

The Fundamentals of FFT-Based Audio Measurements in SmaartLive®



Paul D. Henderson

This article serves as summary of the Fast-Fourier Transform (FFT) analysis techniques implemented in the SIA-SmaartLive® measurement platform. By reading through this document, you will receive an understanding of the fundamental concepts in FFT-based measurements used throughout the SmaartLive application, providing you with insights to better comprehend the measurement parameters, procedures, and resulting data. As a prerequisite to this text, you should be familiar with the basic concepts presented in the article “Getting Started with SmaartLive: Basic Measurement Setup and Procedures”.

Time Domain Sampling: Getting Signals into SmaartLive

Most of the acoustical and electrical signals that we may wish to measure are signals that are *continuous*, that is, they have a defined value for every possible instant in time. Sound pressure and analog voltage are two examples of continuous-time signals that we may wish to measure using our instrumentation. However, in order for these continuous signals to be analyzed using a computer-based measurement system such as SmaartLive, we must convert the signal into a stream of digital *samples*, with each sample representing a numeric value that is proportional to the measured signal at a specific instant in time. This process is called *sampling*: converting the continuous-time signal into a *discrete-time* signal (a process handled by an analog-to-digital converter in the computer sound card).

The sampling process employed for SmaartLive measurements (and for most other purposes in digital audio) creates digital signal data spaced on an even interval of time. The number of *samples per second* is the familiar *sampling rate* (or *sampling frequency*), referred to here as **SR** in units of *Hz*. The sampling rate directly affects the *highest frequency* that we may analyze in the computer, conventionally called the *Nyquist limit frequency* (f_{\max}), which is exactly equal to *one-half the sampling rate*. For measurements on electroacoustical signals and systems, we are most interested in signals that lie in the frequency band from approximately 20 Hz to 20 kHz (the range of human hearing). Therefore, for most measurements, you will wish to choose the highest sampling rate compatible between your sound card and SmaartLive, typically either 48 kHz or 44.1 kHz, which will provide a measurement bandwidth of at least 20kHz.

$$f_{\max} = \frac{\text{SR}}{2}$$

f_{\max} = Nyquist limit frequency (Hz)
SR = sampling rate (samples/sec)

A parameter inversely related to the sampling rate is the *sampling period* (**T**) which refers to the length of time (in seconds) between samples. Our measurements will have a *time resolution* equal to **T**, meaning that we cannot discern any details from the time signal that have a duration less than this value. For example, if we use SmaartLive to measure *signal delays* by finding peaks in the impulse response, we cannot measure a time delay difference of less than **T**.

$$\text{SR} = \frac{1}{\text{T}}$$

SR = sampling rate (samples/sec)
T = sampling period (sec/sample)

In addition to the choice of an optimal sampling rate, we must also consider the *word length* of the analog-to-digital converter when configuring the measurement system. For every sample, the analog-to-digital converter must assign a defined digital bit pattern to represent the

amplitude of the signal at that instant. In effect, the longer the word length, or *bits per sample*, the higher the *dynamic range* of the measurement. In addition (and perhaps more importantly), increasing the word length increases the *amplitude resolution* of the measurement, as the sampled amplitude steps are distributed on a smaller interval. Inexpensive sound cards may offer word lengths of 16 bits, with professional cards offering 24 bit input and output capability. As with the sampling rate, set the word length in SmartLive to the *maximum values compatible with your hardware*.

✓ Quick Reference

Sampling rate (SR)	<i>The number of samples per second (in Hz) used in the conversion process. Sets the maximum frequency that may be analyzed (Nyquist=SR/2). Set to 48kHz or 44.1kHz in SmartLive for most measurements.</i>
Sampling period (T)	<i>The time interval between samples, equal to 1/SR seconds. Time-domain details of duration less than T (sec) will be masked (delays, reflections, etc.).</i>
Word length	<i>Number of bits used by the analog-to-digital and digital-to-analog converters in the sound card. Always use the maximum values compatible with your hardware.</i>

FFT Analysis: Viewing Frequency Domain Information

While performing measurements in the time domain may be interesting and useful in many cases, we require more information for most audio measurements: *spectral* information, or knowledge about the frequency content and behavior of the audio signals and of complete systems. Fortunately, there exists a defined technique for converting, or *transforming* data from the time domain into the *frequency domain*, where information exists about the spectral content of signals. The *Fourier Transform* allows us to convert a time signal to the *complex* frequency domain, meaning the spectral data contains information about both the *amplitude* and *phase* of the sinusoidal components that make up the signal (see sidebar). In addition, the Fourier Transform provides us with an *inverse transform*, which allows us to convert the complex frequency-domain signal data back into the time-domain *without losing information*. Therefore, the both the time- and frequency-domain data are *equivalent*: the two domains simply provide a different view of the same signal (see Figure 1).

SmartLive uses a digital implementation of the Fourier Transform called the *Fast Fourier Transform*, or FFT, which is simply a computationally-efficient method for computing the Fourier Transform on digital signals. The FFT works on *finite-length blocks* of sampled data (called FFT *frames*). We will use the notation N_{FFT} for the length of the FFT data frame in *samples* (or, equivalently *points*). From this value, we can easily compute the length of the FFT frame in units of time

Optional Note: Fourier Theory

The 19th century French mathematician *Jean Baptiste Joseph Fourier* proposed a concept allowing us to express any time signal as a function of fundamental frequencies. Fourier theory states that any complex time signal, be it noise, speech, music, etc., is composed of a combination of *sinusoidal waves* of varying *frequency*, *amplitude*, and *phase*. We can use this basic concept as a *transform*, or a mathematical method for moving signals between the time and frequency domains.

To convert a continuous signal $x(t)$ to its frequency domain counterpart $X(j\omega)$, we may use the forward *Fourier Transform*:

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt$$

We may reverse this operation with no loss of information using the *Inverse Fourier Transform*:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t} d\omega$$

It should be noted that, in the most formal sense, the Fourier Transform requires the complete time history of a signal (for all time: an infinite-length view) and an infinite number of sinusoidal frequency components to fully describe a signal. This is, obviously, of no practical use for measurement, since we can only observe the signal for a finite amount of time. To implement the Fourier Transform computationally, we must utilize *time windowing* to limit our view of the signal to a finite frame of time. We may utilize the *Discrete Fourier Transform* (DFT), which operates on sampled-data signals, or the *Fast Fourier Transform* (FFT), which computationally accelerates the DFT.

using the sampling rate. We will call this value the *time constant TC*, representing the length of time that each FFT frame observes the continuous input signal. Higher the values of *TC* provide a longer time “window” in which we observe the signal, but also increase the time between subsequent updates of the FFT spectrum and the amount of numeric data that must be processed.

$$TC = N_{FFT} / SR$$

TC = FFT time constant (sec)
 N_{FFT} = FFT size (samples)
 SR = sampling rate (samples/sec)

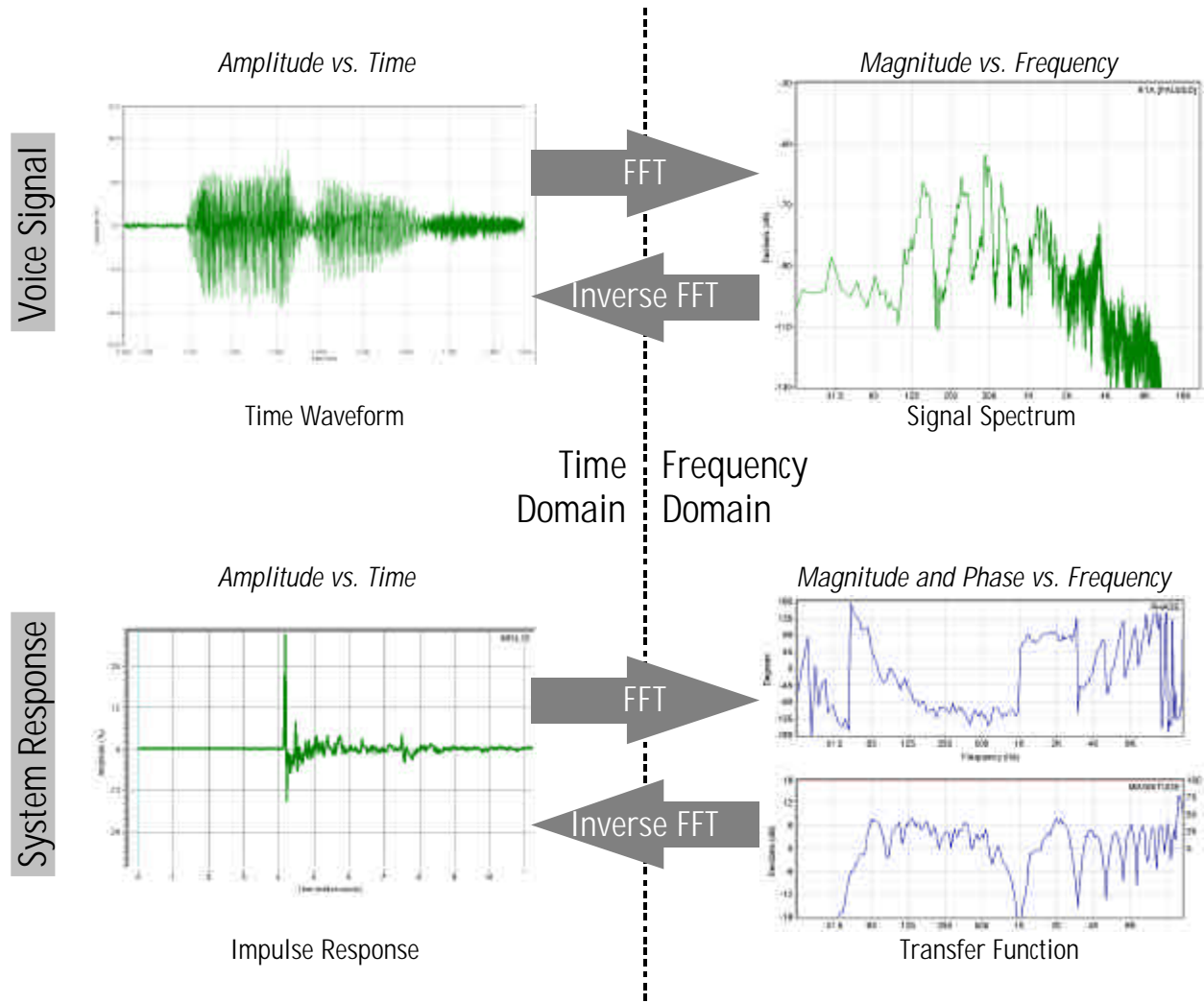


Figure 1: The Fourier Transform: moving signals between the time and frequency domains. Upper example shows the conversion between an impulse response and a transfer function; lower example is the conversion between a time signal (voice sample) and its spectrum.

The FFT generates frequency domain data that is *linearly-spaced* as a function of frequency, unlike our hearing, which primarily perceives frequency in a logarithmic manner. Conveniently, the time constant *TC* is inversely proportional to the *frequency resolution* of the spectral data, referred to here as *FR*. The FFT spectrum will include complex frequency data (magnitude and phase) spaced on even intervals of *FR* extending from 0 Hz (DC) to the Nyquist frequency. With this concept, it is easy to see how resolution is inversely related between the time and frequency domains. Using longer FFT sizes provides higher-

$$FR = 1 / TC$$

FR = Frequency resolution (Hz)
 TC = FFT time constant (sec)

resolution spectral data but more “sluggish” time response, while shorter FFT sizes provide lower spectral resolution but faster time response. Figure 2 graphically demonstrates the effect of changing the FFT parameters on resolution in the frequency domain. When distributing the linearly-spaced FFT data on a logarithmic axis, it can be easily seen that short FFT’s may provide inadequate low-frequency resolution, while long FFT’s may provide excessive high-frequency resolution.

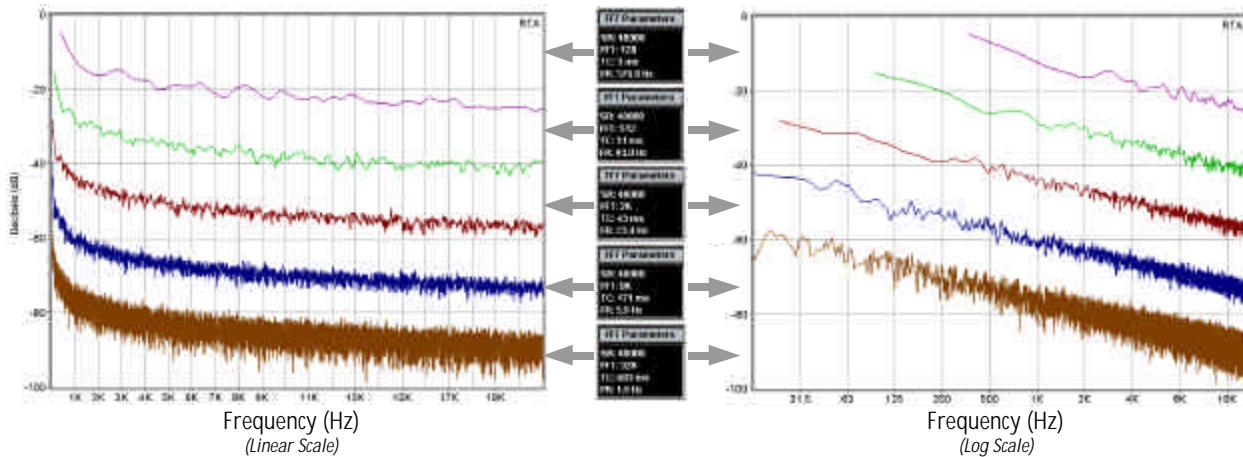


Figure 2: The effect of FFT parameters on frequency resolution. Note that the FFT spectrum data is equally-spaced on a linear frequency scale but exponentially-distributed on a logarithmic frequency scale. This may yield inadequate low-frequency resolution for short FFT’s and excessively detailed high-frequency resolution for long FFT’s.

Fortunately, we are not required to manually calculate the parameters that have been discussed here. The *FFT Parameters* function in SmartLive allows for the independent selection of sample rate, FFT size, time constant, and frequency resolution. A change in any one parameter immediately updates the other dependent parameters, allowing the user to concentrate on the meaning of the values, not on their calculation. An indicator in the main SmartLive window shows the active settings for the live input signals.

FFT Parameters

SR: 48000

FFT: 8K

TC: 171 ms

FR: 5.9 Hz

✓ Quick Reference

FFT	<i>The Fast Fourier Transform, a method for moving digital signals between the time and frequency domains.</i>
FFT length (N_{FFT})	<i>The length of the FFT input data frame in samples.</i>
Time constant (TC)	<i>The length of the FFT input data frame in seconds, equal to N_{FFT}/SR. Indicates the length of time that the FFT observes the signal in each data frame.</i>
Frequency resolution (FR)	<i>The frequency resolution of the FFT spectral data, in Hz, equal to $1/TC$. FFT data is linearly spaced from 0Hz to the Nyquist limit on even intervals of FR.</i>

Improving the Spectral View

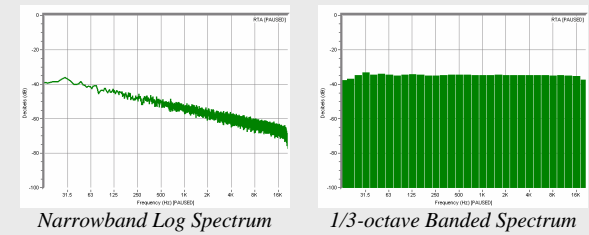
We have shown that the frequency domain spectral data from an FFT is distributed on a constant, *linearly-spaced* interval in frequency. If we evenly distribute the FFT data onto a graph, we will need to employ a frequency axis that is equally spaced in frequency, that is, there is an equal *interval per Hz*. In contrast, our hearing perceives musical pitch on (approximately) equal *frequency ratios*, yielding a *logarithmic frequency axis* for equal-interval analysis (an equal interval *per octave*). Obviously, if we

distribute the linearly-spaced FFT data on a logarithmic frequency axis, the apparent resolution along the frequency axis is not constant, possibly with insufficiently low resolution in the lower octaves, and inappropriately high resolution in the upper octaves (as seen in Figure 2). We can maximize the low-frequency resolution by increasing the FFT size, however, this leads to a longer time constant and, consequentially, slower time response. In addition, for transfer function measurements, the increased time window will also include additional room reflections in the measurement, potentially obscuring details in the response curve. Obviously, we require additional methods for viewing the FFT data in a perceptually-significant manner to better correlate the graphical data with human hearing.

SmaartLive contains several methods for improving the logarithmic-frequency view of FFT data, which vary according to the analysis task at hand. For spectrum-mode data (where we are simply interested in the spectrum of the input signal), we may employ frequency-domain *banding*, which distributes the data into equally-spaced *fractional octave bands*. The spacing and, consequentially, the width of each band is selectable as either 1-, 1/3-, 1/6-, 1/12-, or 1/24-octave. Figure 3 compares the log-distributed FFT spectrum with the banded spectrum for an input signal.

Fractional-Octave Banding

The Spectrum mode banding technique provides an FFT-based approximation of analog Real-Time Analyzer (RTA) filters, which utilize electronics to separate the incoming spectrum into bands and then derive the RMS level of each band over time. The banded views in SmaartLive provide an accumulation of the total energy in each logarithmically-proportioned band (just like a hardware RTA), so signals with constant power per octave (like pink noise) will result in a *flat* banded spectrum (see figure below). Signals with constant power per Hz (like white noise) will indicate a *rising* (+3 dB/octave) banded spectrum. The banded spectrum view is very useful for identifying the generic frequency content of input signals, identifying feedback frequencies, and investigating room noise levels. However, system equalization and tuning procedures are best performed using the *transfer function* measurement mode.



Pink Noise Spectrum: Note the apparent downward slope seen when viewed as a narrowband spectrum becomes a flat curve in the log-proportioned banded view. In this case, the banded view more accurately reflects the perceived timbre of the signal.

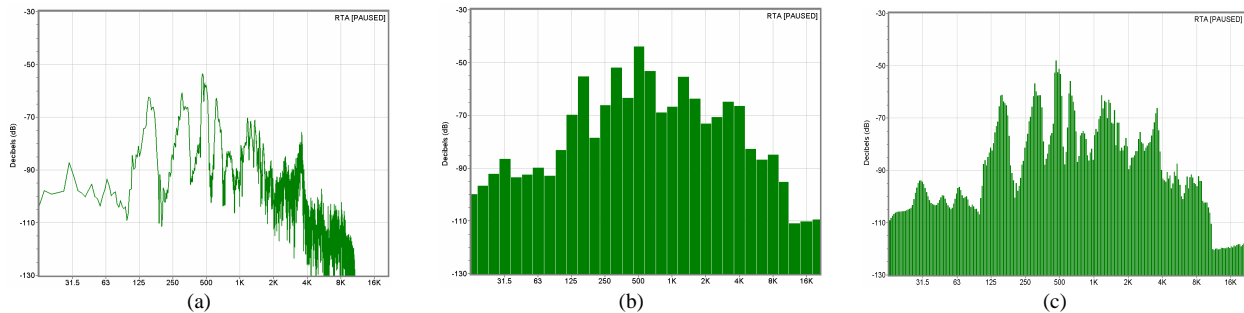


Figure 3: Spectrum-mode frequency banding; (a) original FFT spectrum data distributed onto a log-frequency axis, (b) spectrum data displayed in 1-octave bands, (c) spectrum data displayed in 1/24-octave bands.

In addition to the banding method for handling spectrum-mode signals, the *Fixed-Point per Octave* (FPPO) technique in SmaartLive is used for computing logarithmically-spaced *Transfer Function* (frequency response) data. In effect, the technique utilizes a measurement *time window* that varies as a function of frequency, utilizing a long time window at low frequencies (for narrow frequency resolution) and a successively shorter time window at high frequencies. This method has two main effects: the variable time window is well-correlated with the hearing perception mechanism defining the *perceived*

spectral quality of a loudspeaker operated within a room, and the variable analysis frame lengths provide frequency-domain data that is of *equal density per octave*, in this case, *24 points per octave*.

For most transfer function measurements in SmartLive, especially those involving some acoustical path (a loudspeaker or sound system measurement), the FPPO view provides the best representation of the system response function. Beyond the inherent low-frequency advantages, the FPPO technique typically provides a more easily readable trace at high frequencies, in contrast to the “fuzzy” character of standard FFT data caused by excessive high frequency resolution (see Figure 4).

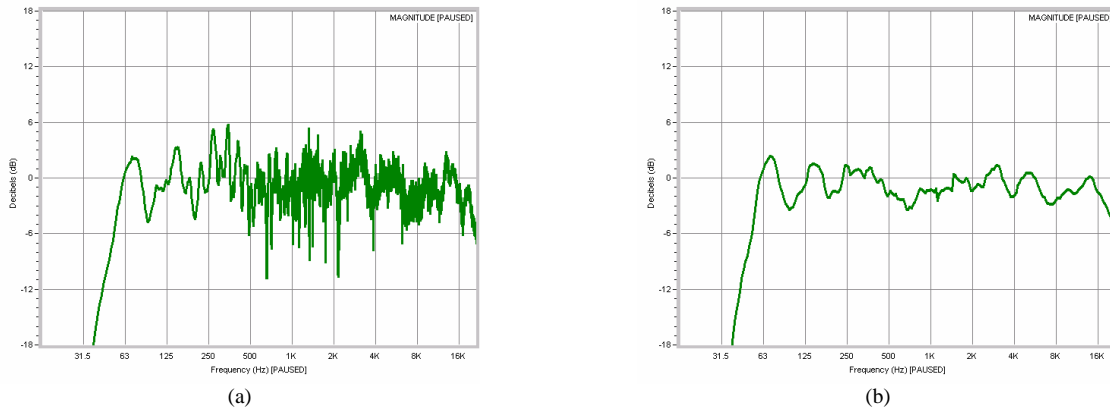


Figure 4: Log-frequency transfer function analysis; (a) transfer function of a small loudspeaker using a 32k-point standard FFT, (b) measurement of the same loudspeaker using the FPPO technique.

In SmartLive’s transfer function mode, an additional option exists for *smoothing* of the transfer function trace over a definable number of points. The smoothing function is, effectively, a moving average filter that is applied to the transfer function data before it is displayed in order to minimize the presence of jagged edges and discontinuities in the displayed data. You may select either 3-, 5-, 7-, or 9-point smoothing depths, which define the number of FFT data points surrounding an individual value that are averaged to derive the displayed value; higher numbers yield a more continuous visual curve. Figure 5 shows an example of smoothing applied to a transfer function measurement of a small loudspeaker. Trace smoothing is available for both standard FFT sizes and FPPO curves.

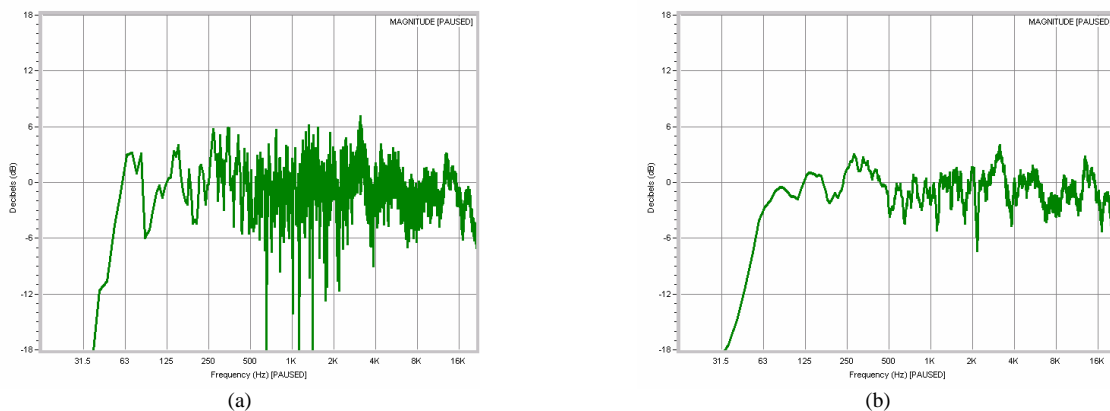


Figure 5: The effect of curve smoothing on transfer function measurement display, 8k-point FFT; (a) measurement of a small loudspeaker with no smoothing, (b) the same data displayed with 9-point smoothing.

✓ Quick Reference

Linear frequency scale	FFT data is distributed on a frequency axis scaled on equal intervals <i>per Hz</i> .
Logarithmic frequency scale	FFT data is distributed on a frequency axis scaled on equal intervals <i>per octave</i> , which corresponds to human pitch perception.
Fractional-octave banding	Used to represent Spectrum mode signals in logarithmically-distributed bands, much like a hardware RTA. Bands may be 1-, 1/3-, 1/6-, 1/12-, or 1/24-octave wide.
Fixed-Point Per Octave (FPPO)	Performs Transfer Function mode analysis on logarithmically-spaced (1/24-octave) frequency bins. Enhances trace readability and correlation to hearing perception when equalizing systems.
Trace smoothing	Further smooths Transfer Function mode signal traces for improved curve readability.

FFT Frame Averaging to Improve Trace Validity

While the signal for each FFT frame is viewed only for a finite amount of time (the length of the *time constant, TC*), we can modify the time response of the measurements by using *frame averaging* techniques. Effectively, averaging causes the current displayed FFT data to reflect some cumulative average between the current data and past FFT frames. By varying the depth and function of the averaging, we can achieve changes in the apparent time response for measurements.

In SmartLive's *Spectrum* mode, several options are available for configuring the averaging technique used for measurement. The spectrum mode contains options for FIFO (First-In/First-Out) averaging, where the last 2, 4, 8, 16, 32, 64, or 128 FFT frames are averaged with equal weighting, and the result is displayed (see Figure 6). These modes allow the user to fine-tune the averaging *depth* depending on the measurement task and input signals: higher averaging frame counts provide a slower time response, with lower numbers better approximating the instantaneous behavior of the signal. The spectrum mode also contains *exponential* averaging techniques, which are marked *Fast*, *Slow*, and *Exp*. The *Fast* and *Slow* settings reflect the standard damping of a sound level meter in the associated integration mode, which are most useful for performing repeatable, standardized measurements. The *Exp* mode allows for the exponential averaging *half-life* to be customized, allowing the exponential averaging modes to be user-optimized based on the measurement task. Finally, the *Inf* setting allows for an (effectively) infinite-length average of the input data, which provides a running, equal-weight average of all FFT frames since the last buffer *reseed*. This is useful for general noise-level analysis tasks and specialized uses in cinema optimization, etc. It should be noted that the spectrum-mode averaging is performed on the *power spectrum* of the input signal.

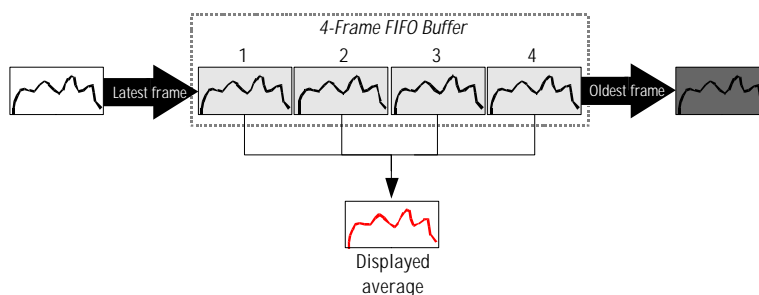


Figure 6: 4-frame FIFO-based averaging example; the displayed curve is a function of the mean curve from the last N FFT frames.

Similarly, SmaartLive's *Transfer Function* mode contains averaging capability, which provides two primary functions: (1) it provides a variable time response (identical to the behavior seen in spectrum mode), and (2) it provides a mechanism for increasing the validity and signal-to-noise ratio of the transfer function measurement. By averaging transfer function measurement frames over time, the estimate of the system transfer function increases in precision, with any random noise effects being reduced. When using the FIFO-based averaging scheme, a doubling of the number of averages corresponds to a 3 dB increase in the signal-to-noise ratio of the measurement. In addition to basic averaging depth options, the transfer function mode allows the user to select either *RMS* or *Vector* averaging types. The difference between these two modes is the transfer function measure that is being averaged; in RMS mode, the geometric mean of the transfer function magnitude is computed, whereas in Vector mode, the complex transfer function (including phase) is used for the averaging operation. Vector mode provides the most robust averaging technique for use in *phase/time-invariant* situations, where no wind gradients or rapid variations in the measured field are present. For most common measurements, however, RMS averaging provides a reasonably effective method of increasing the validity of the transfer function measurement with less sensitivity to these issues. It should be noted that, in RMS mode, the averaged phase trace is derived from a vector average; only the magnitude curve is obtained through RMS averaging.

SmaartLive's *Impulse* mode utilizes an RMS-based averaging technique, which operates solely on a FIFO-style FFT frame buffer. As with the transfer function mode, increasing the averages by a factor of 2 corresponds to a 3 dB increase in signal-to-noise ratio. The sole purpose of averaging in the impulse response mode is to improve data validity and reduce the impact of measurement noise, although higher averaging depths will directly correspond to longer acquisition times in this mode.

✓ **Quick Reference**

Frame averaging	<i>Improves data validity and trace stability by deriving the displayed data from a running average of the current data and past FFT frames.</i>
FIFO averaging	<i>The displayed data is an equal-weight average of the current input data with a finite number of past frames. Available FIFO lengths are 2, 4, 8, 16, 32, 64, or 128 frames.</i>
Exponential averaging	<i>Provides an integrated averaging response similar to that of a sound level meter. Provided are Fast, Slow, and Exp (custom) response characteristics.</i>
RMS averaging	<i>Averaging behavior in this mode utilizes the geometric mean of the response data. Used in the Spectrum and Impulse modes and available as a option in the Transfer Function mode. Useful in Transfer Function mode when the system is varying in time (wind gradients, etc.).</i>
Vector averaging	<i>Averages the complex value of the transfer function data, providing maximum precision when the system is time-invariant.</i>

Measurement Time Windows

We have noted previously that, for real-world measurements, the FFT is only able to observe the signal for a finite amount of time, which we may call the measurement *time window*. Beyond the basic observation that we may only draw conclusions about the signal from this narrow slice of time, this operation also creates anomalies in the frequency domain, which may lead to errors in the observed frequency spectrum.

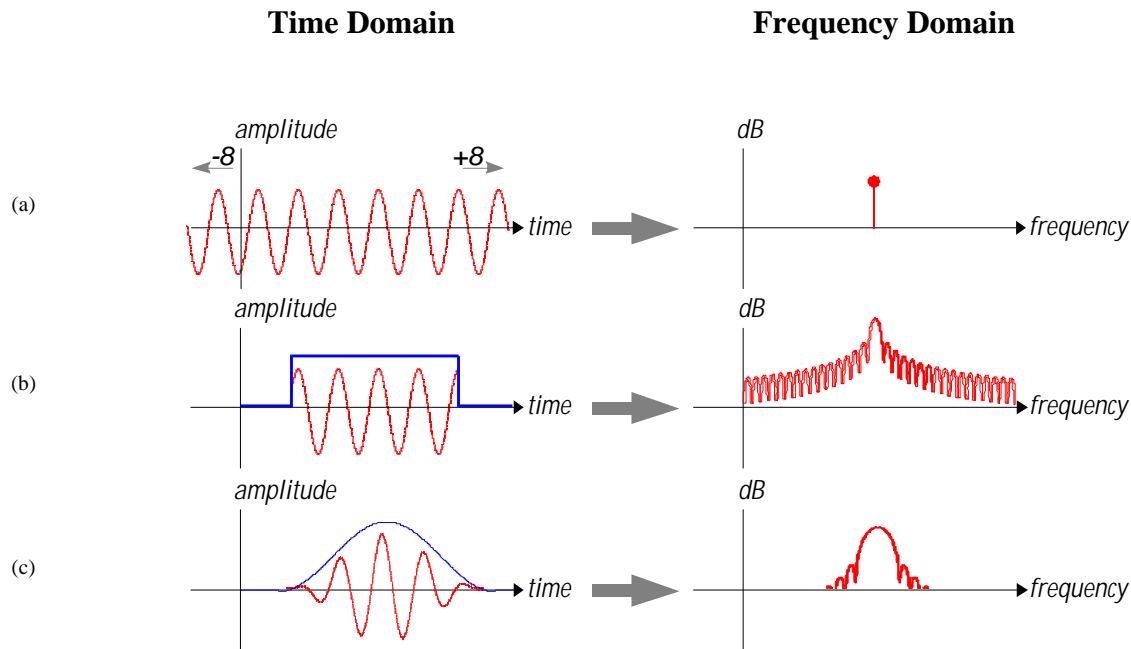


Figure 7: Time windowing effects in the Fourier transform with a sinusoidal signal; (a) an infinite time window, (b) a finite Rectangular time window, (c) a finite Hanning time window.

Figure 7 demonstrates the effect of a finite time window on the FFT spectrum of a sinusoidal (single-frequency) input signal. A steady-state sinusoidal signal has energy at only one unique frequency, so the ideal spectrum produced by the Fourier transform should indicate this by an infinitely narrow “line” at that frequency. As shown in Figure 7a, the Fourier transform is able to accurately determine the spectrum, *given an infinite time record* of the signal. However, if we reduce our view of the signal to a finite-length FFT frame, we, in effect, multiply the time signal by a *rectangular window*, as shown in Figure 7b. The effect of this windowing operation on the frequency domain is obvious, as the energy is dispersed into a single *main lobe* plus a number of *side lobes* (called *spectral leakage*). The pattern of the side lobes is a function of the rectangular window, and is primarily created by the abrupt *discontinuity* at the edges of the window. The high relative level of these side lobes could cause other important spectral information to be hidden, or *masked*. However, if we apply a finite but gentle time windowing function, as seen in Figure 7c, we can significantly reduce the level of the side lobes by gradually tapering the waveform to zero at the ends. A side effect of this operation causes the main lobe to increase in width, which may also mask spectral details from nearby frequency components.

We can balance the side lobe magnitude and pattern against the main lobe width by careful selection of the windowing function, which can be customized depending on the analysis task. SmaartLive contains a number of time windowing functions which provide varying characteristics in time and frequency. For more information, see the SIA technical note “The ‘Mystery’ of Data Windows”, which covers this topic in further detail.

Optimal FFT Parameter Settings for SmaartLive

This section will provide suggestions for reaching the optimal FFT parameter settings depending on the measurement task at hand. As each parameter is highly configurable, a general set of criteria is presented here that allow you to optimize the measurement system depending on the desired temporal and spectral

resolution. For most uses, once an optimal set of parameters is reached for a specific measurement task, the FFT parameters may be left unmodified for subsequent measurements.

A parameter that will not be specifically addressed as a variable is the selection of an optimal *sampling rate*, which defines the highest measurable frequency (Nyquist) and the time resolution of the measurement. As noted previously, you should typically select the highest compatible sampling rate between SmaartLive and your hardware interface, usually either 48 kHz or 44.1 kHz. On rare occasion this guideline may be modified, such as those situations requiring a lower-bandwidth measurement where computational power may be conserved by choosing a lower sampling rate.

Spectrum Mode

For most real-time analysis measurements in *Spectrum* mode, we are primarily concerned with three major criteria: appropriate spectral resolution for the target measurement, an appropriate response characteristic in time, and a reasonable correlation to human hearing perception. We may use these basic criteria to develop guidelines for choosing FFT parameters in this mode.

In spectrum mode, the FFT size should be configured for an optimal trade-off between low-frequency resolution and time response. A good starting point may be to choose a 16k-point FFT frame, which tends to provide acceptable frequency resolution with an ~350 ms time constant (at **SR** = 48 kHz) for a reasonable time response characteristic. For detecting dynamic changes in signals, a *Fast* averaging characteristic is useful, with the *Slow* setting more useful for analytical measurements (noise, etc.). Of course, you may experiment with different FFT parameter settings to optimize performance.


For most measurements in this mode, you will wish to choose a banded display; a 1/3-octave display is a good starting point for many measurements, as it corresponds to the critical bandwidth of hearing perception for most complex signals. Higher resolution displays may be used to more accurately detect spectrum details, such as feedback center frequencies, etc.

Transfer Function Mode

Most transfer function measurements fall into one of two categories: measurement of the *electrical* transfer function of a device (an equalizer, etc.), or measurement of the *electroacoustical* transfer function of a complete system (a loudspeaker, sound system, etc.). For most electrical measurements, choosing a standard linearly-spaced FFT is typically a good starting point. Note that, for proper measurement of low-frequency filters, the FFT parameters chosen should have sufficiently narrow frequency resolution (start with 16k- or 32k-point FFTs). Electrical measurements typically provide an inherently high signal-to-noise ratio, so a short averaging time (8-16 averages or the *fast* response characteristic) may be used for a more dynamic time response.

✓ **Spectrum Mode**

You may wish to begin with these parameter settings:



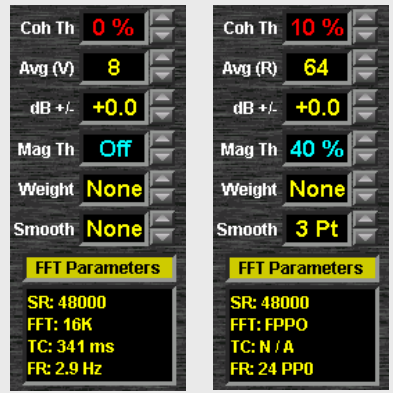
Scale 1/3
Avg Fast
dB +/-
Mag Th Off
Weight None

FFT Parameters

SR: 48000
FFT: 16K
TC: 341 ms
FR: 2.9 Hz

✓ **Transfer Function Mode**

You may wish to begin with these parameter settings:



Coh Th 0 %
Avg (V) 8
dB +/- +0.0
Mag Th Off
Weight None
Smooth None

FFT Parameters

SR: 48000
FFT: 16K
TC: 341 ms
FR: 2.9 Hz

Electrical measurements

Coh Th 10 %
Avg (R) 64
dB +/- +0.0
Mag Th 40 %
Weight None
Smooth 3 Pt

FFT Parameters

SR: 48000
FFT: FPPO
TC: N / A
FR: 24 PP0

Acoustical measurements

When performing electroacoustical measurements, the FPPO measurement mode should typically be used for best results. FPPO will best approximate the hearing perception of spectral effects in most electroacoustical systems and is typically far easier to interpret than standard FFTs due to the natural fractional-octave distribution of data. Electroacoustical measurements are far more susceptible to noise and time variance effects, so the use of a long averaging time is recommended (64+ averages or the *slow* response characteristic). In addition to these parameters, the transfer function mode contains *coherence* and *magnitude* threshold settings, which may be used to ensure valid data. Typical settings for these parameters when performing electroacoustical measurements are: coherence threshold = 10-15%, and magnitude threshold = 35-55%. The details of these parameters are outlined in following articles, and in the *SmaartLive User Guide*.

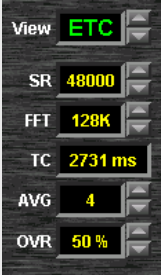
Impulse Response Mode

Selecting appropriate FFT parameters for the *Impulse* mode is somewhat more straightforward than for other modes in SmaartLive. Two primary criteria are at work for impulse response measurements: the decay time of the measured impulse response and the effective signal-to-noise ratio of the measurement. For effective impulse response measurements, the time constant **TC** *must* be greater than the total *decay time* of the system under test in order to assure valid data.

In performing impulse response measurements, it is often difficult to obtain an acceptable signal-to-noise ratio in order to fully view the entire decay of the system. As with transfer function measurements, *increasing the number of averages* will increase the effective signal-to-noise ratio of the measurement. An identical effect is seen by simply *increasing the time constant*; this provides an equivalent increase in the time the system is observed in order to increase the signal-to-noise ratio. Other methods exist for lowering the noise floor, such as *increasing the acoustic level of the stimulus* or by carefully selecting the stimulus signal type (such as using synchronized sinusoidal sweep signals). These advanced concepts are discussed in subsequent articles, as well as in the *SmaartLive User Guide*.

✓ **Impulse Mode**

You may wish to begin with these parameter settings:



Note: Increase FFT size such that TC exceeds the system decay time. You may increase the FFT size, averaging depth, and the stimulus level to maximize the signal-to-noise ratio of the measurement.

Suggestions for Further Reading

If you have an interest in furthering your knowledge of the engineering concepts behind FFT-based measurement, you may find the following texts useful:

- A. Oppenheim, A. Willsky, S. Nawab: *Signals and Systems*, 2nd edition. Upper Saddle River, NJ: Prentice Hall Inc. 1997.
- A. Oppenheim, R. Schaffer, J. Buck: *Discrete-Time Signal Processing*, 2nd edition. Upper Saddle River, NJ: Prentice Hall Inc. 1999.