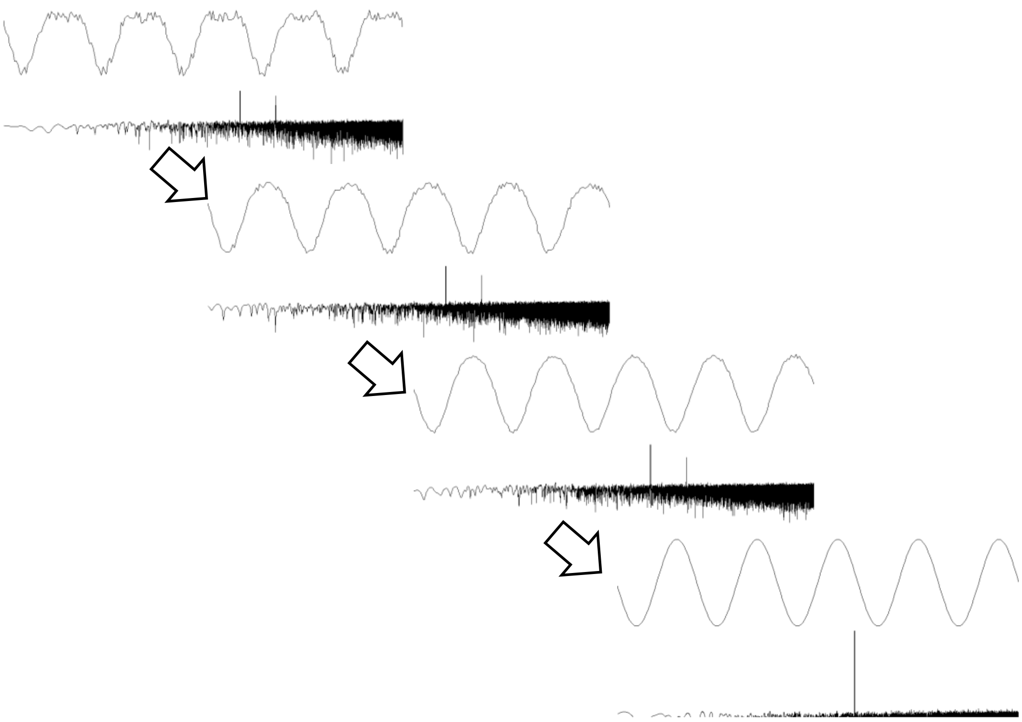
Software Methods for Non-Linear Distortion Compensation and Noise Reduction in Measurements



Wang Hongwei
(Ph.D)

Rev: 01
Sept. 2, 2024

Note: VIRTINS TECHNOLOGY reserves the right to make modifications to this document at any time without notice. This document may contain typographical errors.

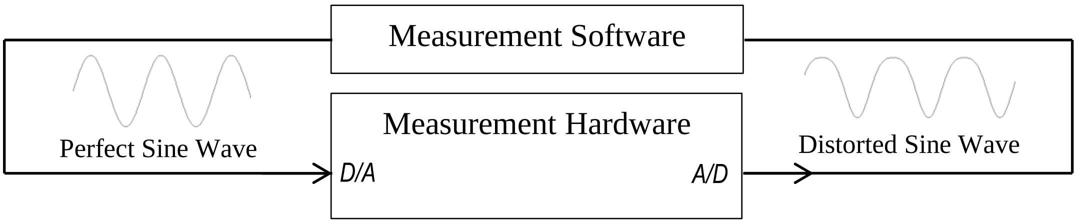
TABLE OF CONTENTS

1. Introduction

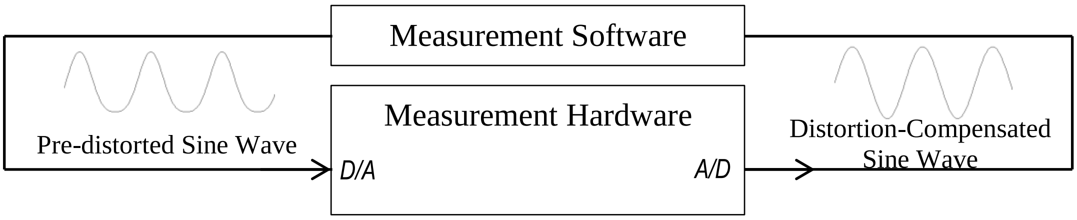
The lowest measurable levels of non-linear distortion and noise by a measurement system are always constrained by the system's own residual non-linear distortion and noise floor. While improving the performance of the system hardware might seem like the only option, there are software methods available to compensate for non-linear distortion and reduce noise in the measurement system. After applying these techniques, the system can measure lower levels of non-linear distortion and noise, potentially even significantly below its original residual levels.

In other cases, all the noise in the measured signal, whether embedded in the original signal or introduced by the measurement system, needs to be reduced to achieve a cleaner measurement. Several software approaches exist to accomplish this goal.

2. Non-Linear Distortion Compensation for a Measurement System



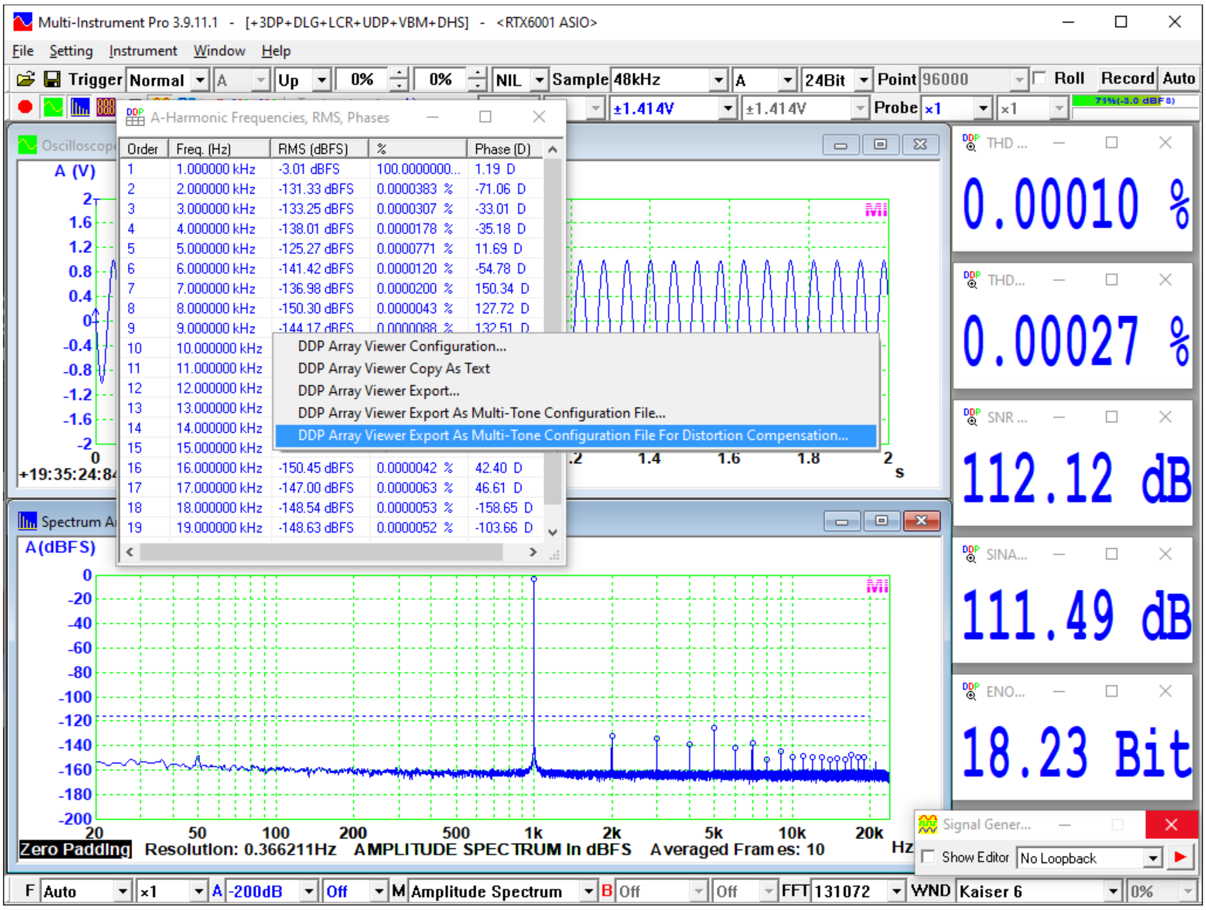
Before Distortion Compensation



After Distortion Compensation

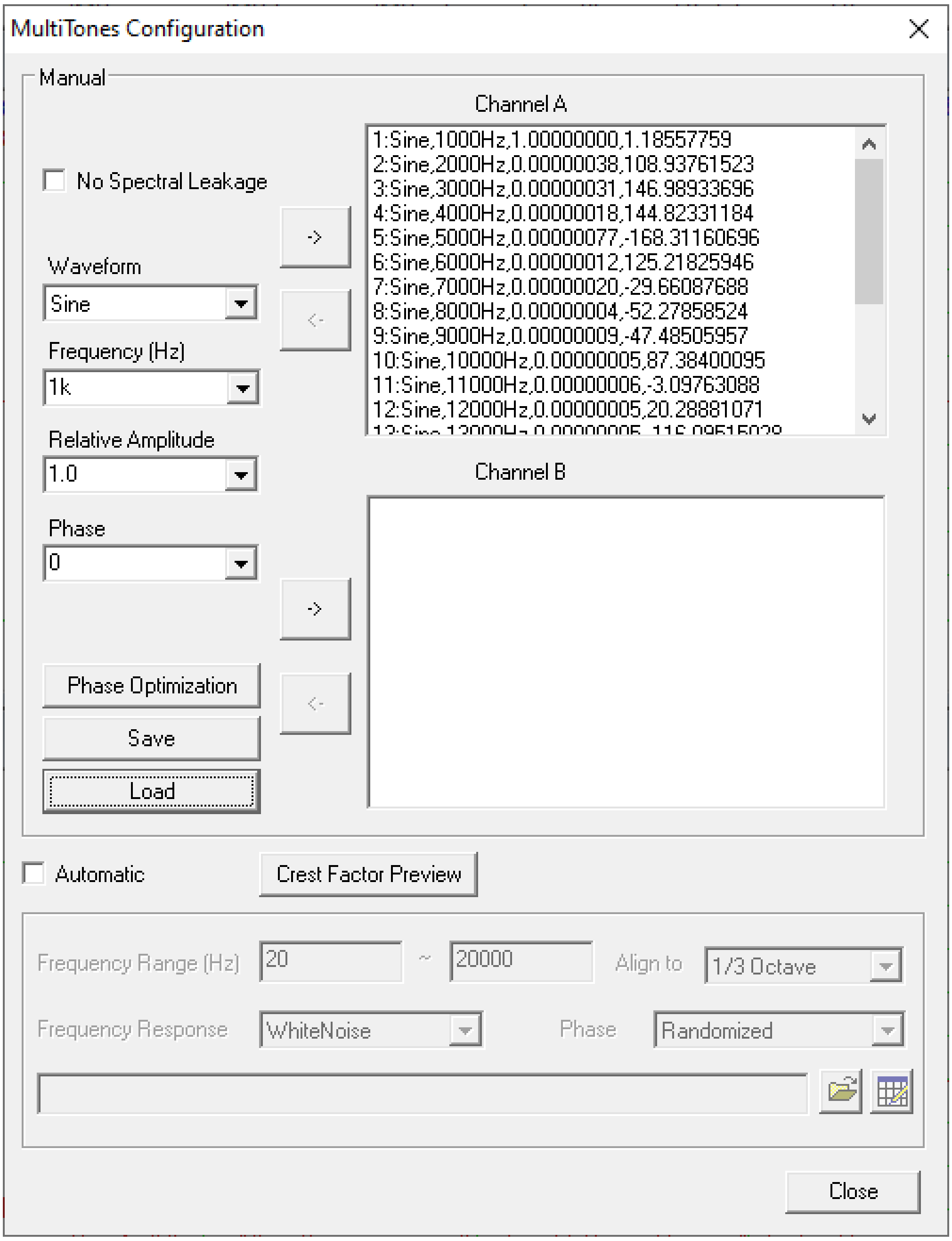
The techniques used to compensate for a system's frequency response (both magnitude and phase frequency responses), also known as linear distortion, are well-established. In contrast, techniques for compensating for non-linear distortion are less frequently addressed. The challenge arises because non-linear distortion varies with frequency content and magnitude. In many tests, such as THD (Total Harmonic Distortion) and IMD (Intermodulation Distortion) tests, the test signal typically contains only one or two single-frequency tones, resulting in a limited number of harmonic and intermodulation distortion components. In such cases, it is possible to counteract the distortion components by pre-distorting the test signal with an equal amount of distortion, but with inverted phases. This method will work if the distortion is static at the specified magnitudes and frequencies.

The example below demonstrates how to compensate for the residual harmonic distortion of a RTX6001 audio analyzer at 1kHz 1Vp, including both its ADC and DAC portions. First, its residual harmonic distortion at 1kHz 1Vp was measured through a loopback test. The results are shown in the figure below. The software used is Multi-Instrument, which can be downloaded from www.virtins.com. The measured THD is 0.00010% (-120dB). The DDP Array Viewer of the software lists the main composition of the measured signal, including the amplitudes and phases of its fundamental and harmonics. These data can be exported as a multi-tone configuration file for signal reconstruction with the measured noise removed. This file is a plain TXT file. The phase of each harmonic in the file can be manually inverted (i.e. +/- 180°) using a plain text editor such as Windows Notepad. The software also provides a convenient option to export the multi-tone configuration file with all harmonic phases inverted automatically, leaving only the phase of the fundamental untouched.



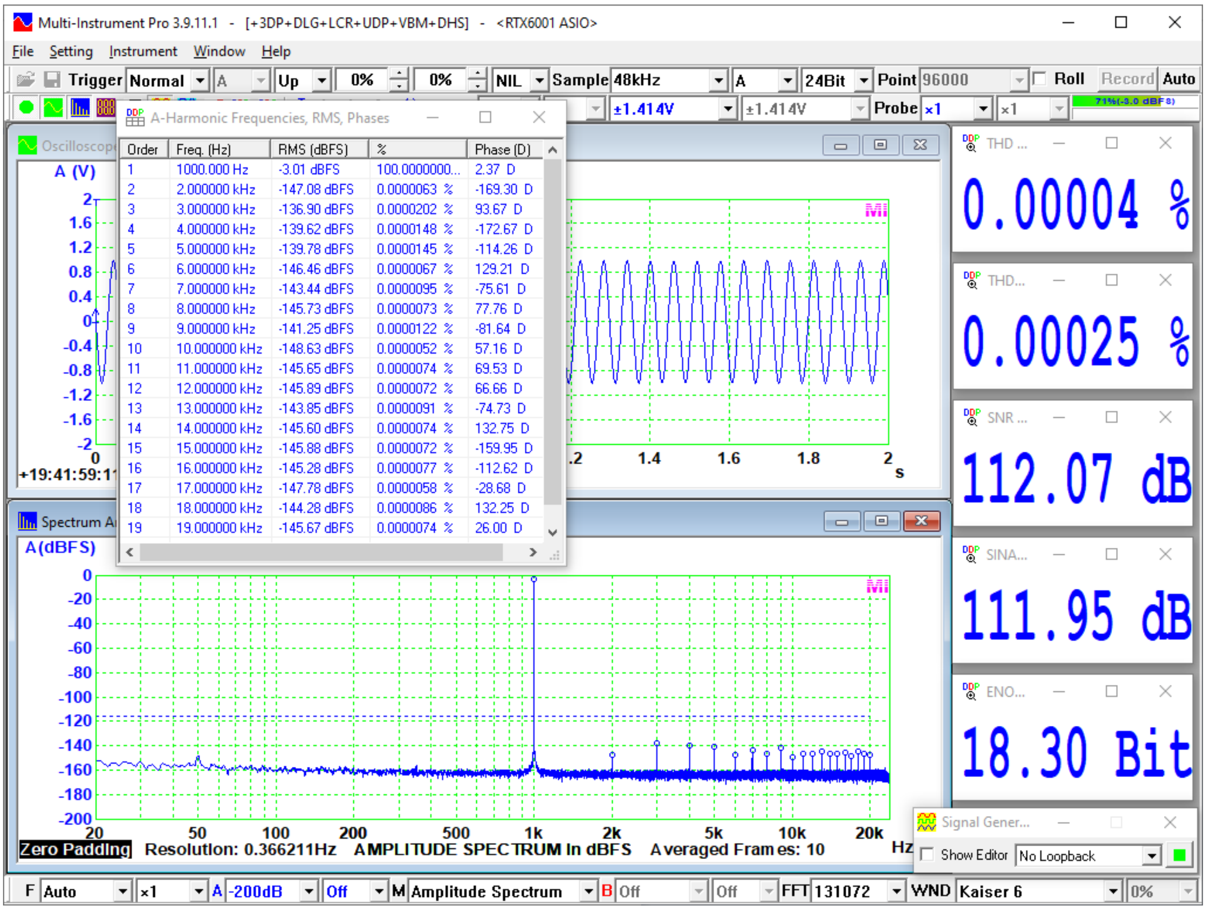
Before Distortion Compensation

The exported multi-tone configuration file was then loaded into the Signal Generator of Multi-Instrument to generate a pre-distorted 1kHz test signal (see its configuration below).



<u>Pre-distorted 1kHz Test Signal Configuration in Multi-Instrument</u>

The corresponding THD loopback test results are shown as follows. It can be observed that the residual THD dropped from 0.00010% (-120dB) to 0.00004% (-128dB), an improvement of 8dB.



<u>After Distortion Compensation</u>

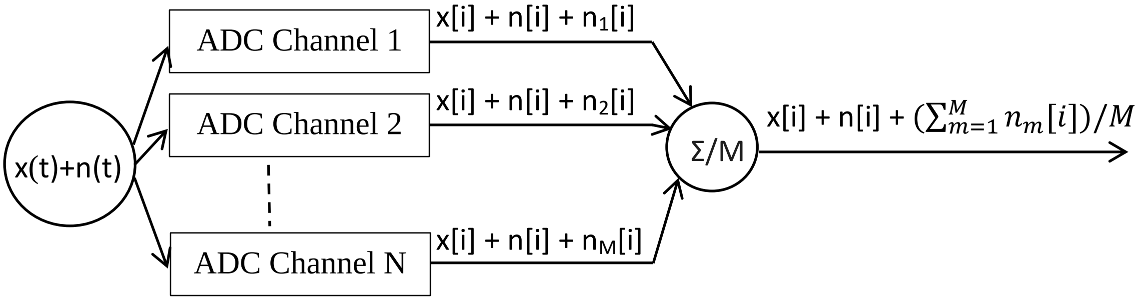
This method will not completely eliminate non-linear distortion due to factors such as amplitude and phase measurement errors, non-static distortion, and distortion of pre-distortion.

It can also be used to compensate for the ADC and DAC portions separately, as long as their respective residual distortions are measured accurately first.

3. Noise Reduction Methods for Various Purposes

Several noise reduction software methods are available, tailored to specific needs: either targeting noise solely from the measurement system or addressing all noise within the measured signal, regardless of its source. Time-domain methods are beneficial for analysis in both the time and frequency domains, whereas frequency-domain methods may or may not be effective in the time domain.

3.1 Multi-Channel In-Phase Averaging in Time Domain

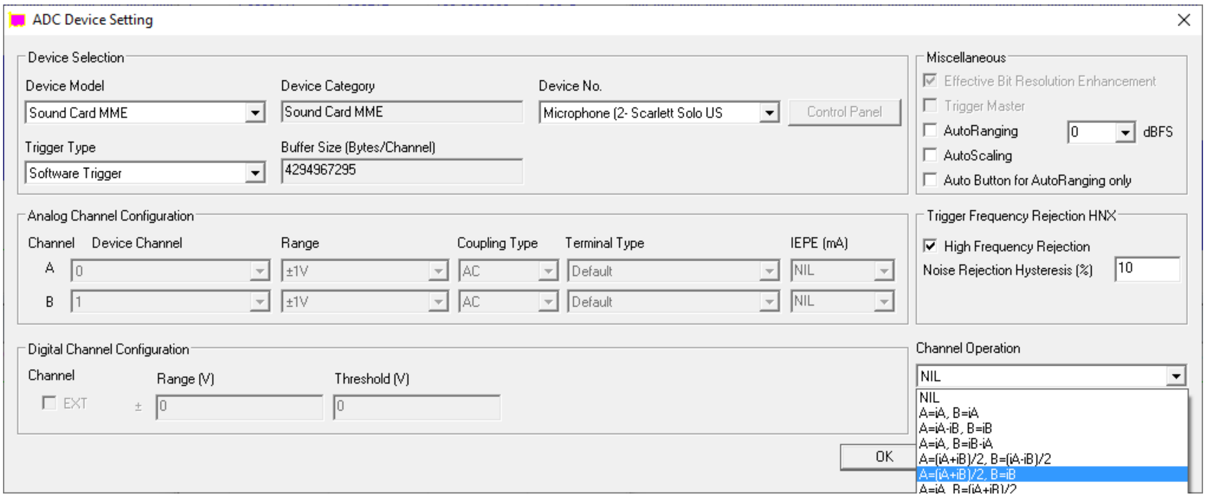


- x(t) is the analog signal to be measured, and x[i] is its sampled version.
- n(t) is the analog noise that comes with x(t), and n[i] is its sampled version.
- $n_m[i]$ is the additional noise introduced by the ADC Channel m, typically uncorrelated with others.
- M is the number of channels, when M→∞, $(\sum_{m=1}^{M} n_m[i])/M$ → 0, and the measurement→x[i]+n[i]
- Uncorrelated noise reduction rate: $10\log_{10}(M)$ dB

Block Diagram

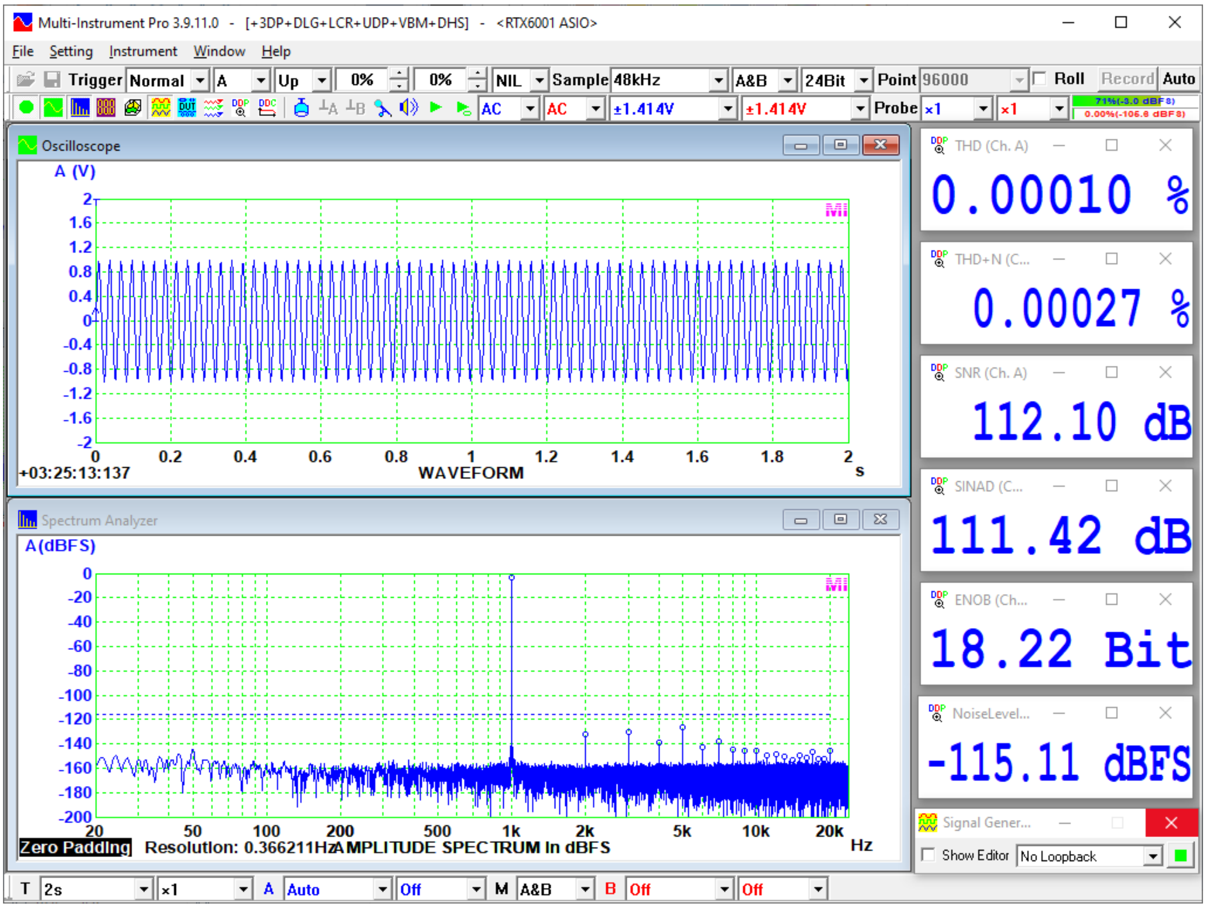
The objective of this method is to reduce the noise of the measurement system rather than the noise in the original signal under test. It requires feeding the same signal under test into multiple channels of the measurement system simultaneously. To make the time-domain averaging feasible, the signals in these channels must be in phase. The noise in the signal, which needs to be measured along with the signal, remains in phase in each channel and will not be averaged out. In contrast, the noises added by the measurement system are typically uncorrelated and will be reduced by $10\log_{10}(M)$ dB, where M is the number of channels. For instance, when M=2, the noise reduction is about 3dB theoretically.

In Multi-Instrument, this can be done by going to [Setting]>[ADC Device]>"Channel Operation" and selecting "A=(iA+iB)/2 …", which means that the data in the logical Channel A will be an average of the original data in the physical input Channel A and Channel B (see figure below).

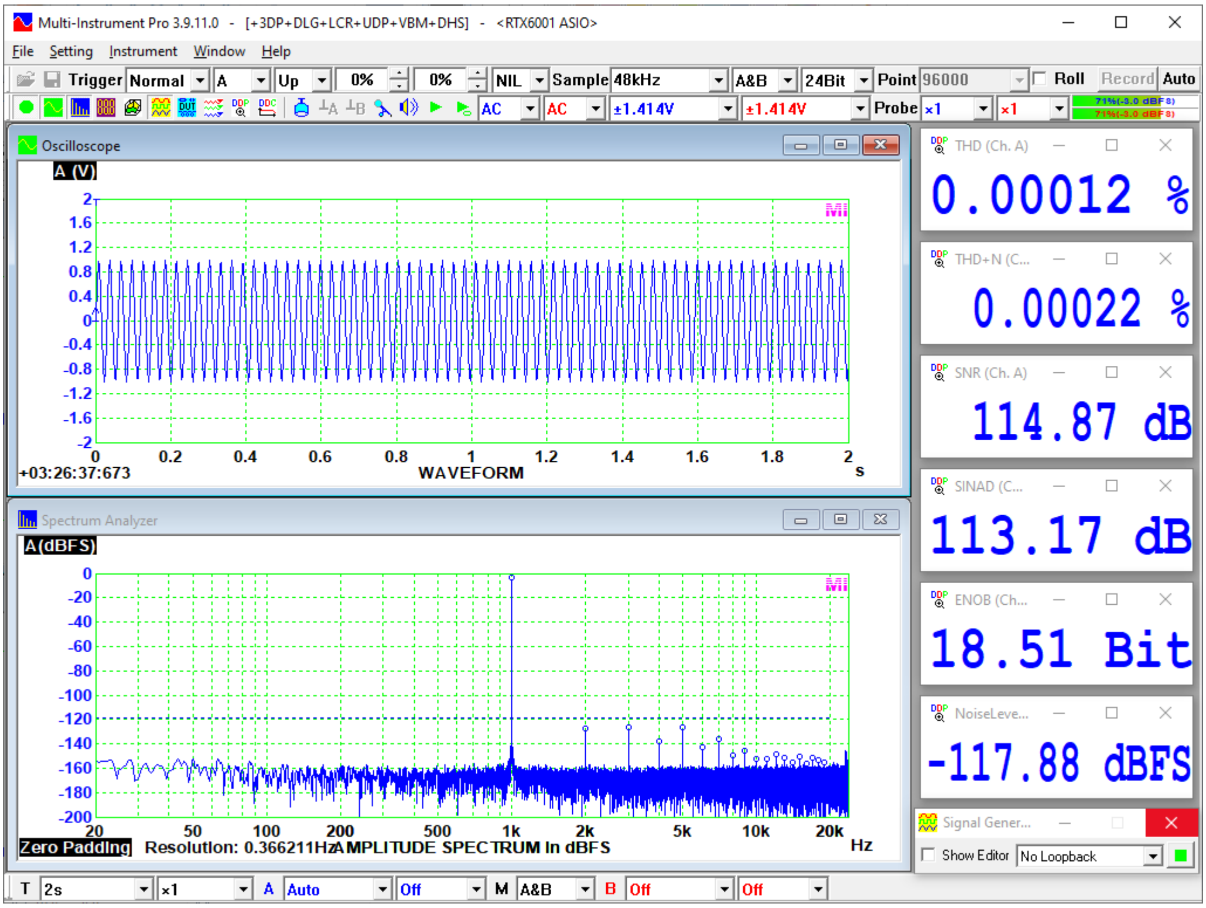


Time-Domain Dual-Channel In-Phase Averaging Configuration in Multi-Instrument

The following two figures show a 1kHz THD loopback test results of the RTX6001 audio analyzer with and without using this method. The measured noise levels are -115.11dBFS and -117.88dBFS, respectively. The noise reduction is about 2.8dB.



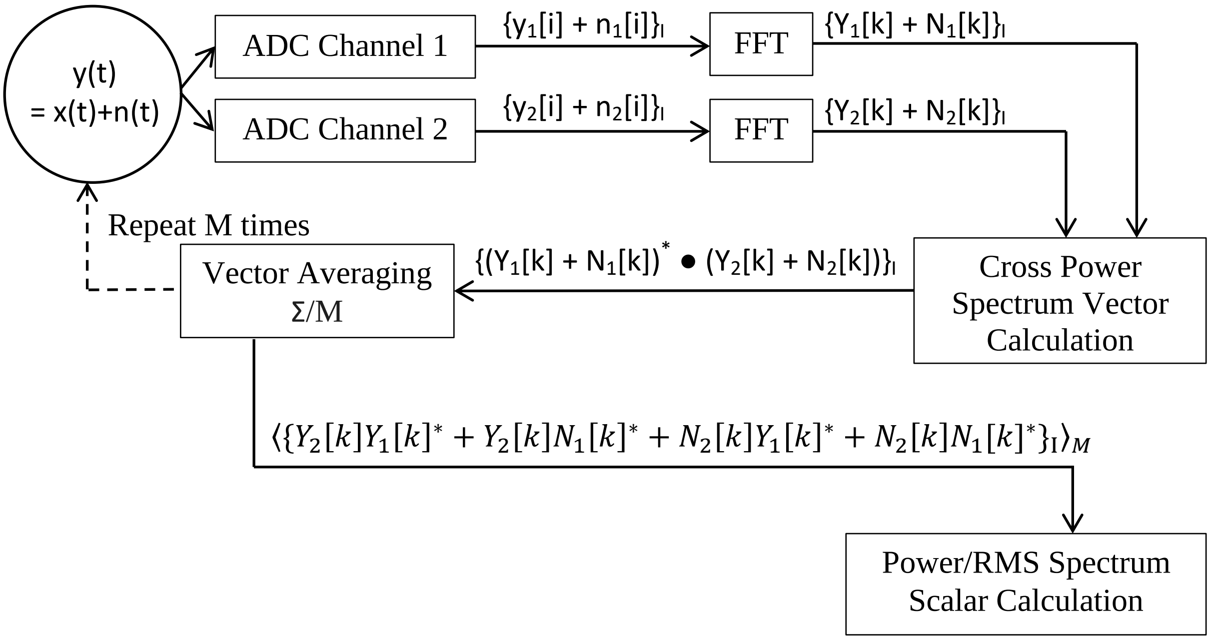
Before Dual-Channel In-Phase Averaging in Time Domain



After Dual-Channel In-Phase Averaging in Time Domain

The advantage of this method is that it reduces noise in both the time and frequency domains. However, further noise reduction requires additional channels, which increases costs.

3.2 Dual-Channel Cross Power Spectrum Vector Averaging in Frequency Domain

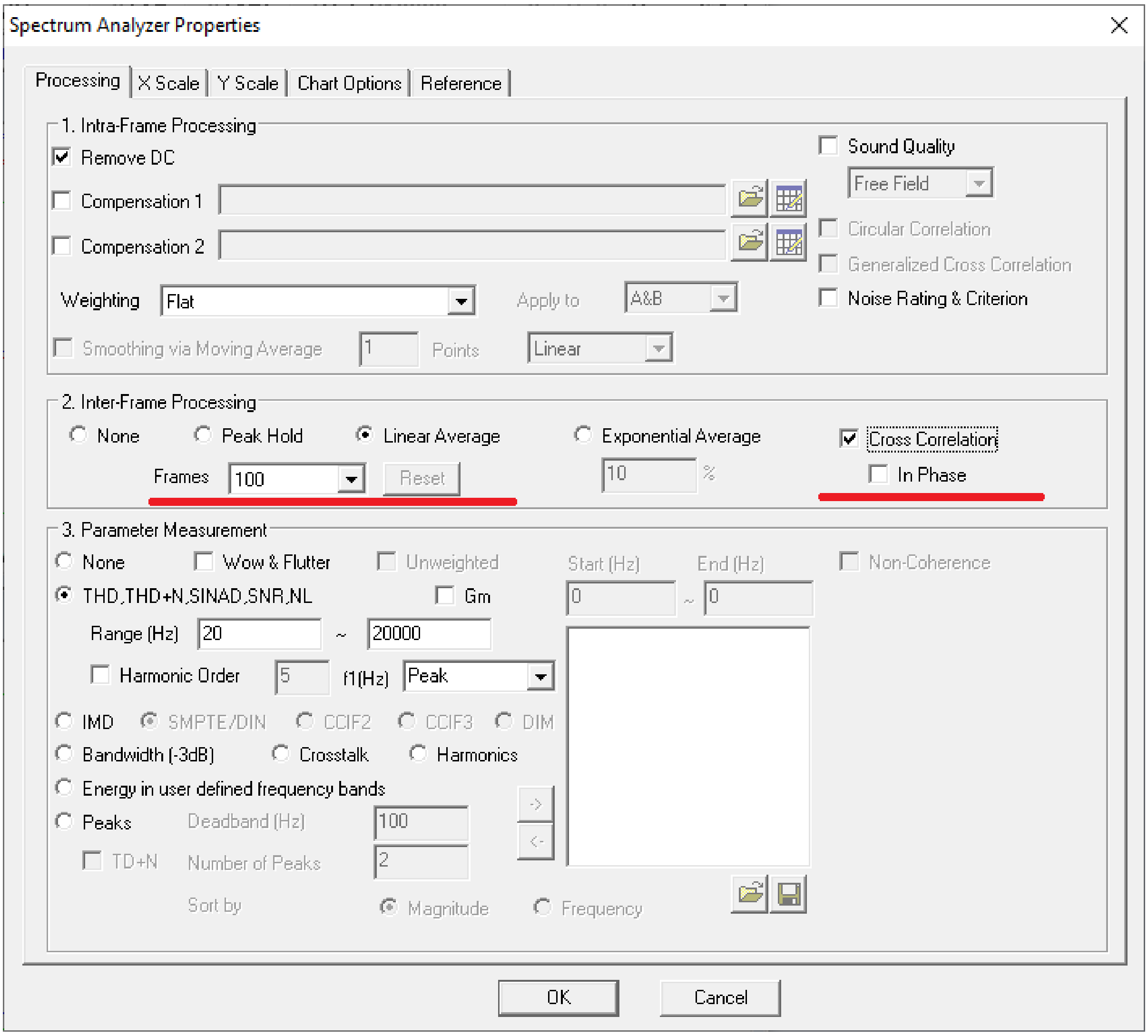


- y(t) is the analog signal to be measured containing signal x(t) and noise n(t), and $y_1[i]$ and $y_2[i]$ are its sampled versions in ADC Channels 1 & 2, respectively, which are not necessarily in phase.
- $n_1[i]$ and $n_2[i]$ are the additional noise introduced by the ADC Channels 1 & 2, respectively, typically uncorrelated with others.
- {} represents a sequence of data, and I is the FFT Size and the number of elements in the sequence.
- $\{Y_1[k]\}_I$, $\{Y_2[k]\}_I$, $\{N_1[k]\}_I$ and $\{N_2[k]\}_I$ are the Fourier Transform of $\{y_1[i]\}_I$, $\{y_2[i]\}_I$, $\{n_1[i]\}_I$ and $\{n_2[i]\}_I$, respectively, i=0,1,…, I-1. They are all complex vectors. The superscript * denotes complex conjugate.
- < > denotes an ensemble average. M is the number of averages.
  When $M\to\infty$, $\langle\{Y_2[k]N_1[k]^* + N_2[k]Y_1[k]^* + N_2[k]N_1[k]^*\}_I\rangle_M \to 0$, and the measured power spectrum[k]→Power Spectrum of Y[k].
- Uncorrelated noise reduction rate: $5\log_{10}(M)$ dB

<u>Block Diagram</u>

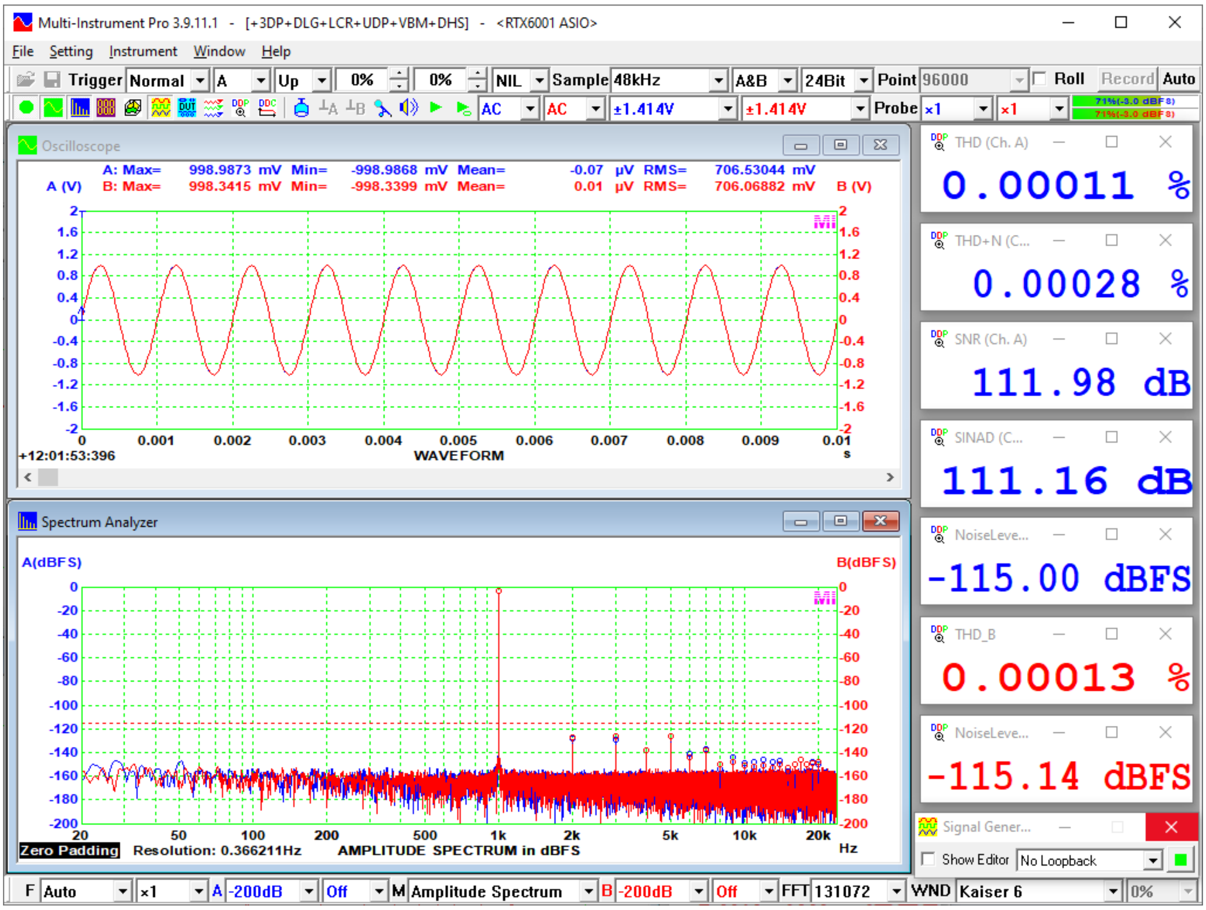
Just like the aforementioned method, the objective of this method is to reduce the noise of the measurement system rather than the noise in the original signal under test. It requires feeding the same signal under test into two channels of the measurement system simultaneously. Unlike the previously mentioned method, the signals in the two channels are not required to be in phase, thanks to the cross spectrum vector averaging algorithm. This method is often referred to as “Cross Correlation (Averaging)” due to the nature of the method and the fact that cross correlation and cross power spectrum are a Fourier Transform pair. The noise in the original signal, which needs to be measured along with the signal, remains untouched during “cross correlation” and will not be averaged out. In contrast, the noise added by the measurement system is typically uncorrelated in the two channels and will be reduced by $5\log_{10}(M)$ dB, where M is the number of averages. For instance, with M=100, the noise will be reduced by approximately 10 dB theoretically. Increasing the number of averages results in greater noise reduction. It is thus possible to measure a noise level significantly lower than the system's own noise floor after a sufficient number of averages.

In Multi-Instrument, this can be done by selecting the “Cross Correlation” option in the inter-frame processing of the Spectrum Analyzer (see figure below).



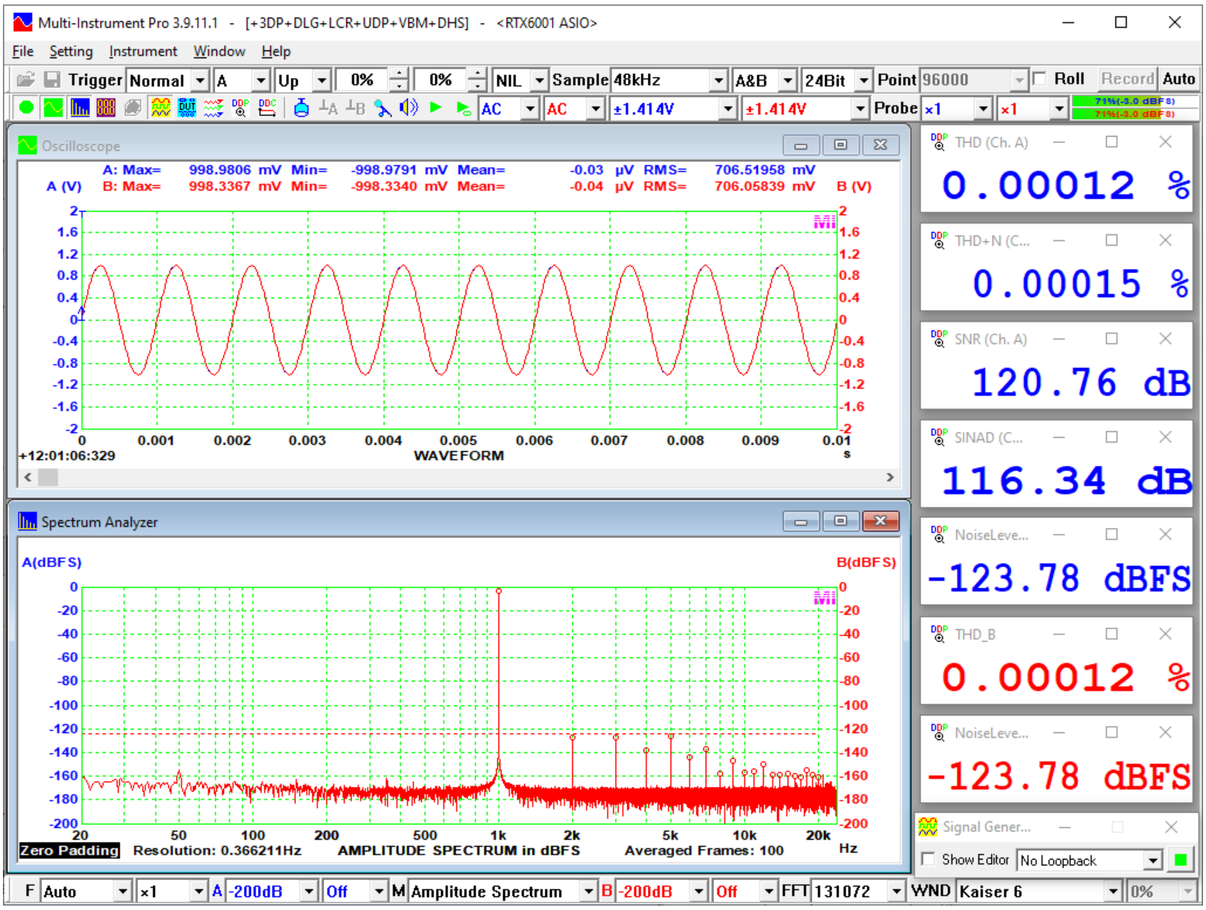
<u>"Cross-Correlation" Averaging Configuration in Multi-Instrument</u>

The following screenshot shows a 1kHz THD loopback test results of the RTX6001 audio analyzer without any averaging and distortion compensation. The THD and noise level values are 0.00011%, -115.00 dBFS for Channel A, and 0.00013%, -115.14 dBFS for Channel B.



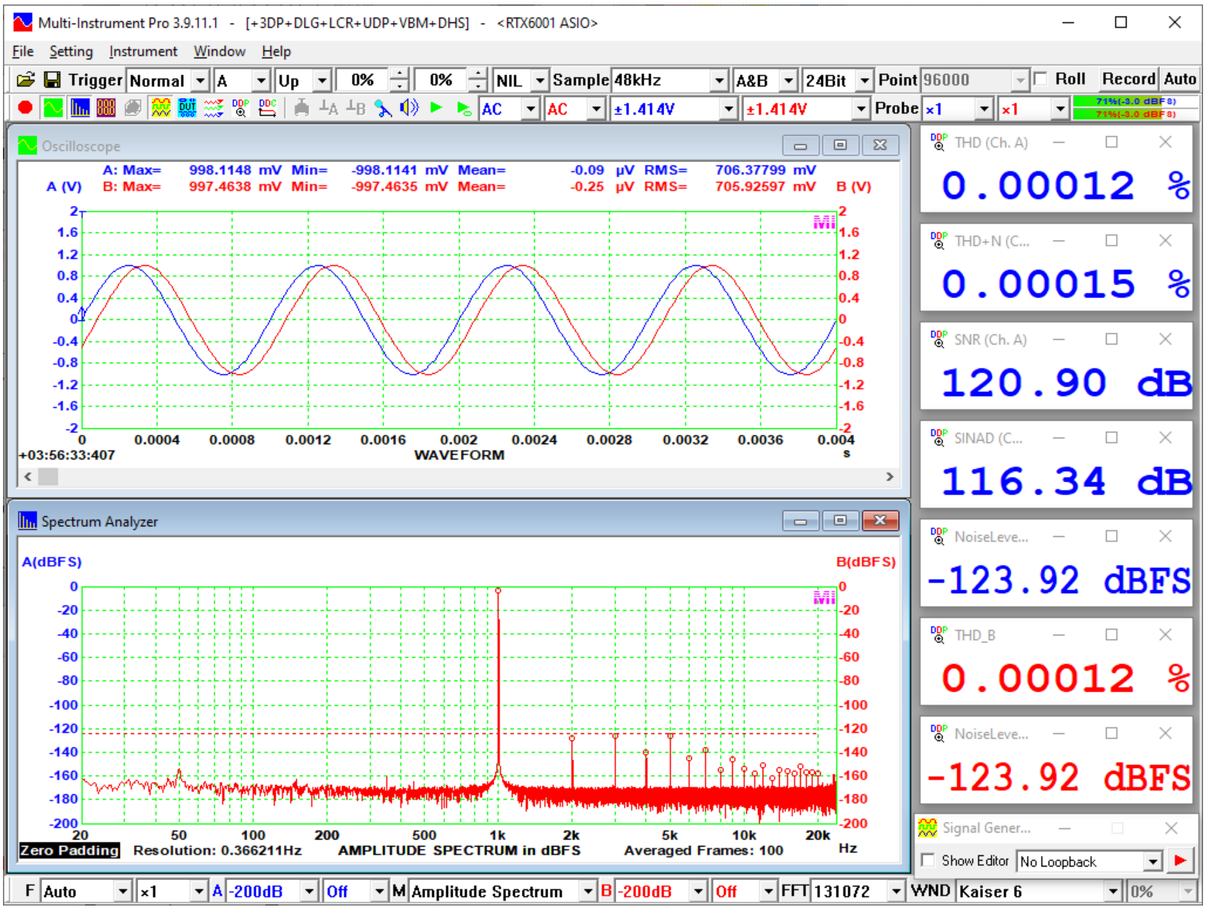
Before Cross Power Spectrum Vector Averaging in Frequency Domain

The same test was repeated using the "Cross-Correlation" method with 100 averages ("In Phase" option not used). The results are shown below. The measured THD is 0.00012%, which is about the average of the two independent channels. The noise level was measured to be -123.78 dB, a reduction of about 8.8 dB. Other noise-related parameters, including THD+N, SNR, SINAD, all show pronounced improvement.



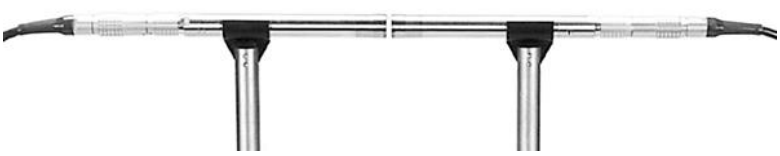
After Cross Power Spectrum Vector Averaging (100 times) in Frequency Domain

Please note that the "Cross Correlation" method in Multi-Instrument does not require the signals in the two channels to be in phase. This is advantageous, for example, when measuring extremely low sound pressure levels using a microphone pair, where the signals in the two channels may not be in-phase. The following figure shows the same test as above, except that the signals in the two channels are out of phase by 30º. Nearly no differences in the test results can be observed.



After Cross Power Spectrum Vector Averaging (100 times) in Frequency Domain with the Signals in the Two channels out of Phase by 30º

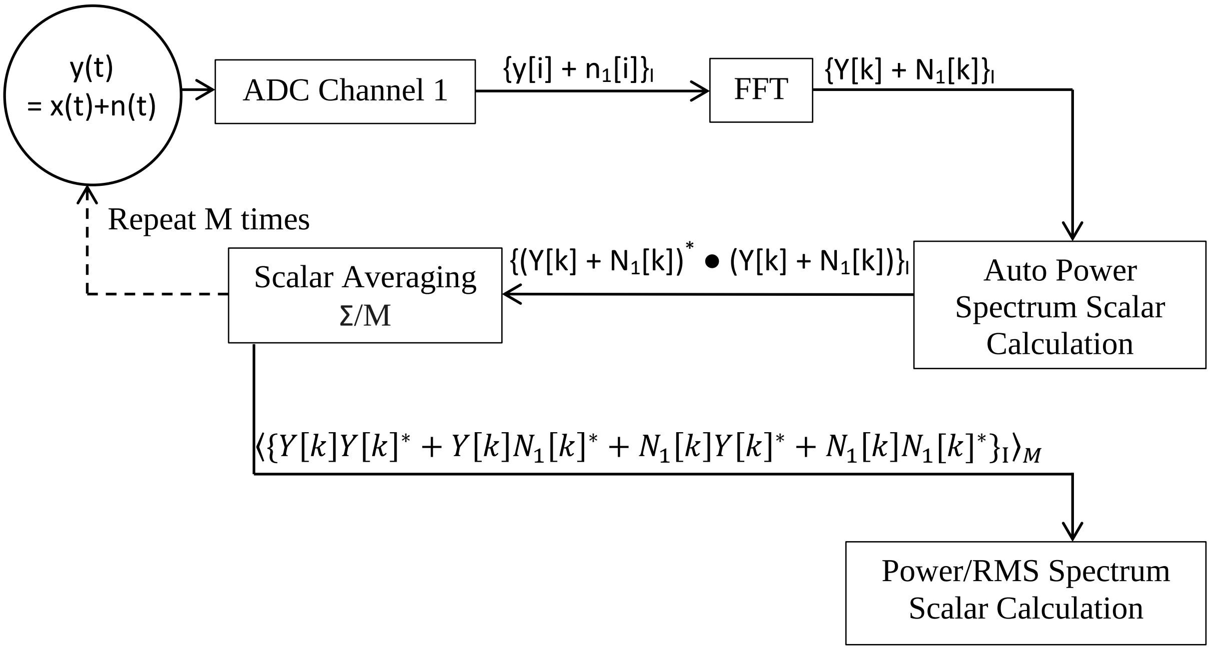
Measuring sound pressure levels below 20 dBA is very challenging because most ½” Class 1 measurement microphones have a noise floor between 14 and 20 dBA. However, by using a pair of microphones, such as those employed in sound intensity measurements or the VT RTA-268 series with a two-microphone configuration, the “Cross Correlation” technique can enable the measurement of sound pressure levels significantly lower than the noise floor of the individual microphones and the measurement system. The two microphones should be placed closely and face-to-face so that they measure the same acoustic field signal (see figure below). The distance between the microphone’s diaphragms should be less than ¼ of the wavelength of the highest frequency of interest. For lower frequencies, the microphones can also be placed side-by-side. Unlike in sound intensity measurements, stringent phase matching is not required for the two microphones.



Face-to-face Microphone Pair for Ultra-low Sound Pressure Level Measurement

If the signals in the two channels are strictly in phase or strictly with inverted polarities, an "In Phase" option is provided in Multi-Instrument, which will improve the noise reduction performance by $5\log_{10}(2)$=1.5dB.

3.3 Auto Power Spectrum Averaging in Frequency Domain

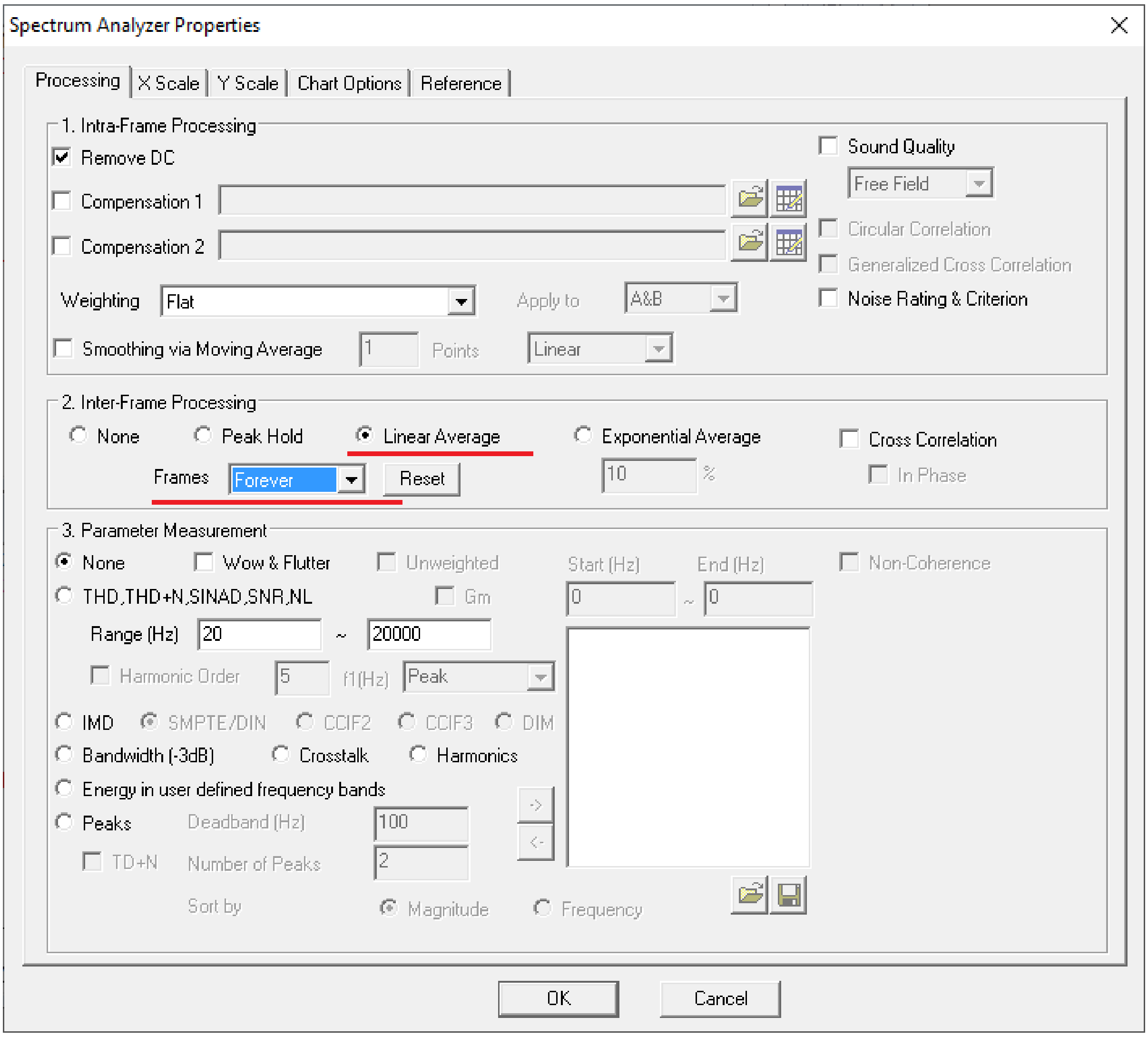


- y(t) is the analog signal to be measured containing signal x(t) and noise n(t), and y[i] is its sampled version.
- $n_1[i]$ is the additional noise introduced by the ADC Channels 1, typically uncorrelated with others.
- {} represents a sequence of data, and I is the FFT Size and the number of elements in the sequence.
- $\{Y[k]\}_I$ and $\{N_1[k]\}_I$ are the Fourier Transform of $\{y[i]\}_I$ and $\{n_1[i]\}_I$ respectively, i=0, 1, …, I-1. They are all complex vectors. The superscript * denotes complex conjugate.
- < > denotes an ensemble average. M is the number of averages.
  When M→∞, $\langle\{Y[k]N_1[k]^* + N_1[k]Y[k]^*\}_I\rangle_M$→0, and the measured power spectrum [k]→(Power Spectrum of Y[k] + Power Spectrum of $N_1[k]$).
- No absolute noise level reduction.

<u>Block Diagram</u>

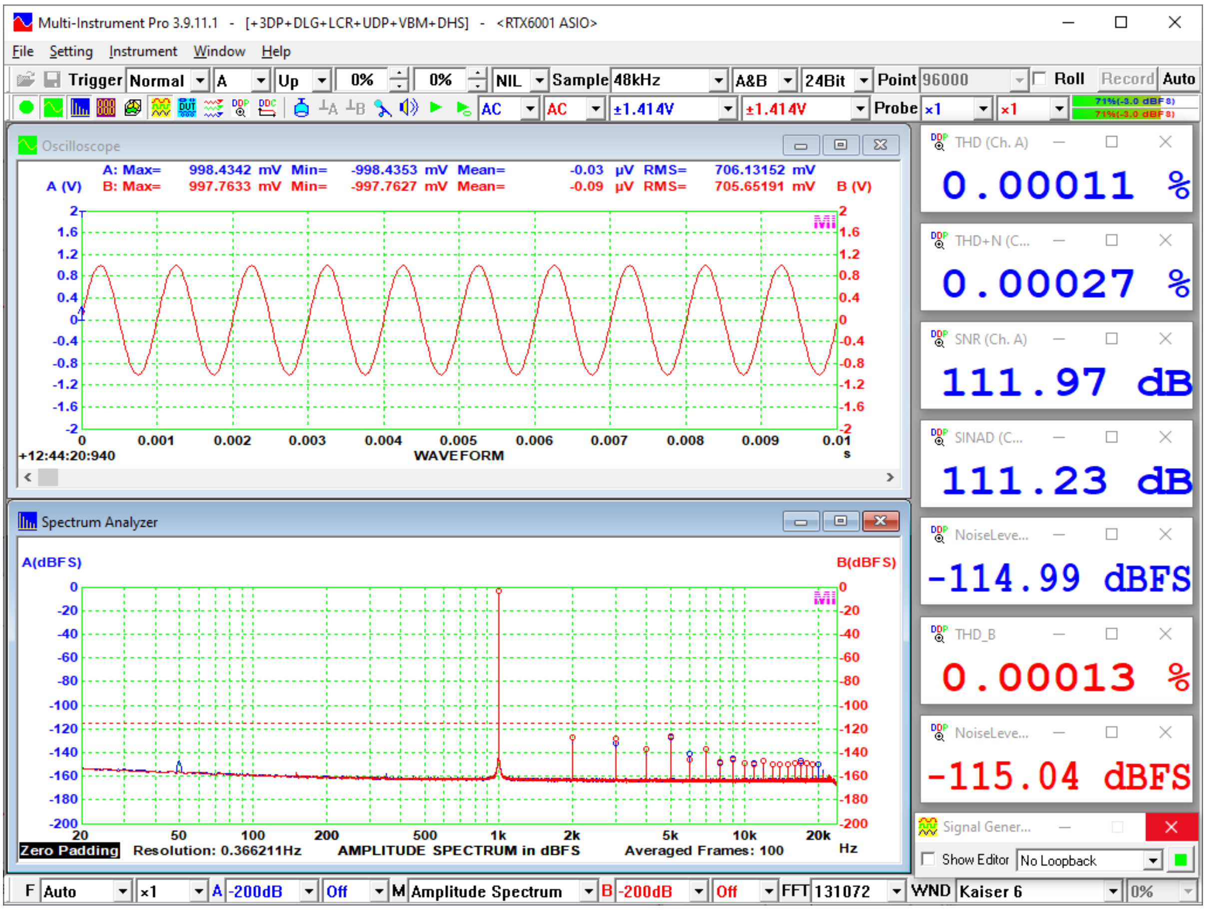
Auto power spectrum averaging is a common method used in many spectrum analysis software applications. It does not require cross-channel operations. The goal is to average the power spectra during measurement, which reduces the spectrum fluctuations but does not lower the absolute noise level. Sporadic interferences will still be averaged out by this method.

In Multi-Instrument, this can be done by selecting "Linear Average" without ticking the "Cross Correlation" in the inter-frame processing of the Spectrum Analyzer (see figure below).



Auto Power Spectrum Averaging Configuration in Multi-Instrument

The following screenshot shows a 1kHz THD loopback test results of the RTX6001 audio analyzer using this method with 100 averages. Compared with the results of the same test without averaging and distortion compensation in the previous section, it can be observed that there are nearly no changes in the measured THD (0.00011% and 0.00013%) and noise levels (-114.99 dBFS and -150.04 dBFS), except that the noise floor becomes much cleaner, with no significant fluctuations.

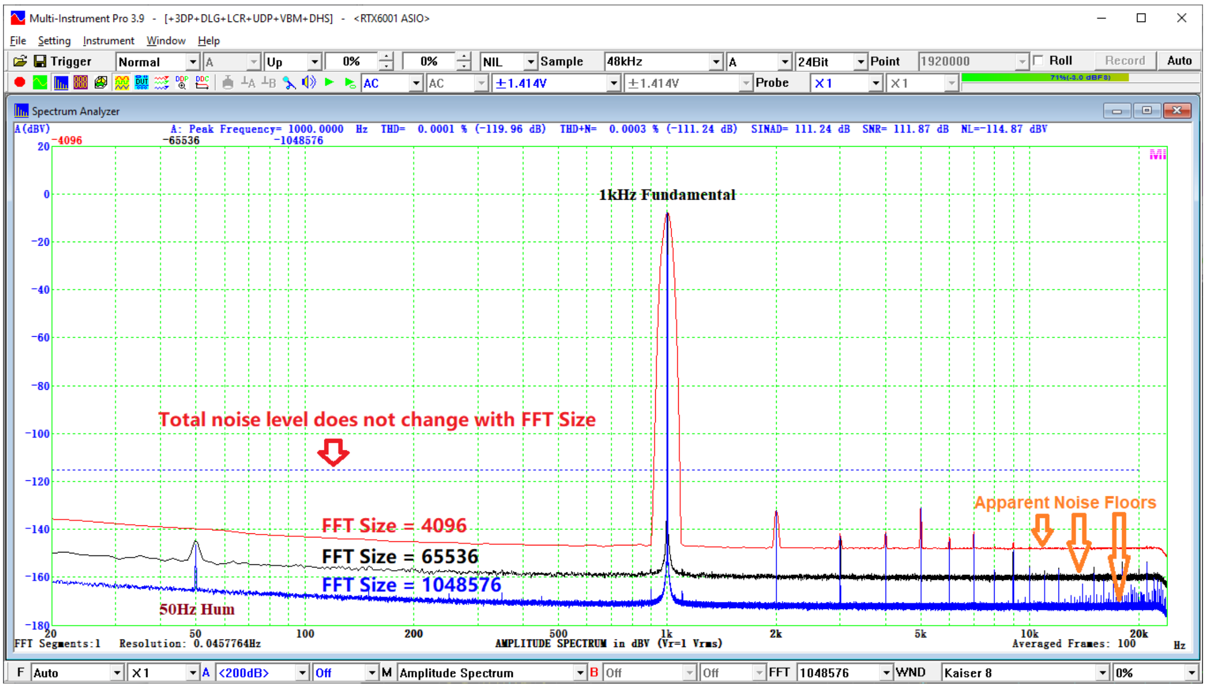


After Auto Power Spectrum Averaging (100 times) in Frequency Domain

3.4 Extracting Periodic Components Buried Under Noise Floor by Increasing Frequency Resolution in Frequency Domain

This method improves the measurement accuracy of periodic components by sharpening the FFT frequency resolution without reducing the overall noise level. The frequency resolution of FFT, also known as FFT bin width, is determined by the ratio [Sampling Rate]/[FFT Size], assuming no zero padding is used. A larger FFT size results in narrower FFT bins, which contain less noise energy per bin, thereby lowering the apparent noise floor while keeping the total noise level unchanged, as illustrated in the figure below.

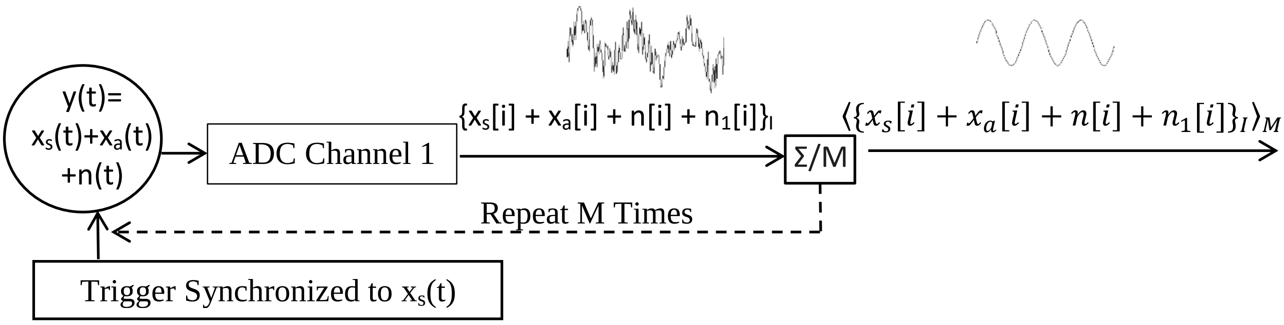
The figure shows three 1kHz THD measurement results where only the FFT size differs. Each time the FFT size is doubled, the apparent noise floor drops by about 3 dB. Therefore, when the FFT size is increased from 4096 to 65,536 and then to 1,048,576, the apparent noise floor drops by 12 dB with each increase, totaling a 24 dB reduction. This effect can be clearly observed in the figure. Unlike the wide-band noise, the height of a periodic single-frequency peak does not change with the FFT size, because its energy remains concentrated in the FFT bin(s) corresponding to that frequency, rather than being spread along the frequency axis. From the figure, it can be observed that the heights of the fundamental and harmonics remain unchanged across the three cases. This phenomenon is often leveraged to make small-amplitude periodic components that may have been buried by the noise floor at lower FFT size, more detectable and pronounced by increasing the FFT size. This effect is particularly evident in the case of the 50Hz hum, which becomes discernible and then prominent in the figure as the FFT size increases.



Comparison of Harmonic Distortion Spectra with Different FFT Sizes (No Zero Padding)

By the way, the width of a periodic single-frequency peak is influenced by spectral leakage and results from the convolution between the single frequency and the window function spectrum. The width decreases as FFT size increases to conserve the energy of the single-frequency component.

3.5 Time-domain Synchronous Averaging



- $x_s(t)$ is the analog periodic component synchronous with the trigger, and $x_s[i]$ is its sampled version.
- $x_a(t)$ is the analog periodic component asynchronous with the trigger, and $x_a[i]$ is its sampled version.
- $n(t)$ is the analog noise that comes with $x_s(t)$ and $x_a(t)$, and $n[i]$ is its sampled version.
- $n_1[i]$ is the additional noise introduced by the ADC Channel 1.
- { } represents a sequence of data, and I is the number of samples in the sequence.
- < > denotes an ensemble average. M is the number of averages.

When $M\to\infty$, $\langle\{x_a[i] + n[i] + n_1[i]\}_I\rangle_M \to 0$, and the measurement $\to \langle\{x_s[i]\}_I\rangle_M$

- Noise reduction rate: $10\log_{10}(M)$ dB. The reduction rate of the asynchronous periodic component depends on its asynchrony relative to the trigger.

Block Diagram

Time-domain synchronous averaging, often referred to as Time Synchronous Averaging (TSA) or simply Synchronous Averaging, is a powerful technique for extracting harmonically related periodic components from a composite signal that may also contain other periodic components and noise. To ensure that the time-domain averaging is meaningful, the start of the waveforms to be averaged must be synchronized with the fundamental frequency of the harmonically related periodic components to be extracted. The synchronization is achieved through proper triggering, either by using the signal itself, as long as a stable trigger point synchronized with the fundamental frequency in question can be identified, or by employing an external signal, such as a trigger pulse, that is synchronized with the periodic components to be extracted. The averaging process will gradually eliminate the random noise and any periodic components not synchronized with the trigger, leaving only the periodic components that are synchronized with it. The noise will be reduced by $10\log_{10}(M)$ dB, where M is the number of averages. For instance, when M=100, the noise reduction is about 20dB theoretically. The reduction rate of the asynchronous periodic component depends on its asynchrony relative to the trigger. The frequency response of TSA can be derived as follows.

In the time domain,

$$tsa(t) = \frac{1}{M}\sum_{m=0}^{M-1} x(t - mT)$$

where M is the number of averages, and T the time interval between averages. Taking the Fourier transform of both sides yields:

$$TSA(f) = \frac{1}{M}\sum_{m=0}^{M-1} X(f)e^{-j2\pi fmT} = \frac{1}{M}X(f)\sum_{m=0}^{M-1} e^{-j2\pi fmT}$$

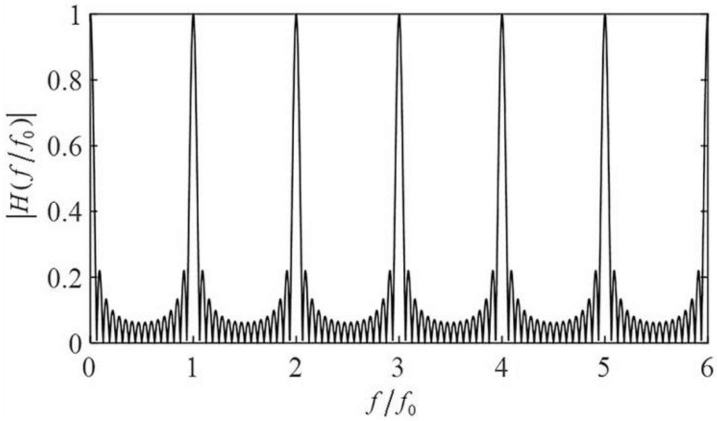
$$= \frac{1}{M}X(f)\frac{1-e^{-j2\pi fMT}}{1-e^{-j2\pi fT}} = \frac{1}{M}X(f)\frac{e^{j\pi fMT}-e^{-j\pi fMT}}{e^{j\pi fT}-e^{-j\pi fT}}e^{-j\pi f(M-1)T}$$

$$= \frac{1}{M}X(f)\frac{\sin(\pi fMT)}{\sin(\pi fT)}e^{-j\pi f(M-1)T} = \frac{1}{M}X(f)\frac{\sin(\pi Mf/f_0)}{\sin(\pi f/f_0)}e^{-j\pi(M-1)f/f_0}$$

where f_0 =1/T, the fundamental frequency of the periodic components to be extracted. The magnitude frequency response $|H(f/f_0)|$ is thus:

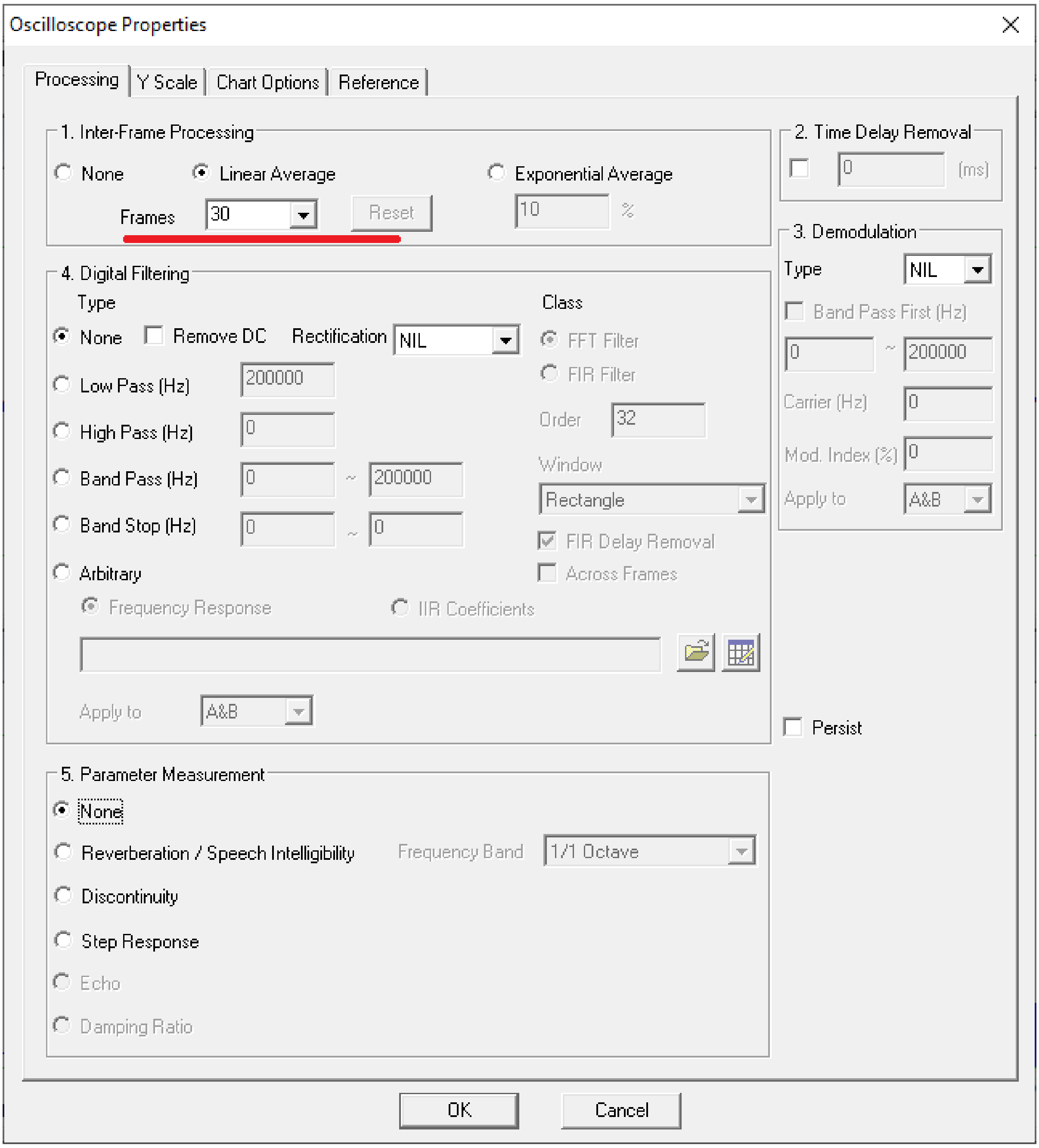
$$|H(f/f_0)| = \frac{1}{M}\frac{\sin(\pi Mf/f_0)}{\sin(\pi f/f_0)}$$

It has a form of a comb filter, with main lobes centered at the integer multiples of the fundamental frequency f_0 (see figure below). Increasing the number of averages M results in narrower main lobes and more side lobes. Only the fundamental frequency and its harmonics will be kept untouched and all the other frequencies will be attenuated.



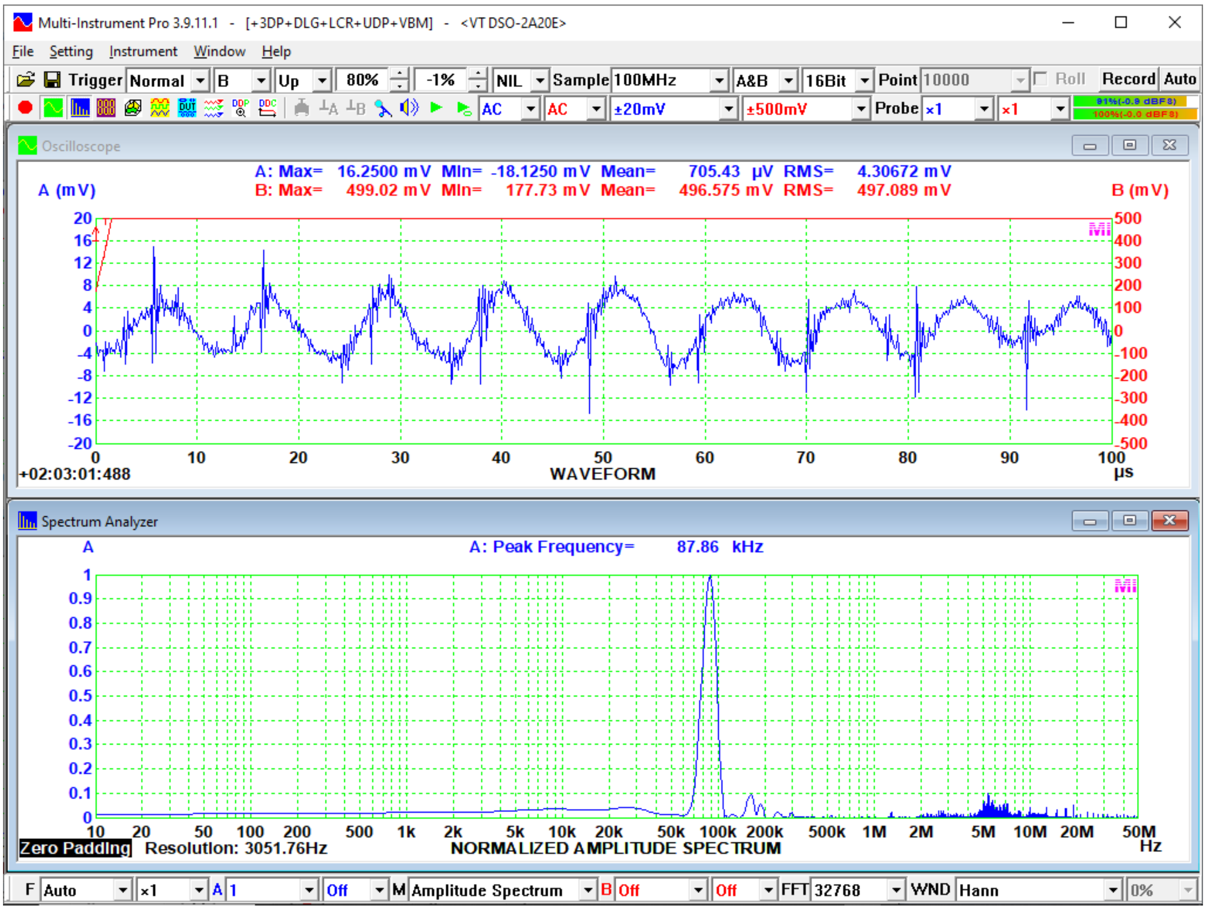
Magnitude Frequency Response of TSA

In Multi-Instrument, time-domain synchronous averaging can be configured by right clicking the Oscilloscope window and selecting [Oscilloscope Processing]> “Inter-frame Processing”> “Linear Average”, as shown in the figure below. Trigger Mode must be set to “Normal” and Trigger Source, Trigger Edge, Trigger Level, Trigger Delay, Trigger Frequency Rejection must be configured properly to ensure accurate and stable triggering. The sampling rate should be high enough to minimize the uncertainty in the timing of the synchronous triggering so as to reduce the smearing of the waveform.



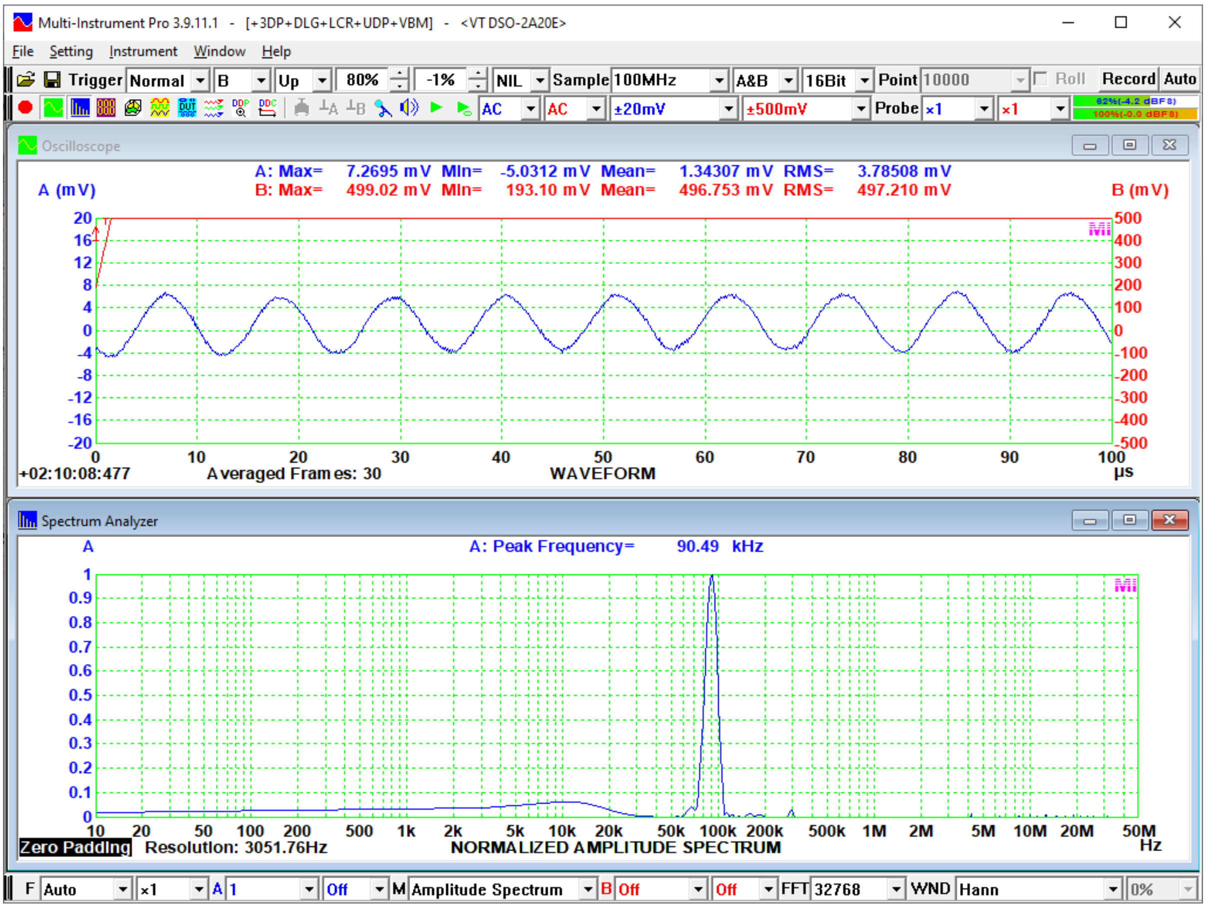
Time-Domain Synchronous Averaging Configuration in Multi-Instrument

The following is an example of a 5 mVp 90 kHz sinewave sampled by VT DSO-2A20E at 100MHz in Channel A. The 90 kHz signal was interfered significantly with some unsynchronized periodic signals and contaminated by a large amount of random noise, as shown in the figure below.



<u>Before Time-Domain Synchronous Averaging</u>

Luckily, the 90 kHz signal in Channel A was accompanied by a synchronous 100Hz rectangular pulse signal in Channel B, which was used as the trigger source. The following figure shows the results after 30 time-domain synchronous averages, displaying a much cleaner waveform.

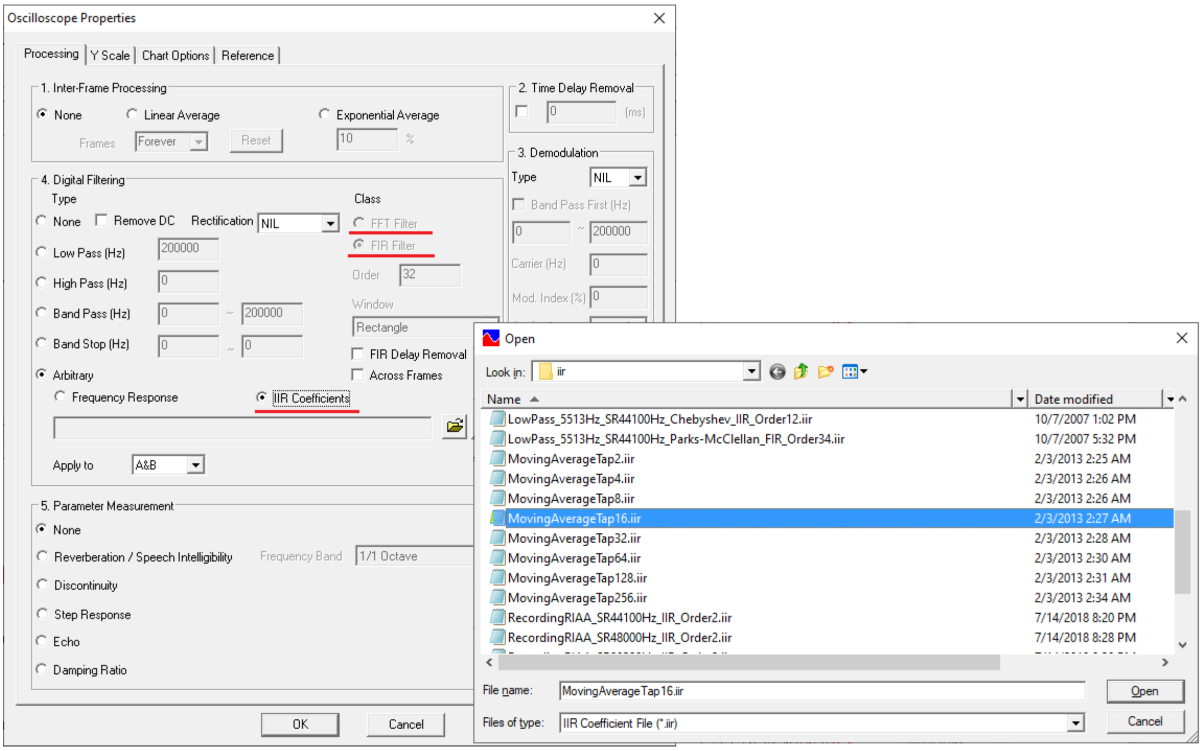


After 30 Time-Domain Synchronous Averages

Time-domain synchronous averaging is a common method employed in rotating machinery diagnosis for extracting periodic waveforms related to the rotation under test from interfering periodic signals and noise. It is also often used to recover the response to repetitively applied stimuli, improving the signal-to-noise ratio when the response is embedded in noise and other interferences, as in the direct measurements of acoustic impulse responses and evoked responses in EEG (Electroencephalography).

3.6 Time-domain Digital Filtering and Moving Averaging

Digital filtering is a common method for removing out-of-band noise and interference. Multi-Instrument supports IIR, FIR, and FFT filtering in the time domain, in addition to frequency-domain filtering and smoothing. Moving averaging is supported in Multi-Instrument using IIR coefficients (see figure below).



Configuration of Moving Averaging in Multi-Instrument

Moving averaging is a type of low-pass filter. Its frequency response can be derived as follows:

$$ma(i) = \frac{1}{M}\sum_{m=0}^{M-1} x(i-m)$$

where M is the number of adjacent samples used in the averaging. Taking the Z transform of both sides yields:

$$MA(z) = \frac{1}{M}\sum_{m=0}^{M-1} X(z)z^{-m} = \frac{1}{M}X(z)\frac{1-z^{-M}}{1-z^{-1}}$$

So the frequency response is:

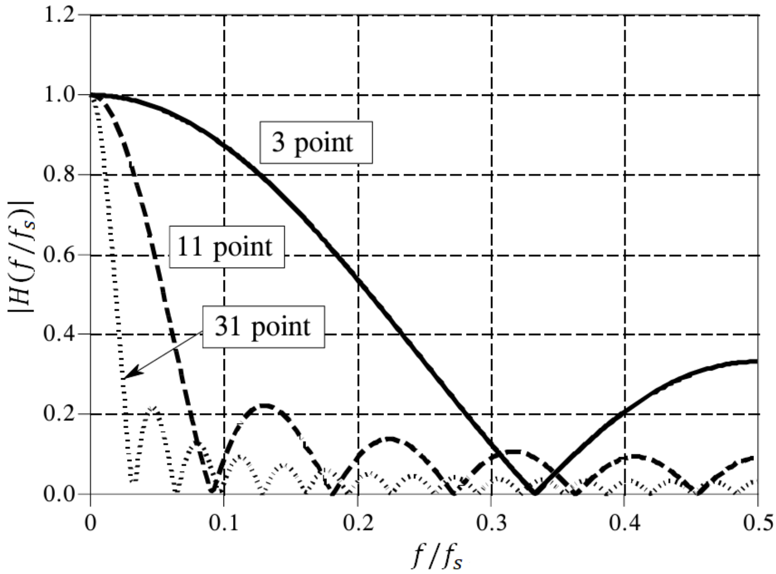
$$H(z) = \frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}$$

Let $z=e^{j2\pi f/fs}$, where f_s is the sampling frequency, the frequency response would be:

$$H(f/f_s) = \frac{1}{M}\frac{1-e^{-j2\pi Mf/f_s}}{1-e^{-j2\pi f/f_s}} = \frac{1}{M}\frac{\sin(\pi Mf/f_s)}{\sin(\pi f/f_s)}e^{-j\pi(M-1)f/f_s}$$

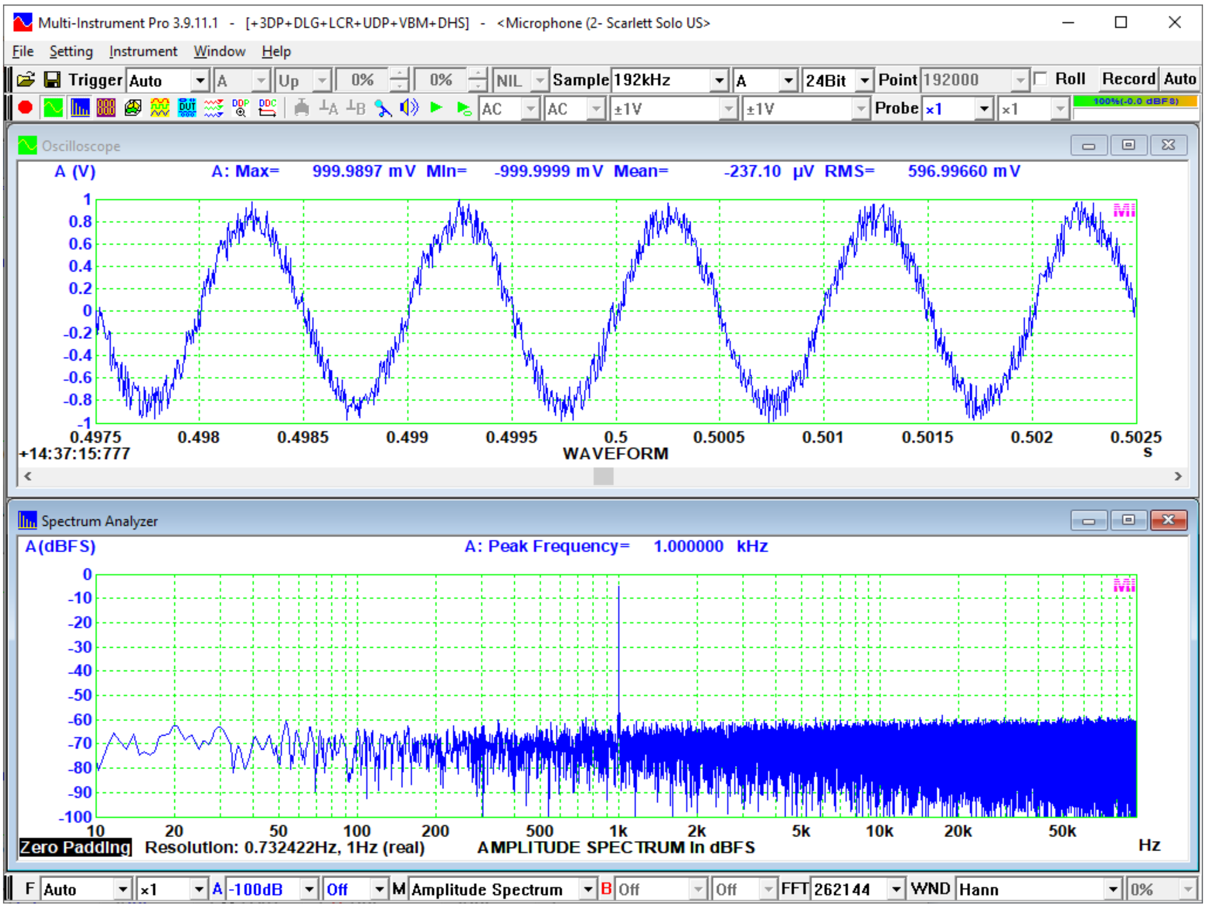
$$|H(f/f_s)| = \frac{1}{M}\frac{\sin(\pi Mf/f_s)}{\sin(\pi f/f_s)}$$

The magnitude frequency responses for M=3, 11, 31 are plotted in the figure below. The bandwidth decreases with the number of averages. Moving averaging performs exceptionally well for smoothing in the time domain, but is not as effective as typical low-pass filters in the frequency domain.



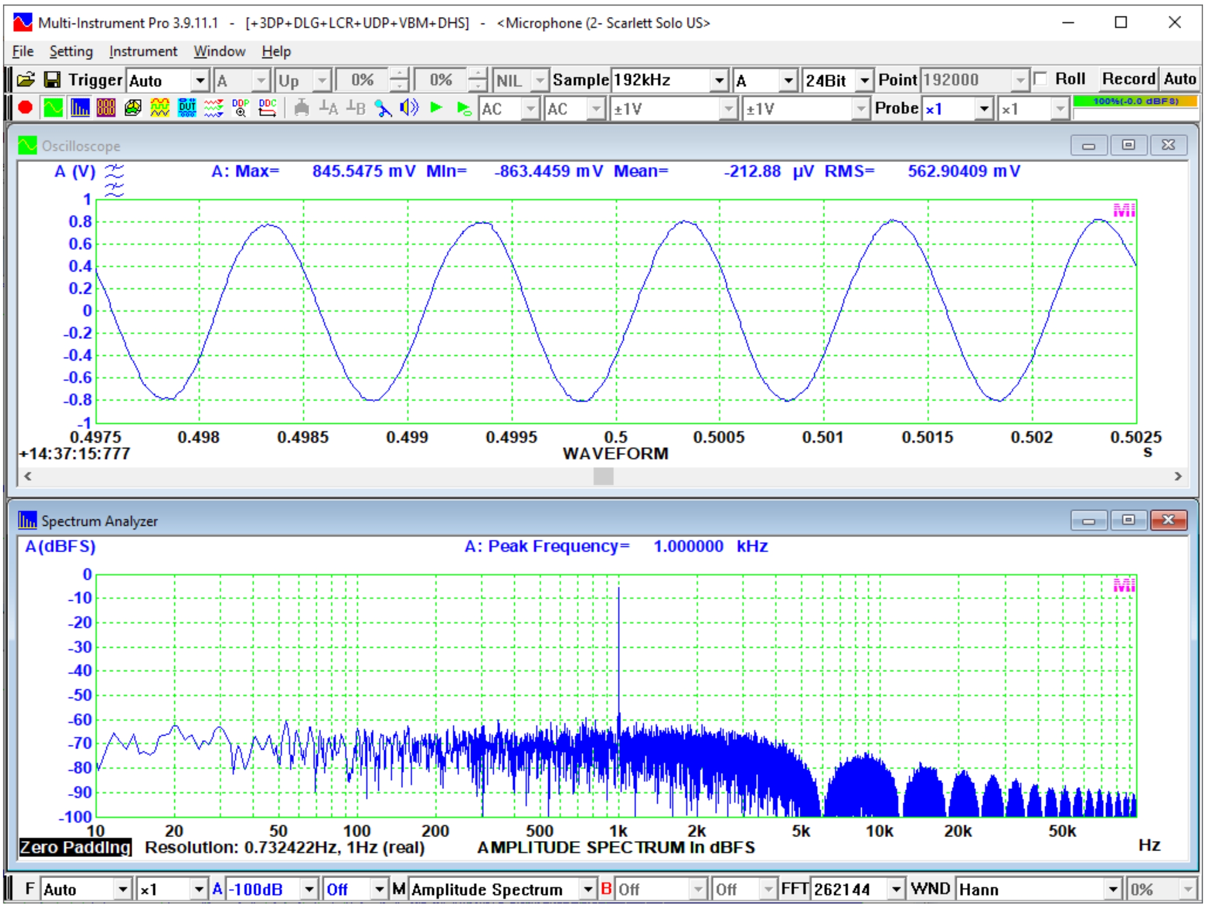
<u>Magnitude Frequency Response of Moving Averaging</u>

The following figure shows a 1 kHz sine wave mixed with white noise.



<u>Before Moving Averaging</u>

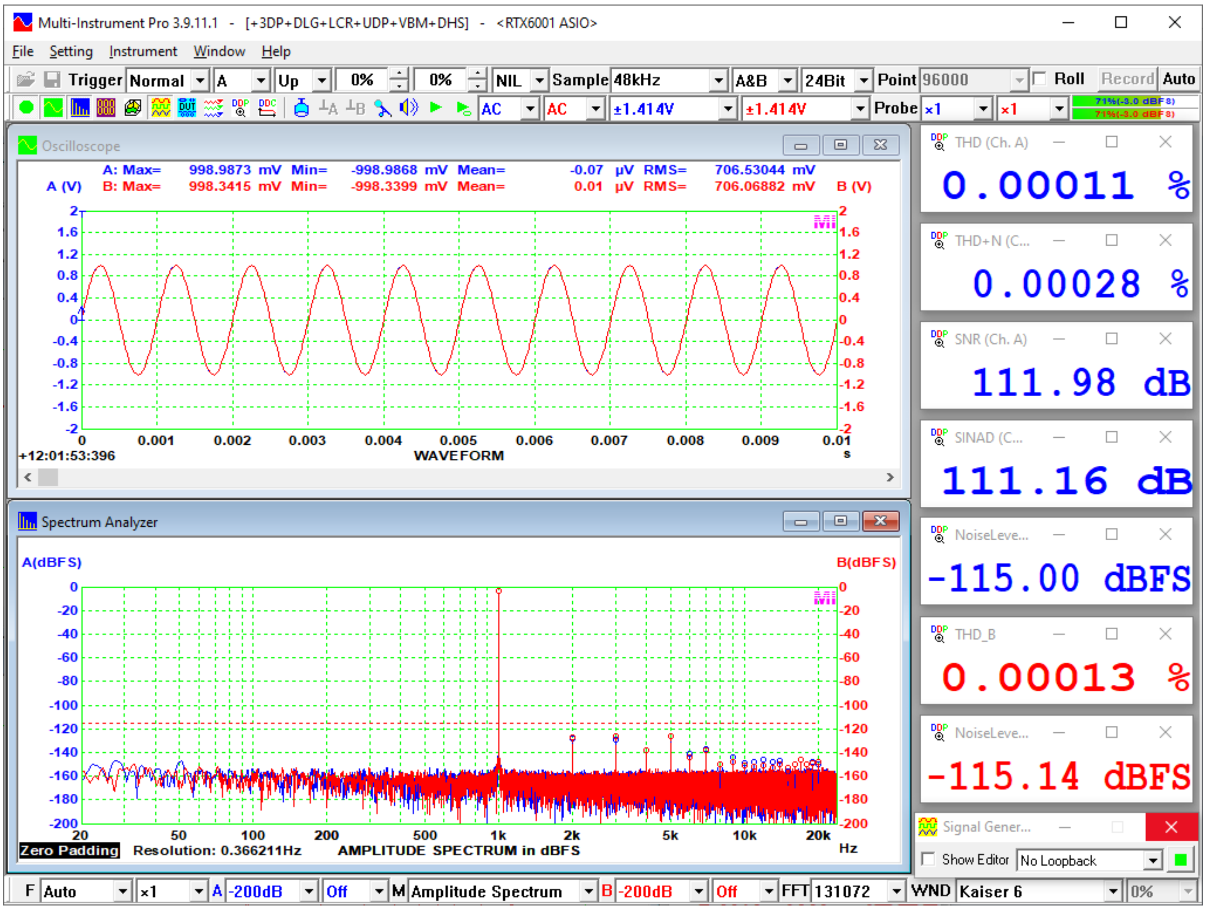
Its waveform becomes much smoother after applying a 32-point moving average (see figure below). The noise floor in the figure reflects the low-pass magnitude frequency response of the 32-point moving average.



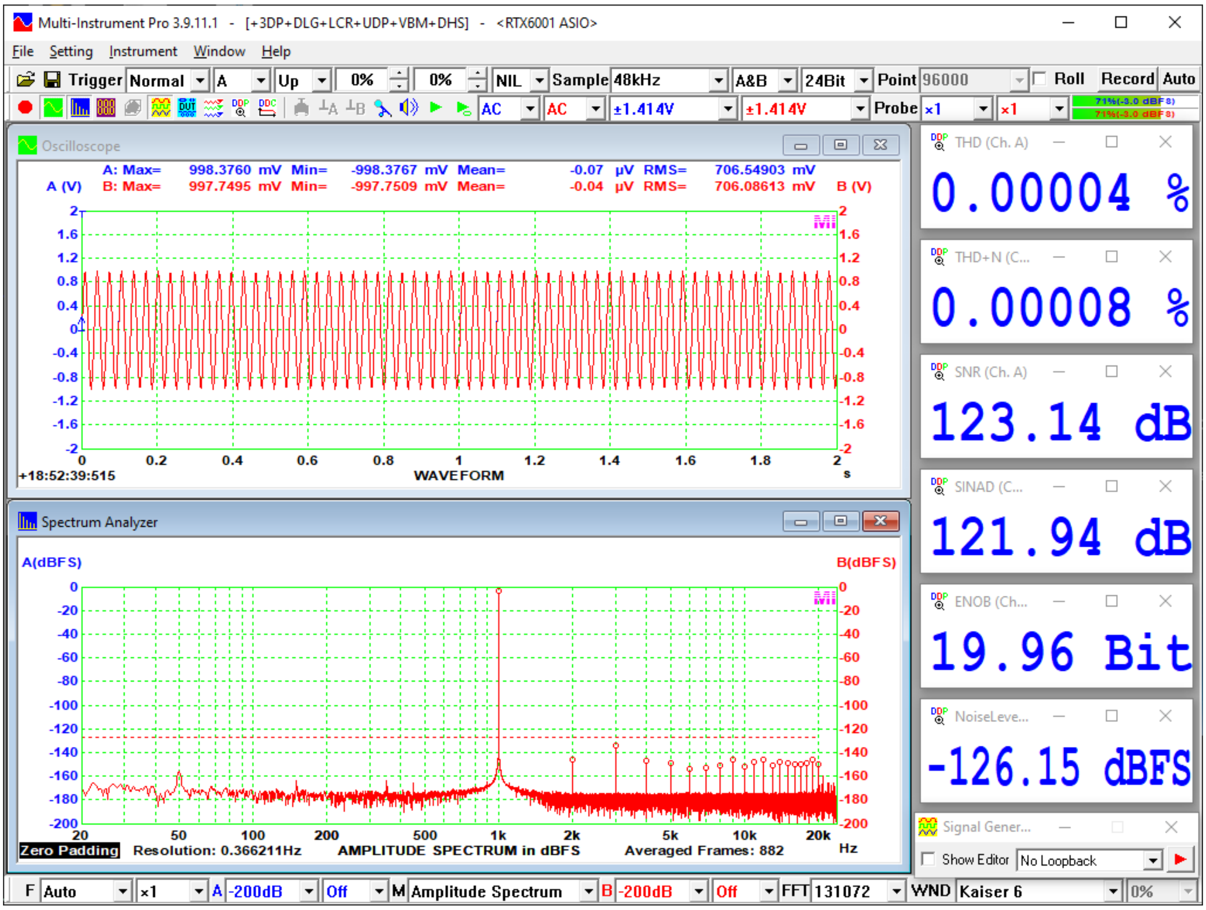
<u>After 32-point Moving Averaging</u>

4. Combination of Distortion Compensation and Noise Reduction Techniques

It is possible to combine the distortion compensation and noise reduction techniques together to achieve better overall performance. The following two figures show a 1 kHz THD loopback test results of the RTX6001 audio analyzer before and after applying the distortion compensation and "cross correlation" noise reduction techniques. It can be observed that the residual THD+N and THD drop from 0.00028% (-111.06 dB) and 0.00011% (-119.17 dB) to 0.00008% (-121.94 dB) and 0.00004% (-127.96 dB), respectively, representing improvements of 10.9dB and 8.8dB. In addition, the noise level is reduced from -115dBFS to -126dBFS after more than 800 averages ("in-phase" option not used), an 11dB improvement. Other noise-related parameters, including SNR and SINAD, all show pronounced improvement.



Before Distortion Compensation and Noise Reduction



After Distortion Compensation and Noise Reduction