = \frac{P_{ERP}}{D_{NET}}$$

The drawing below shows a rough sketch of what a configuration might look like for an RLR measurement.



For a typical measurement, D_{NET} will likely originate from the PC's playback soundcard and P_{ERP} will be recorded by one of the recording sound card inputs. If the left channel is used, then the *Transfer function to use for calc* in the dialog box would be set to *Left/Output transfer function*. *Data set w Receive/Send xfer fn* will be the data set that contains the desired transfer function H.

The RLR specification requires that the ERP (Ear Reference Point) pressure be used, but the ear simulators are designed to provide a DRP (Drum Reference Point) pressure, so a DRP to ERP correction is required. The Correction that's chosen depends on the type of artificial ear used. A list of all of the ear simulator corrections supported by Ares is shown below.



Select the type of ear simulator used for the measurement and press the *Calculate RLR* button to compute the RLR value. If the frequency range of the data wasn't wide enough for the calculation, a warning dialog box will be given, but an RLR value will be calculated from what frequency data was available.

The RLR value can be saved in the data set's comment field. The data set's comment field is available for editing at the bottom of the dialog box. To make it easy to add the RLR value, press the *Append RLR to data set comment* button and the RLR value will be appended to the comment field text.

SLR Calculator

The SLR Calculator allows for the Send Loudness Rating to be computed from transfer function measurements. It consists of a dialog box where you specify the transfer function that's to be used for the calculation. It requires that you specify whether or not the measurement uses an intermediate MRP mic over artificial mouth voltage transfer function, and whether or not the calculation should be done for a narrow band or wide band telephony network. (As of this

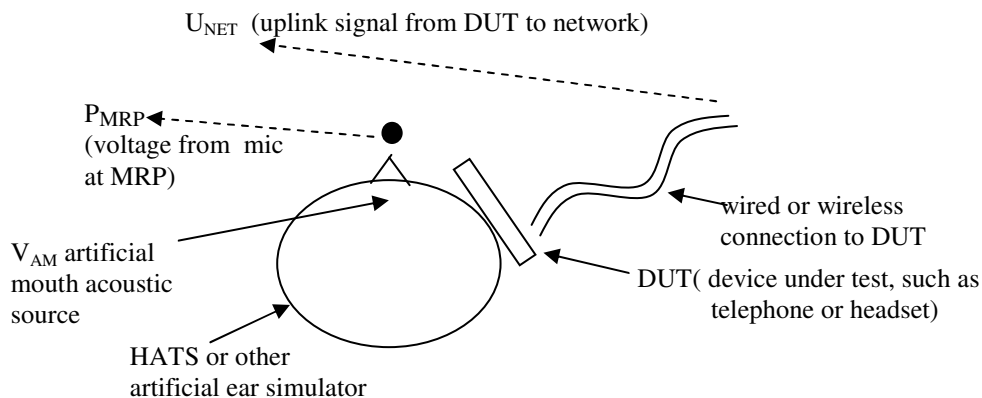
writing, almost all telephony networks are narrow band. Internet communications services such as Skype are wideband.)

The SLR calculator uses the specification given in ITU P.79. To compute SLR, the transfer function H should be the uplink signal to the telephony network U_{NET} over the microphone signal P_{MRP} recorded at MRP (Microphone Reference Point).

$$H = \frac{U_{NET}}{P_{MRP}}$$

where U_{NET} is a digital signal for digital telephone networks, and a voltage for analog telephone networks.

The drawing below shows a rough sketch of what a configuration might look like for an SLR measurement



If it's possible to measure H directly, then to compute the SLR:

- 1) measure H of U_{NET} / P_{MRP}
- 2) uncheck *use voltage to MRP pressure reference transfer function*
- 3) use the *Data set w Uplink/output xfer fn* and *Uplink/output transfer function* list boxes to select H
- 4) if this is a wide band measurement, check *wide band audio*
- 5) press the *Calculate SLR* button

The result will appear in the *SLR* edit field.

Frequently, H cannot be measured directly because either the MRP microphone cannot be in place while the DUT is in place (i.e. for a large phone that covers the HATS mouth area), or because the uplink signal is digital and it's not possible to place the analog P_{MRP} signal and the digital U_{NET} signal on the inputs of a sound card simultaneously (sound cards are either fully analog or digital, not mixed). In these cases, an intermediate transfer function is measured from the voltage input to the artificial mouth V_{AM} speaker to the MRP microphone P_{MRP} . Call this transfer function H1,

$$H1 = \frac{P_{MRP}}{V_{AM}}$$

Then the transfer function from the artificial mouth voltage V_{AM} to the uplink network signal (DFS_{NET} or V_{NET}) is measured with the DUT in place on the HATS. Call this transfer function H2,

$$H2 = \frac{U_{NET}}{V_{AM}}$$

Then H is simply H2/H1

$$\begin{aligned} H &= \frac{H2}{H1} \\ &= \frac{U_{NET} / V_{AM}}{P_{MRP} / V_{AM}} \\ &= \frac{U_{NET}}{P_{MRP}} \end{aligned}$$

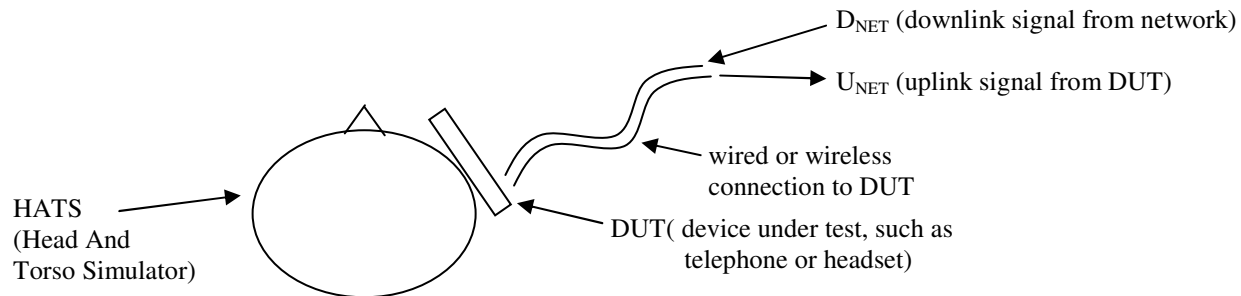
To compute SLR using this approach, follow these steps:

- 1) measure H1 by placing the MRP mic signal on one of the recording soundcard inputs, H1 is then the MRP mic over the output voltage transfer function
- 2) measure H2 by selecting a sound card with the U_{NET} network signal on one of its input channels, H2 is this uplink signal over the output voltage transfer function
- 3) check *use voltage to MRP pressure reference transfer function* option
- 4) use the *Data set w output to MRP xfer fn* and *Output to MRP transfer function* list boxes to select H2
- 5) use the *Data set w Uplink/output xfer fn* and *Uplink/output transfer function* list boxes to select H1
- 6) if this is a wide band measurement, check **wide band audio**
- 5) press the *Calculate SLR* button

The result will appear in the SLR edit field.

TCLw Calculator

The TCLw Calculator computes the weighted Terminal Coupling Loss according to the ITU P.342 specification. This is essentially the uplink echo returned from a telephone device from a downlink signal.



Let H be the transfer function of the uplink signal over the downlink signal, or

$$H = \frac{U_{NET}}{D_{NET}}$$

When the downlink signal D_{NET} is played through the DUT's speaker into the HATS ear, some portion of that audio will reach the DUT's microphone. The DUT will attempt to use an echo canceller to eliminate the D_{NET} signal from the microphone signal, but typically it's not able to eliminate all of it, so some portion D_{NET} will appear on the uplink U_{NET} signal.

With modern echo cancellers, their behavior can be nonlinear and time dependent, so obtaining a single accurate measurement for H can be difficult to achieve with simple test signals such as pure tones and chirps. It may be that only WAV file measurement using a real speech signal will

produce a good estimate for H. It is recommended that you try both the WAV and chirp measurement methods to see which method achieves the most consistent estimate for H.

Some echo cancellers may use a voice gating technique that will simply turn off its uplink signal U_{net} , or replace it with comfort noise. This type of behavior can typically be seen on the recorded time domain data, in which case measuring a consistent value for H may not be possible. Since the WAV measurement technique only uses power spectrums, and not correlated noise, comfort noise will be included in the uplink signal's power spectrum level and may lead to misleading uplink levels.

Also, when using WAV type measurements, it's important for the ambient acoustic environment to be quiet, as any ambient uplink noise will also contribute to the measured uplink power and be considered as an echo signal. (Stepped sine and chirp measurements will largely be immune to such noise.)

The units of U_{net} and D_{net} should both be digital (DFS) (or analog voltage for old telephone systems), so the units of H is essentially dimensionless.

The ITU P.342 specification recommends that TCL_w be measured with a downlink signal level of -15 dBm0. According to the ETSI Digital Enhanced Cordless Telecommunication (DECT) TBR10 requirements document which references the ITU G.711 specification, 3.14 dBm0 corresponds to a full scale sine wave. Ares uses a digital reference where a full scale digital sine wave is -3.01 dB DFS. So

-3.01 dB DFS and 3.14 dBm0 are equivalent levels

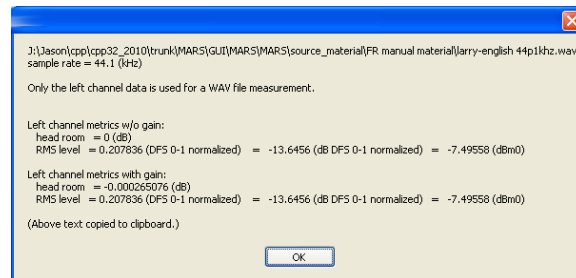
To convert dBm0 units to dB DFS units, subtract 6.15 dB from the dBm0 value

$$\text{dB DFS} = \text{dBm0} - 6.15$$

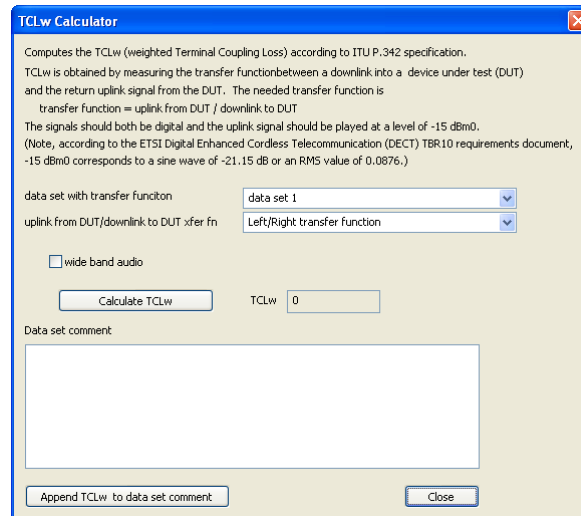
So the -15 dBm0 recommended value is equivalent to -21.15 dB DFS. For stepped sine or chirp measurements, it's straightforward to specify the RMS output level as -15 dBm0 using the units list box.

fixed level
RMS output: -15
units: dBm0, DFS, dB DFS, dBm0
data set name: data set 1

For WAV file measurements, use the *Calculate WAV file metrics* button to determine what the dBm0 level for the WAV file is. As shown below, the “larry-english 44p1khz.wav” file has an RMS level of -7.5 dBm0. To bring this down to the suggested -15 dBm0 level, the WAV gain should be set to $-15 - (-7.5) = -7.5$ dB, or a linear value of 0.42 .



The TCLw Calculator dialog box interface is shown below.



To obtain the TCLw value:

- 1) measure $H = U_{NET}/D_{NET}$
- 2) specify the transfer function H by using the *data set with transfer function* and *uplink from DUT/downlink to DUT xfer fn* list boxes
- 3) check the *wide band audio* option if using wide band data
- 4) press the *Calculate TCLw* button

The TCLw will appear in the edit field to the left of the calculate button.

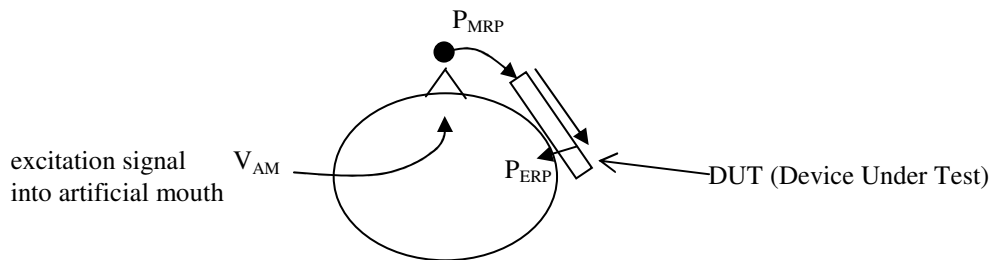
The comment field for the data set that was selected with the *data set with transfer function* list box is shown at the bottom of the dialog box. It can be used to record the TCLw value. Pressing the *Append TCLw to data set comment* button makes it easy to add the value to the comment field.

STMR Calculator

The STMR Calculator computes the Side Tone Masking Rating from measured transfer functions. Many phones play the user's voice back into the phone's earpiece speaker, so the user can essentially hear his or her own voice. A "side tone" is this feedback audio. STMR is a measure of the loudness of this side tone audio.

The side tone audio gives the talker on the "near end" of the conversation an idea of what the listener on the "far end" is hearing. For land line phones, it has been used as a means of informing the near end talker that the "line is still up". With cell phones and Bluetooth headsets, the side tone signal is a good indication of how much wind noise is present on the uplink microphone signal, and will encourage a user to find a less windy location to carry on the conversation.

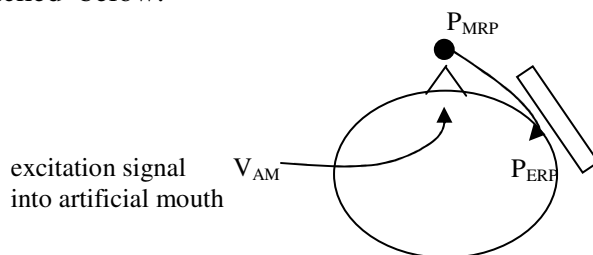
STMR is computed from the transfer function of the pressure P_{MRP} at an MRP microphone to the pressure P_{ERP} at the ERP (ear reference point) of an artificial ear while the artificial mouth is driven, as sketched below.



Let's call the transfer function H_{STMR}

$$H_{STMR} = \frac{P_{ERP}}{P_{MRP}}$$

Critical to the measurement of H_{STMR} is that the pressure P_{MRP} is entirely a product of the HATS' artificial mouth and that the pressure P_{ERP} is produced entirely by the DUT's speaker. However, in practice, there is a significant second path from the MRP to ERP directly through the air, as sketched below.



Let's call the transfer function under this condition H_{off}

$$H_{off} = \frac{P_{ERP}}{P_{MRP}}$$

indicating that this path is measured with the DUT turned off. If the transfer function P_{ERP}/P_{MRP} is measured with the DUT turned on, it will be the sum of H_{STMR} and H_{off} , or

$$H_{on} = H_{STMR} + H_{OFF}$$

Thus the desired transfer function can be computed from the two measurements, H_{on} and H_{off} as

$$H_{STMR} = H_{on} - H_{off}$$

It is possible for H_{off} to be zero. One such case is when the DUT fully occludes the ear, so essentially no sound exterior to the DUT makes it into the artificial ear. An example of this would be a Bluetooth headset with an earpiece that seals into the ear canal, somewhat like an ear plug. Another possibility is when a sealed ear simulator, such as the Type 1 B&K 4185, is fully sealed (probably with putty) to a phone's earpiece. In these cases $H_{off} = 0$, and so a single H_{on} measurement needs to be taken.

(Note that if a Type 1 ear simulator is used, then it should be safe to assume that a HATS isn't being used. In this case, exciting the DUT with an artificial mouth and getting a good MRP pressure measurement becomes a problem. However, if you understand that the main function of HATS is to properly produce the path loss from MRP to the device's microphone, then it should be possible to configure a standalone artificial mouth, MRP microphone, and DUT to simulate this path loss. But the path loss from the MRP to the DUT microphone must be known.)

As was the case with the SLR Calculator, it may not be possible to measure the MRP microphone P_{MRP} signal with the DUT in place. In this situation, the transfer function from the artificial mouth voltage V_{AM} to the MRP microphone pressure P_{MRP} can be used to aid in making the H_{off} and H_{on} measurements.

Call this the P_{MRP}/V_{AM} transfer function $H1$

$$H1 = \frac{P_{MRP}}{V_{AM}}$$

which will be measured with the DUT not in place. This provides a good estimate of the P_{MRP} pressure for a given V_{AM} signal. Since we assumed that the DUT interferes with the MRP microphone when it's in place on the HATS, P_{MRP} can no longer be measured. So $H1$ will have to serve as the P_{MRP} estimate given a V_{AM} signal.

Then, with the DUT in place, the transfer functions from the artificial mouth to the artificial ear measurements are taken with the DUT off and on. Call these transfer functions $H2_{off}$ and $H2_{on}$, or

$$H2_{off} = \frac{P_{ERP}}{V_{AM}} \quad \text{with DUT off}$$

and

$$H2_{on} = \frac{P_{ERP}}{V_{AM}} \quad \text{with DUT on}$$

Then

$$H_{STMR} = \frac{H2_{on} - H2_{off}}{H1}$$

Finally, an artificial ear typically measures the DRP (drum reference point) pressure and not the ERP (ear reference point) pressure. So a correction from DRP to ERP must be applied. Let this correction be C

$$C = \frac{P_{ERP}}{P_{DRP}}$$

If the transfer functions $H2_{on}$ and $H2_{off}$ actually measured P_{DRP} instead of P_{ERP} , C must be applied to the measurement, so the final estimate for H_{STMR} becomes

$$H_{STMR} = \frac{H2_{on} - H2_{off}}{H1} C$$

CRITICAL NOTE: It is vital that the acoustic path from the MRP microphone to the artificial ear not change between H_{on} and H_{off} , or $H2_{on}$ and $H2_{off}$. Moving the DUT even a fraction of a millimeter to turn it on or off could alter the acoustic path and ruin the measurements. It is advised that some means to turn the device on or off without actually touching it be devised. Perhaps a hard wired switch that's several inches away could be used to disable the battery.

The full STMR Calculator dialog box is shown below.

STMR (Side Tone Making Rating) is computed according to the ITU P.79 specification using a transfer function from an MRP mic to the unit generated ERP audio. If the unit doesn't allow an MRP microphone to be in place, an output-to-MRP mic transfer function can be used to predict the MRP signal. If the unit doesn't occlude the ear, a "unit turned off" MRP mic-to-ear mic transfer function can be used to subtract the direct audio path from the "unit turned on" MRP-to-ear mic transfer function, thus correcting for the direct audio path.

For normal telephony bandwidth, the data should at least cover 176 Hz to 4590 Hz.
For wideband telephony bandwidth, the data should at least cover 88 Hz to 9000 Hz.

☐ use output voltage to MRP pressure reference transfer function
Data set w output to MRP xfer fn: Pmrp/Vam
Output to MRP transfer function: Left/Output transfer function

☐ use "unit turned off" transfer function to correct for direct MRP to ear audio path
Data set w unit turned off: Pdrp off / Vam
transfer function: Left/Output transfer function

Data set w unit turned on: Pdrp / Pmrp
transfer function: Left/Right transfer function

correction: Type 3.3 B&K HATS Simulator DRP to ERP

☐ wide band audio

Calculate STMR STMR: 0

Data set comment:

Append STMR to data set comment Close

Example 1 $H_{\text{off}} = 0$ and MRP microphone is in place

Assume we're using a HATS and the DUT fully occludes the ear so $H_{\text{off}}=0$. Also, assume we can measure the P_{MRP} pressure with the DUT in place. Then the H_1 transfer function isn't needed. In this case, we only need to specify the H_{on} transfer function $P_{\text{DRP}}/P_{\text{MRP}}$, but we'll need to specify the correction factor C .

Assuming that the $P_{\text{DRP}}/P_{\text{MRP}}$ was measured with the left and right input channels and stored in a data set named "Pdrp/Pmrp", then the STMR Calculator controls would be set to the following.

☐ use output voltage to MRP pressure reference transfer function
Data set w output to MRP xfer fn: Pmrp/Vam
Output to MRP transfer function: Left/Output transfer function

☐ use "unit turned off" transfer function to correct for direct MRP to ear audio path
Data set w unit turned off: Pdrp off / Vam
transfer function: Left/Output transfer function

Data set w unit turned on: Pdrp / Pmrp
transfer function: Left/Right transfer function

correction: Type 3.3 B&K HATS Simulator DRP to ERP

☐ wide band audio

Note that:

- use output voltage to MRP pressure reference transfer function is unchecked
- use "unit turned off" transfer function to correct for direct MRP to ear audio path is unchecked
- H_{on} is specified using the Data set with unit turned on and transfer function list boxes
- the HATS type 3.3 ear is selected in the correction list box

Finally, it's assumed that we're using narrow band data, so wide band audio is unchecked.

Pressing Calculate STMR will compute the STMR value.

Example 2 H_{off} is non-zero and MRP microphone is in place.

Example 2 will be assumed to be similar to Example 1, except that this time the H_{off} path is non-zero. In this case, the *use “unit turned off” transfer function to correct for direct MRP to ear audio path* option is checked and the $H2_{\text{off}}$ transfer function is *specified in the Data set w unit turned off* and its associated *transfer function* list box.

Assume that the $H2_{\text{off}}$ data resides in a data set with the name “Pdrp / Pmrp off” and the $H2_{\text{on}}$ data resides in a data set with the name “Pdrp / Pmrp on”. Since the MRP and DRP microphone were both analog voltages, they could both be recorded simultaneously with the left and right inputs. The DRP microphone was put onto the left channel, and the MRP microphone was put onto the right channel.

Again, it’s assumed that we’re using narrow band data, so *wide band audio* is unchecked.

The STMR dialog box controls will be set as shown below. Pressing *Calculate STMR* will compute the STMR value.

☒ use "unit turned off" transfer function to correct for direct MRP to ear audio path

Data set w unit turned off: Pdrp / Pmrp off

transfer function: Left/Right transfer function

Data set w unit turned on: Pdrp / Pmrp on

transfer function: Left/Right transfer function

correction: Type 3:3 B&K HATS Simulator DRP to ERP

☐ wide band audio

Example 3 H_{off} is non-zero, and MRP mic is removed for DUT to be in place

In this example, the MRP microphone must be removed in order for the DUT to be put in place on a HATS so the $H1$ transfer function of $P_{\text{MRP}}/V_{\text{AM}}$ must be measured, and then the subsequent measurements $H2_{\text{off}}$ and $H2_{\text{on}}$ will be $P_{\text{DRP}}/V_{\text{AM}}$ with the unit off and on, respectively.

The $H1$ transfer function $P_{\text{MRP}}/V_{\text{AM}}$ is specified by:

- enabling *use output voltage to MRP pressure reference transfer function*
- specify the name of the data set containing the $H1$ in the *Data set w output to MRP xfer fn* list box
- specifying which transfer function in the *Output to MRP transfer function* list box

We'll assume that the DRP artificial ear microphone was placed onto the left input channel, and so the H1 transfer function will be the *Left/Output transfer function*.

The H_{2off} transfer function P_{DRP}/V_{AM} is selected by:

- enabling the *use "unit turned off" transfer function to correct for direct MRP to ear audio path*
- specifying the name of the data set containing H_{2off} in the *Data set unit turned off* list box
- specifying which transfer function in the associated *transfer function* list box

Again, we'll assume that the DRP artificial ear microphone was placed onto the left input channel, and so the H_{2off} transfer function will be the *Left/Output transfer function*.

The H_{2on} transfer function P_{DRP}/V_{AM} is selected by:

- specifying the name of the data set containing H_{2on} in the *Data set w unit turned on* list box
- specifying which transfer function in the associated *transfer function* list box

Again, it's assumed that we're using narrow band data, so *wide band audio* is unchecked.

The STMR dialog box controls will be set as shown below. Pressing *Calculate STMR* will compute the STMR value.

The screenshot shows a dialog box with the following controls:

- ☒ use output voltage to MRP pressure reference transfer function
 - Data set w output to MRP xfer fn: Pmrp/Vam
 - Output to MRP transfer function: Left/Output transfer function
- ☒ use "unit turned off" transfer function to correct for direct MRP to ear audio path
 - Data set w unit turned off: Pdrp / Vam off
 - transfer function: Left/Output transfer function
- Data set w unit turned on: Pdrp / Vam on
- transfer function: Left/Output transfer function
- correction: Type 3.3 B&K HATS Simulator DRP to ERP
- ☐ wide band audio

Channel Noise Measurement

The Channel Noise Measurement tool will perform a measurement of the weighted RMS power on the input channels of the recording soundcard device. This is different from the other tools in that it will record new data and not process previously acquired transfer function data.

Four weighting options are provided

- none just compute a straight RMS value from the sound card inputs
- A standard audio SPL A weighting, typically used for sounds under 60 dB SPL
- C standard audio SPL C weighting, typically used for sounds over 80 dB SPL
- P psophometric weighting, typically used for telephony noise spectra

When the *Perform channel noise measurement button* is pressed, the measurement will be performed, the time domain signal filtered through the appropriate weighting filter, an RMS level will be computed, and the results reported in the *Left channel noise* and *Right channel noise* display area.

The Channel Noise Measurement dialog box interface is shown below. The results are presented in the correct units that the input channels have been assigned in the *Setup* control section. In the example below, the left channel was setup to be a voltage input, while the right channel was setup to be a microphone input. So the left channel units are voltage and dBV, while the right channel units are Pascals and SPL.

Channel Noise Measurement

Measures the RMS channel noise on the Recording device sound card.

When no signal is present, this will measure the idle channel noise on the Recording device sound card channels.

This will alternatively play a WAV file out of the Playback device sound card during the measurement. If a proper WAV file is chosen, the maximum SPL output of the audio device can be computed for safety testing.

This tool doesn't use existing frequency response datasets, but performs a new time domain RMS signal measurement.

☐ Play WAV file out Playback device

WAV file ...

length of time to record channel (seconds)

☐ Apply no filtering

☐ Apply A weighting filter

☐ Apply C weighting filter

☒ Apply psophometric weighting filter

Perform channel noise measurement

Left channel noise

Right channel noise

Close

The amount of time data that's recorded and analyzed is determined by the *length of time to record channel (seconds)* field.

If the *Play WAV file out Playback device* option is selected, you will be able to browse for a WAV file to play out of the Playback soundcard device. The WAV file will be played out both the left and right channels with an unmodified amplitude. The amount of time the playback runs is equal to the amount of time entered into the *length of time to record channel (seconds)* field. If the WAV file is longer than this time, it will be truncated and if it's shorter, the WAV data will be looped until the desired time is achieved.

Playing a WAV file out can be useful for measuring the maximum SPL that can be generated by an audio device.

For more information about Ares and acoustical measurement and modeling tools and services, contact MAE at:

info@MAELLC.COM
support@MAELLC.COM
(678) 234-5079

Or see us at MAELLC.COM.