

Getting Started with SmaartLive: Basic Measurement Setup and Procedures



Paul D. Henderson

This document serves as a starting point for learning to use SIA SmaartLive[®] for basic measurements of audio systems and components. Here, we will discuss the capabilities of SmaartLive and the basic measurement hardware necessary to perform successful measurements. A series of tutorial examples will be presented, which will serve as a hands-on introduction to making measurements with the system.

I. The Primary Measurement Functions of SmaartLive

Fundamentally, SmaartLive is a software-based dual-channel audio analyzer, capable of performing a large number of measurement tasks required by the audio professional. While SmaartLive is not intended to replace critical listening and human experience, the intelligent application of the measurement platform to the task of configuring, troubleshooting, and optimizing systems provides the user with significant advantages. Table 1 outlines the basic measurement modes of SmaartLive, with a synopsis of some of the common uses and capabilities for each.

SmaartLive Mode	Primary Capabilities	Applications
Spectrum	<ul style="list-style-type: none"> • Real-time spectrum analysis • Narrowband and fractional-octave display • Calibration to real-world sound pressure level with SPL metering • Running SPL log and spectrograph functions 	<ul style="list-style-type: none"> • Live source spectrum monitoring • SPL monitoring for live performance • Noise level analysis • Feedback detection
Transfer function	<ul style="list-style-type: none"> • Real-time transfer function analysis • Configurable magnitude <i>and</i> phase display • Narrowband and Fixed-Point-per-Octave analysis • Real-time coherence display 	<ul style="list-style-type: none"> • Transfer function measurements of loudspeakers, equalizers, sound systems • Real-time optimization of systems (incl. equalizers, crossovers, delays, etc.)
Impulse	<ul style="list-style-type: none"> • Impulse response measurement • Linear, log, and ETC display • Automatic estimates of propagation delay times 	<ul style="list-style-type: none"> • Measurement of sound system / room impulse response • Configuration of loudspeaker delays, etc.

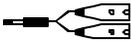
Table 1: *Basic measurement modes in SmaartLive.*

In addition to the capabilities in Table 1, SmaartLive contains an internal *signal generator* that simplifies the measurement process by creating the appropriate excitation signals for each measurement. This eliminates the need for an external device dedicated to producing measurement signals for use with SmaartLive. SmaartLive also includes significant capabilities for controlling external devices, such as loudspeaker processors, equalizers, etc. This useful functionality will not, however, be discussed in this document.

SIA Software Company, Inc. is ***not*** responsible for damage to your equipment resulting from improper use of this product. Be sure that you understand and observe the proper input and output levels, impedances and wiring conventions of all system components before attempting the measurements described in this document.

II. Components of a Basic SmartLive Setup

To use SmartLive effectively, you should have on hand a basic set of measurement equipment. The list below outlines the fundamental components of a SmartLive measurement setup. Details regarding the connection of your equipment for specific measurement tasks will be presented in later sections.

Component	Guidelines
 Measurement Microphone	If you intend to measure anything beyond simple electronic devices, you will need a measurement microphone to acquire acoustical signals. The basic job of the microphone is to convert acoustical pressure at a point into a voltage as accurately as possible, so your microphone should be omnidirectional with a flat frequency response . Most commercially available measurement microphones are based on an <i>electret condenser</i> design, which will require some form of power, either through phantom power from your preamplifier or by an internal battery. You may also desire a microphone calibrator to accurately perform calibrated sound pressure measurements.
 Microphone Preamplifier	To interface with your microphone, you will typically need some form of microphone preamplifier. The preamplifier should possess a low noise floor and sufficient gain for reasonable use. For most measurement microphones, the preamplifier should include phantom power to power the microphone. Note that many users may prefer to use a small mixer, routing device, or even a front-of-house console in place of a dedicated preamplifier. In these cases, make sure to disable all channel processing before use (equalization, dynamics processing, etc.).
 Sound Card	For full functionality, SmartLive requires a compatible sound card with at least two independent line-level input channels (usually in the form of a single stereo input) and a line-level output . Some notebook computers with built-in sound capabilities only offer a single channel (mono) input, so make certain your system meets this criterion. Without a stereo input, you will be unable to utilize the <i>transfer function</i> and <i>impulse</i> measurement capabilities. Note that external hardware solutions are available, some of which combine high-quality A/D and D/A converters with built-in microphone preamplifiers. These may prove to be maximally convenient for field use, are readily available with USB, PCMCIA, and Firewire interfaces. Regardless of the input device you select, it <i>must</i> use a Windows®-compatible Wave audio device driver . Other device driver types, including ASIO, are not currently supported by SmartLive.
 Computer with SIA-SmartLive™	The computer should adhere to at least the minimum requirements for running SmartLive, which are available in the <i>SmartLive User Guide</i> or by accessing the SIA Software website at http://www.siasoft.com . For portable field operation, a notebook computer is most convenient.
 Cabling and Interconnections	You should have on hand the appropriate cables for connecting your measurement system and interfacing with the equipment that you are measuring. Use only professional-quality cables, avoiding inferior adapters and consumer-grade interconnections. If your computer sound card uses 3-conductor 1/8-inch stereo phone connectors for interface, you may obtain <i>breakout</i> cables that allow you to convert this interface to 1/4-inch phone or XLR connectors.

III. Getting Signals into SmartLive

Now that you have assembled the equipment necessary to operate SmartLive, we will discuss the process of setting up the system to recognize your hardware and adjusting the signal levels through the system. Make sure your sound card is online (if external, connect it) and start SmartLive. If you are using an external audio device, *do not* disconnect it while SmartLive is running. To select the proper sound card for use by the system, click **Options ? Devices** on the menu, or press **Alt+A**. The window in Figure 1 will appear. Select your sound card input device in the *Wave In* drop-down box, and do the same for the output device in the *Wave Out* box. If your device supports input or output resolution higher than 16 bits, select the appropriate values in the *Bits Per Sample* boxes. Now, you may connect devices to the outputs and inputs of your soundcard, and the signals will be correctly handled by SmartLive.

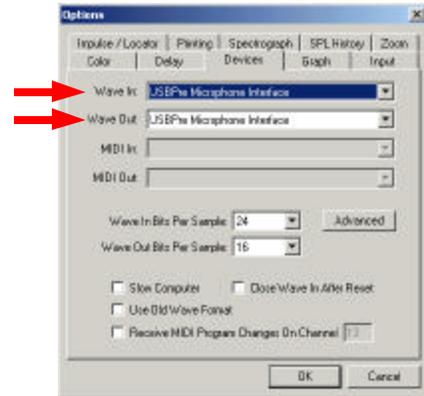


Figure 1: Select your sound card as the input and output device.

Many soundcards use internal mixing circuitry to mix many audio streams together to the master output, or to select and/or mix the signals together that will be seen by the sound card inputs. You may need to configure these options for your card to enable the *line-level input* and also to enable the *wave output*. These options may be set by launching the Windows® mixer application, seen in Figure 2. You may launch the mixer by pressing **Alt+V** on the keyboard in SmartLive. See the *SmartLive User Guide* for more detailed information.

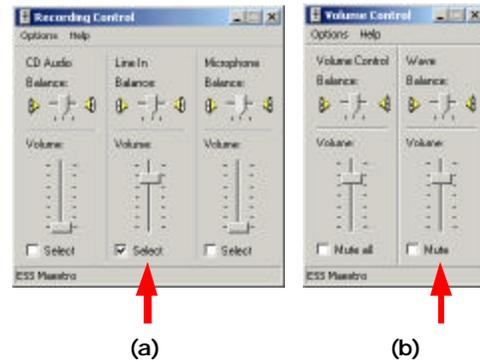


Figure 2: Configure the input (a) and output (b) mixers for your device if necessary.

A note about levels: the signal generator level is controlled by the level setting spinner on the generator, which defaults to a low level to prevent equipment damage at first use. This, in combination with output levels and mixer settings for your device, varies the signal level seen at the sound card output. For most measurements where the signal generator is required, you will desire to adjust the overall generator level to a high enough level to avoid noise in the measurement, but not so high a level as to cause damage or discomfort.

The input levels, however, must be carefully adjusted to eliminate signal *clipping* on the sound card A/D converters while maintaining a high enough signal level to minimize extraneous noise. At any time SmartLive is running, the input level meters in Figure 3 are active, which indicate the *peak* input levels seen by the A/D converters in the sound card. If the levels are too low (Figure 3a), noise from the sound card and other analog devices may compromise the measurement. If the levels are too high, the clipping indicators will light (Figure 3c), and you must reduce the input level to perform meaningful measurements. For most measurements, maintain a nominal input level between -12 and -6 dB.

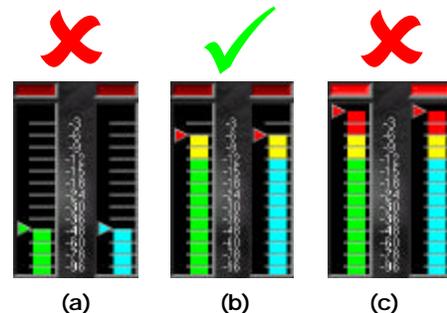


Figure 3: Make sure input levels are neither too low (a) nor too high (c). Input levels should usually be around -12 to -6 dB for basic measurements (b).

IV. Measurement Examples

Now, we will explore some examples of basic measurements using SmartLive. The following pages will introduce you to the fundamentals of making *Spectrum* (RTA), *Transfer Function*, and *Impulse Response* measurements. The examples presented here are arranged in order of complexity; the later examples build on concepts presented in preceding sections. We recommend that you proceed through these exercises in order. Each example will present the basic hardware configuration, as well as sample measurement results. Keep in mind that your data may appear different than the data presented here; the sample results are simply representative of what you might see.

Example Application 1

SmartLive as a Real-Time Spectrum Analyzer (RTA)

The most basic functionality of SmartLive lies in its *Spectrum* mode, which enables two channels of real-time spectrum analysis. In this mode, SmartLive contains functions similar to a hardware RTA (real-time analyzer), where the incoming signals are decomposed into frequency components and displayed dynamically. By default, SmartLive displays the two channels as a real-time bar graph of *energy vs. frequency*, with each bar representing a *band* of energy $1/12^{\text{th}}$ -octave wide, although many other displays are possible.

Connecting the System

To make use of this mode, connect any compatible audio signal to at least one input of your sound card. In RTA mode, you may monitor two channels simultaneously, so you could connect a measurement microphone to one channel and the output of a mixing console to the other, among other possibilities.

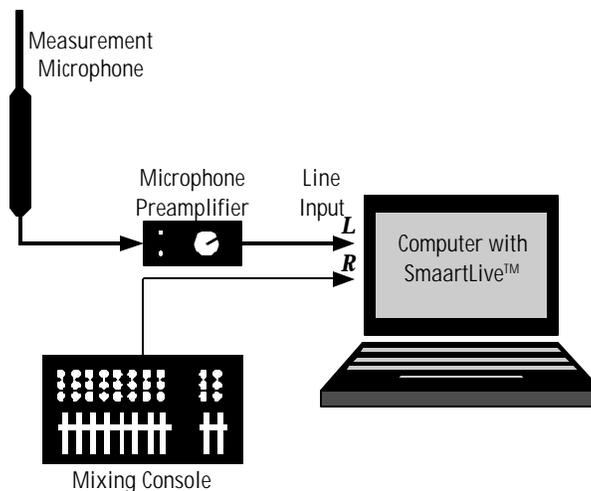
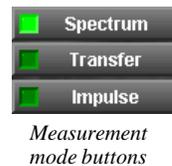


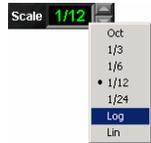
Figure 4: Example hardware configuration for RTA measurement.

Figure 4 shows the hardware configuration for this example. Any line-level signal may be used as an input source, although it is somewhat educational to perform this introductory measurement with a live microphone connected through a preamplifier. Now, launch SmartLive, which should default into *Spectrum* mode. At any time, you may change the current measurement mode by clicking one of the mode buttons. Press the button to begin the real-time measurement.



You should now see a live banded spectrum of the input signals as in Figure 5 (both channels are visible by default). If you are using a live microphone, the spectrum display will respond to any noise in the room, and whistling near the microphone will drive the bands noticeably upward near the frequency of your whistle.

You may experiment with different views of the incoming data by selecting a different data scale with the *Scale* spinner; data may be presented in 1/1, 1/3, 1/6, 1/12, and 1/24-octave bands or as a narrowband power spectrum on either a logarithmic or linear frequency axis.



Also configurable are the basic Fast Fourier Transform (FFT) parameters, averaging, and weighting. Averaging may be used to vary the time behavior of the spectrum display, allowing you to look at either the instantaneous behavior of the signal or its long-term spectral content. Averaging may be configured with various options, including *Slow*, *Fast*, and *Exp*, which represent exponentially decaying behavior, *Inf*, which maintains an evenly-weighted average of all data since initialization, and numeric values between 1 and 128, which evenly weight the last n measurements. You may “reseed” the averaging buffers at any time by pressing **V** on the keyboard.

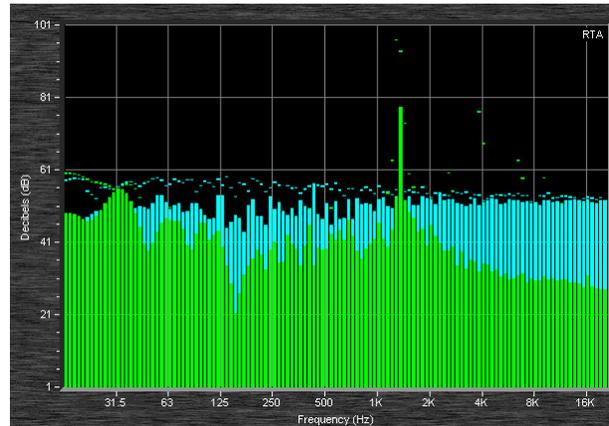


Figure 5: Default live RTA spectrum display (1/12th-octave).

The spectrum mode contains many other useful features, including absolute calibration, integrated SPL metering and logging, and real-time spectrogram functions (Figure 6). Note that these functions will display data for the active input channel only. The active channel may be assigned by clicking on the associated input level meter. Please see the included documentation for more information on the use of these more advanced functions.

The *Spectrum* mode is very useful for identifying feedback frequencies, looking at room noise, studying the spectral content of musical material, and has many other uses. Historically, many have employed RTA methods for measuring the frequency response of a system and performing equalization; we do *not* recommend using this technique, as the transfer function measurement capability of SmaartLive is a far more useful and accurate tool for this task. The *Spectrum* mode is inherently unable to distinguish direct sound from reflective energy and to discriminate between the excitation signal and uncorrelated noise, which limits its usefulness in sound system response optimization tasks.

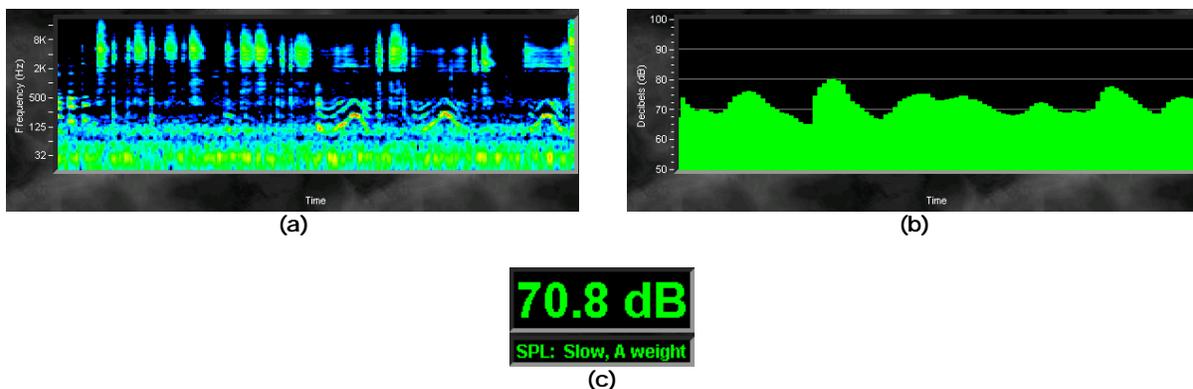


Figure 6: Sample advanced Spectrum-mode measurements; (a) Spectrogram, (b) SPL time history, (c) Real-time calibrated SPL.

Example Application 2

Measuring an Analog Equalizer

In this example, we will introduce the use of SmaartLive's real-time *Transfer Function* measurement capability to measure the frequency response of an analog equalizer. To perform this exercise, you will need an equalizer, crossover, or some other filtering signal processor. An analog device is best for this example; digital devices include some throughput delay from input to output that must be found and compensated for, using SmaartLive's *internal signal delay* (we will discuss this in subsequent examples).

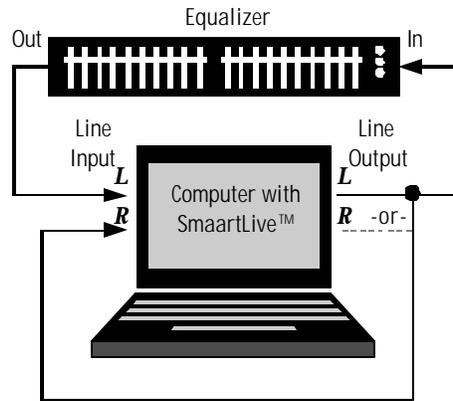


Figure 7: Hardware configuration for measuring an equalizer.

Connecting the System

Connect the system as shown in Figure 7, with the sound card output driving both the input of the equalizer and the *right input* of the sound card (called the *reference* input). The output of the equalizer is brought into the computer through the *left input* (called the *measurement* input). The internal signal generator in SmaartLive will be used to excite the equalizer and measure its response. By using this connection scheme, SmaartLive compares the exact representation of the generator output to the returning signal from the equalizer, effectively canceling any imperfections introduced by the sound card.

Adjusting Signal Levels

Start the SmaartLive generator by clicking the *Generator On* button; the system will now begin generating random pink noise that will appear on the sound card output.



The generator defaults to a low level (-36 dB), so use the generator level spinner to increase the output to a reasonable level. The level should be high enough that the dynamic range of the system and equalizer is adequately utilized; try for a generator level of around -6 dB, as a rule of thumb target, to optimize the performance of the internal generator.

If the analyzer is not running, click **ON** to start it. Now, adjust the input controls to bring the input signal to a reasonable level (as in Figure 3). If the equalizer is bypassed or its controls set to 0 dB, the transfer function display should be an approximately flat line. If not, it is likely that there is some gain error in the system or the device itself contains some gain or attenuation factor. You may adjust the *measurement* channel gain independently to correct for this, or use the *dB +/-* spinner to adjust the “zero” level of the displayed transfer function.



Performing the Transfer Function Measurement

SmaartLive's *Transfer Function* mode measures a system's frequency response by comparing its input signal to its output signal. This measurement shows the difference between those two signals in both magnitude and phase and represents the processing behavior of the system as a function of frequency. Gain and loss show up as deviation from the center 0 dB line on the magnitude trace.

By adjusting filter settings on the equalizer, you should be able to see the changes being made in the frequency domain on the SmaartLive display. If attenuation (or “cut”) on the equalizer shows up as a peak on the display, it is likely that you have inadvertently reversed the input signals. You can either physically swap the input cables or press the  button to reverse the signals and obtain the proper display.

SmaartLive defaults to the FPPO (*Fixed-Point per Octave*) transfer function mode, which provides you with transfer function measurement points distributed equally on a logarithmic frequency scale by varying the FFT length at different frequencies. You may wish to experiment with different fixed-width FFT parameters, sampling rates, and excitation signals to see the effect of the various parameters. You may also press the  button to see the transfer function phase (in addition to magnitude) as a function of frequency. Figure 8 shows an example measurement of a single parametric equalizer filter, including the phase plot.

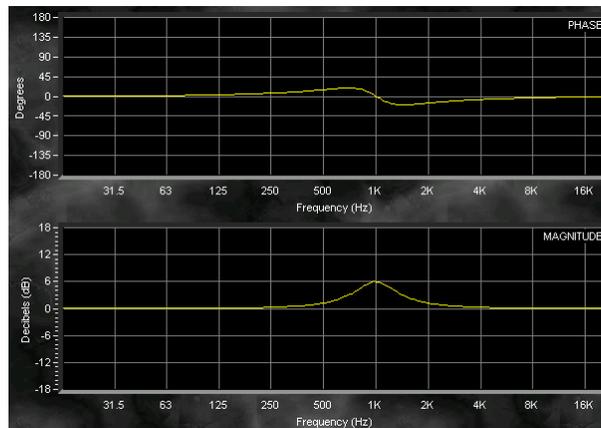


Figure 8: Sample measurement of an analog parametric equalizer filter.

Note that, for this measurement, we have not included any compensation for delay through the equalizer, since the delay through almost any analog equalizer will be insignificant relative to the length of the FFTs used in the calculations. When measuring a digital device, loudspeaker, or almost any electroacoustic system, we must first measure the propagation delay time and use SmaartLive's internal alignment delay to compensate for any delay in the external system. This will be covered in Example 3, “Measuring a Loudspeaker”.

Example Application 3

Measuring a Loudspeaker

In this example, we will use both the *Impulse* and *Transfer Function* modes in SmaartLive to measure the behavior of a loudspeaker in a room. To perform this exercise, you will need a loudspeaker and power amplifier (or powered loudspeaker) in addition to your SmaartLive measurement system.

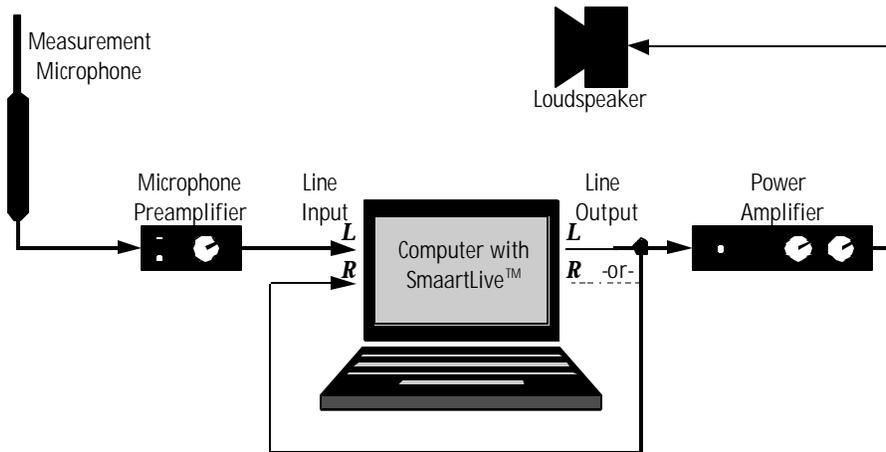


Figure 9: Hardware configuration for measuring a loudspeaker.

Connecting the System

Connect the system as shown in Figure 9 above. The output of the sound card drives the power amplifier and loudspeaker as well as the *right* input (the *reference* signal) of the computer as in the preceding example. For this measurement, however, a measurement microphone is used as the *measurement signal*, which is brought into the *left* sound card input via a microphone preamplifier.

You will want to position the measurement microphone at a nominal distance from the loudspeaker, for example, 1m. The farther the microphone is from the loudspeaker, the more difficult it is to separate the direct sound of the loudspeaker from the influence of reflections in the room.

Adjusting Signal Levels

We must now adjust the signal levels to obtain a reasonable gain structure through the system. Enter SmaartLive's *Spectrum* mode by clicking  and turn the analyzer on. As before, turn on the internal signal generator and adjust its level until the sound level from the loudspeaker is appreciably higher than the ambient noise in the room. You may wish to start with a low power amplifier gain and then slowly increase the level to prevent unexpectedly loud signals from reaching your ears. Next, you may adjust the input levels on the sound card and the microphone preamplifier gain to achieve a proper input level, following the guidelines in Figure 3. For best results, you will want to match the levels at the *reference* and *measurement* inputs as closely as possible, so adjust the preamplifier gain to achieve this.

Performing and Interpreting an Impulse Response Measurement

Now, we will perform a basic impulse response measurement of the loudspeaker in the room. Click the  button to enter the *Impulse* mode; SmaartLive will automatically measure an impulse response of the system and display it in the graph window. You may make another impulse response measurement at any time by pressing the  button. An example of what you might see is shown in Figure 10.

The impulse response in Figure 10 is quite typical of what you might see when measuring a loudspeaker in any non-anechoic room (any room with reflective surfaces). The impulse response view shows a time history of energy arriving at the microphone, and is very useful for understanding exactly what you are measuring. Figure 10 shows the *Log* magnitude view, which indicates the magnitude of the impulse response in dB. SmaartLive can also display the impulse response in *Linear* units (which preserves polarity information) and as an *ETC* (Energy-Time Curve), which extracts the decay envelope from the impulse response, displayed in dB. The large peak at the beginning of the plot in Figure 10 is due to the direct sound from the loudspeaker, which, in this case, is the component that interests us. SmaartLive automatically detects the time and magnitude of this peak, which, in this case, is arriving with a *time delay* of 2.29 ms. This delay is due to the propagation time from the loudspeaker to the microphone through the air over distance of approximately 2½ feet. We will use this concept in correcting for the propagation delay when we perform a transfer function measurement. The other energy shown in impulse response is due to reflections in the room and noise in the measurement. The noise floor can be seen to have a constant average level; the accuracy of your measurements is dependent upon an adequate *signal-to-noise* ratio between the direct sound and this noise level.

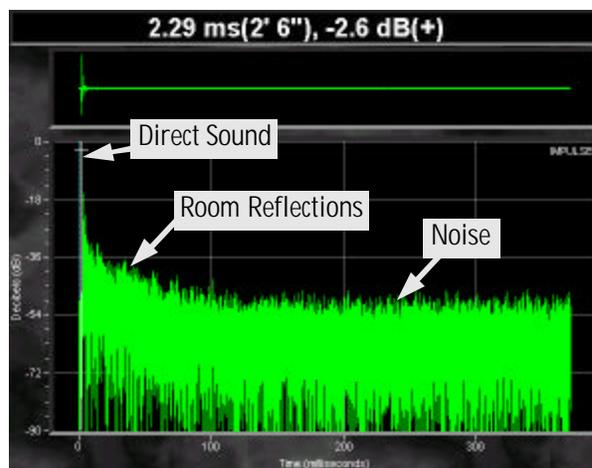
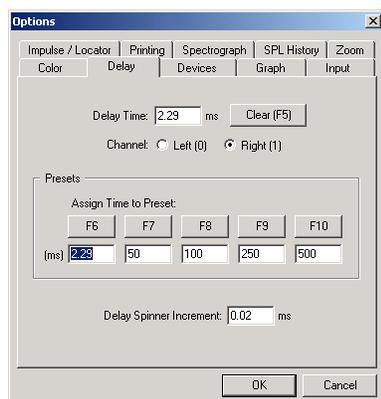
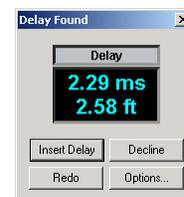


Figure 10: Impulse response of a small loudspeaker in a room.

Performing a Transfer Function Measurement of the Loudspeaker

Next, we will measure the frequency response function of the loudspeaker using the *Transfer Function* mode in SmaartLive. Click the  **Transfer** button to switch to the transfer function mode, and make sure the analyzer is running. We now need to set SmaartLive’s internal delay to compensate for the propagation time between the loudspeaker and microphone. Click the  **Auto Sm** button to do this; SmaartLive will run an impulse response measurement in the background and automatically measure the delay time. The “Delay Found” dialog box appears with the measured delay time; click “Insert Delay” to accept the shown delay time to compensate for the propagation delay during transfer function measurements. If the shown delay time seems impossibly long, you likely have the *reference* and *measurement* inputs swapped; simply swap the input cables and try again. This process may also be performed manually using the *Impulse* mode; simply click the  **Set Delay To Peak** button in the impulse response mode to set the transfer function delay to the peak arrival time.



You may use the *delay presets* feature to store several delay times for quick recall using the keyboard. This is useful if you are switching between microphones in different locations or between a measurement microphone and an equalizer, for example. To assign the current value to a preset, click the “Delay” label ( **Delay**) above the delay readout field and then click on one of the five buttons labeled **F6-F10** on the delay tab of the options dialog box. You may then recall this delay at any time by pressing the corresponding function key on the keyboard. For this example click on the button marked **F6** to store the current delay into the first preset. Press OK to exit the dialog box.

With the analyzer running in transfer function mode, you should now see the frequency response of the loudspeaker (energy vs. frequency) displayed in the window as shown in the example measurement in Figure 11. You may find that the display is somewhat erratic; the transfer function may be stabilized by increasing the number of averages using the *Avg* spinner. Set the number of averages 16 or higher to see improved behavior.

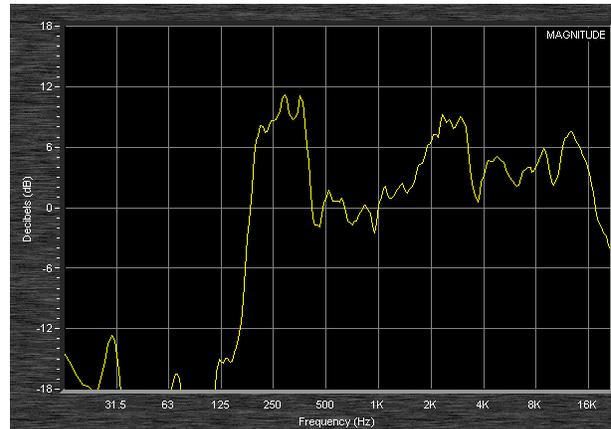


Figure 11: Sample transfer function measurement of a small loudspeaker.

Saving a Measurement Trace using the Reference Registers

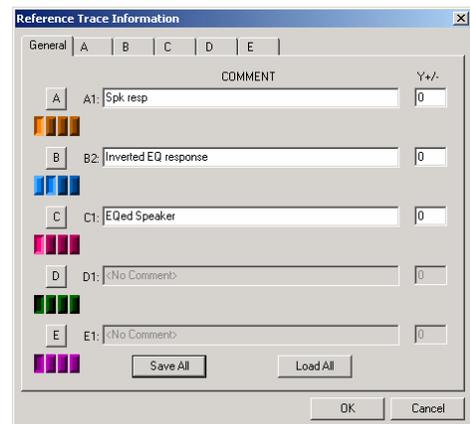
SmaartLive's *Reference Registers* are used to capture and store "snapshots" of the active live trace. The *Reference Registers* are represented by five groups of small solid-color buttons, labeled A, B, C, D and E, located below the plot area.



Click the button for register *A1* (the first register button in the *A* group). This "activates" the register even if the button was already depressed.

Click the **Capt** button below the plot area to sample and display the current trace as an overlay on the plot. Click the **A** button to remove the captured trace from the display. The sampled trace data, called a *Reference Trace* will remain stored in the register until you erase it or capture another trace to the same register.

To permanently save a *Reference Trace* to a file on disk, called a *Reference File*, click the **Info** button to the right of the capture button. This opens the *Reference Trace Information* dialog box. This dialog box has six tabbed "pages." Click on the tab labeled *A* in the upper portion of the dialog box to bring that page to the front.



Select the register containing the Reference Trace you just captured (by clicking the first of the four solid-color register buttons on the left) and click the *Save* button. This opens a Windows Save file dialog box prompting you to select a file name ending with the (**.ref*) extension. *Reference Files* may be recalled later and displayed as traces by selecting a register in this same dialog box and pressing the *Load* button. You can also save and reload the contents of all 40 reference registers as *Reference Group* (**.rgp*) files using the *Save All* and *Load All* buttons on the General tab of this dialog box. After saving the reference trace to a file, click the *OK* button to exit the *Reference Trace Information* dialog box.

Note that when you capture a reference trace, the stored trace is initially displayed “in front” of the live trace on the plot. The text color in the *dB +/-* spinner field to the right of the plot changes to match the reference trace color and when cursor tracking is enabled, the mouse tracking cursor follows the stored trace instead of the live trace. You can return the focus of the display to the live transfer function trace by clicking anywhere on either input level meter with your mouse.



Example Application 4 Equalizing a Loudspeaker

In this example, we will use the transfer function measurement from Example 3 to set an equalizer to optimize the performance of the loudspeaker. To perform this exercise, you will need the equalizer, loudspeaker, and power amplifier from the previous examples.

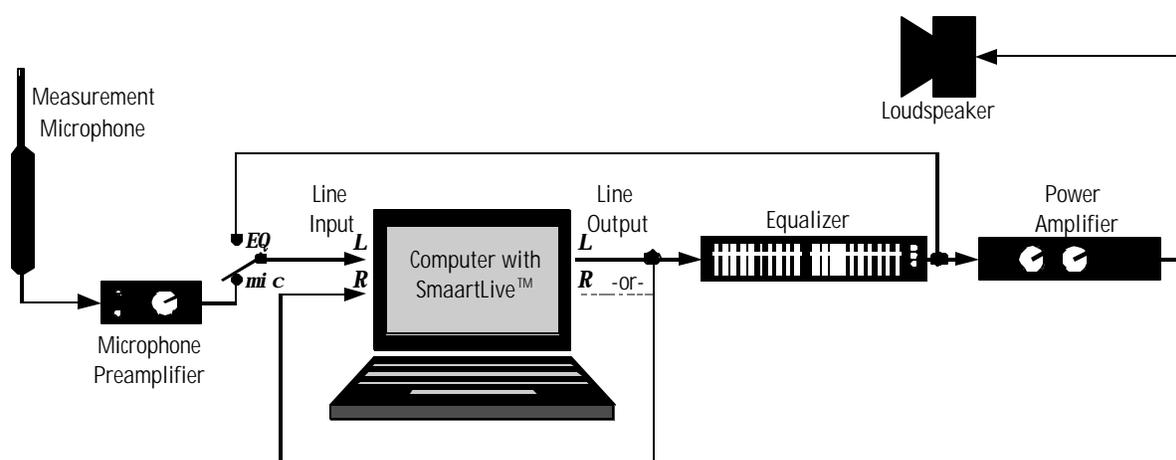


Figure 12: Hardware configuration for equalizing a loudspeaker.

Connecting the System

Connect the system as shown in Figure 12 above. When compared to the last example, this configuration simply inserts the equalizer into the signal chain between the sound card output and the power amplifier. A “return line” is also included, which simply receives the signal that is present after the equalizer, allowing you to adjust the filters in real-time exactly as in Example 2. The switch shown on the *left* sound card input is simply to show that you will wish to select between the measurement microphone signal and the signal coming directly from the output of the equalizer. This may be accomplished in many ways, including the use of a small mixer, a physical switch, or by simply re-patching the input cables.

Performing the Measurements and Setting the Equalizer

We will assume that you have performed the measurement of the loudspeaker as described in Example 3, and that the transfer function trace from the measurement is stored in reference register *A1* with the propagation delay time stored into the first delay preset (recallable by pressing **F6** on the keyboard). If this is not the case, perform the measurements described in Example 3 before proceeding.

Now, reconfigure the inputs such that the *left* or *measurement* input is taken from the output of the equalizer. Make certain that the input and output levels are appropriate and no clipping is present on any device. If you are using an analog equalizer, press **F5** on the keyboard to reset the internal delay to 0 ms.

If you are using a digital equalizer (and/or digital mixer), click the **Auto Sm** button to measure and insert the appropriate delay time into SmaartLive.

To equalize the system, we will calibrate the equalizer to have a frequency response that is approximately *inverse* to the response of the loudspeaker. In other words, the *peaks* in the loudspeaker response will correspond to equivalent *nulls* in the equalizer response, flattening the frequency response curve. We will use the stored reference trace containing the measured loudspeaker response as a template by which to adjust the equalization filters.

Make sure that the reference trace containing the loudspeaker response measurement is active on the display (in this case, *AI*) by using the reference trace control buttons. Use the *dB +/-* spinner to center the curve vertically near 0 dB. Now, click the **Swap** button to *invert* the measured frequency response of the equalizer. Any *cut* filters will now be displayed as *boosts*, and vice-versa, which allows us to visually match the inverted equalizer response to the loudspeaker transfer function to optimize the equalizer settings. Now you may adjust the equalizer to best fit the inverted equalizer response to the measured loudspeaker transfer function (Figure 13). After the equalizer settings have been optimized, you may perform another transfer function measurement of the entire system to see the final result (Figure 14). To perform this measurement, select the measurement microphone as the *measurement* input, and recall the appropriate delay preset by pressing **F6** on the keyboard.

As a practical note, boost filters are best used sparingly when optimizing the frequency response of a sound system. Excessive use of boost filters may destabilize a sound system by reducing gain-before-feedback and/or headroom, as the nulls that you observe in measurements may not be present in all spatial locations (due to comb filtering from reflections, room modes, loudspeaker interference, etc.). If you find that you require a large number of boost filters or that any required filter is very narrow in bandwidth, your problem may not be best resolved through equalization alone. Correction of acoustical conditions, crossover settings, or loudspeaker arrangements may be necessary. Typically, electro-acoustical phenomena that produce wide bandwidth (low-Q) peaks in frequency response are most effectively addressed through equalization. Additionally, we recommend the use of *parametric* equalizers for precision equalization, to enable the selection of proper *bandwidth* and *center frequency* for each filter.

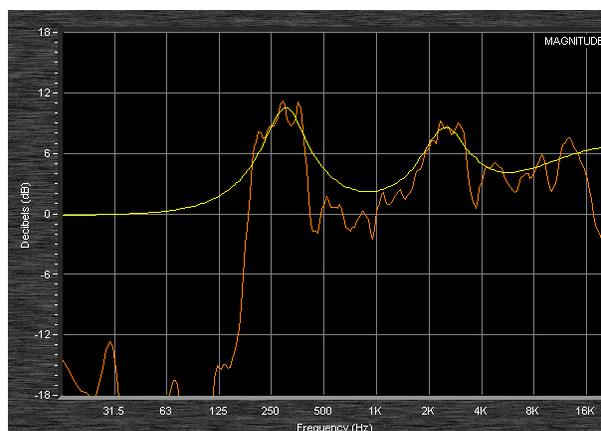


Figure 13: Real-time inverted equalization filters (yellow) matched to the measured loudspeaker response (orange).

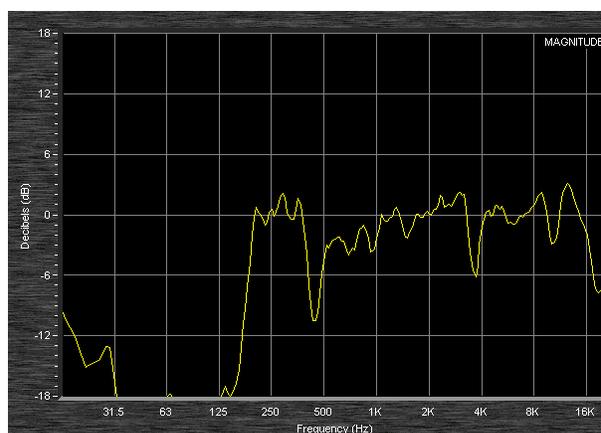


Figure 14: Resulting equalized loudspeaker response after applying the filters from Figure 12.