FFT Basics and Case Study using Multi-Instrument

By Dr. Wang Hongwei

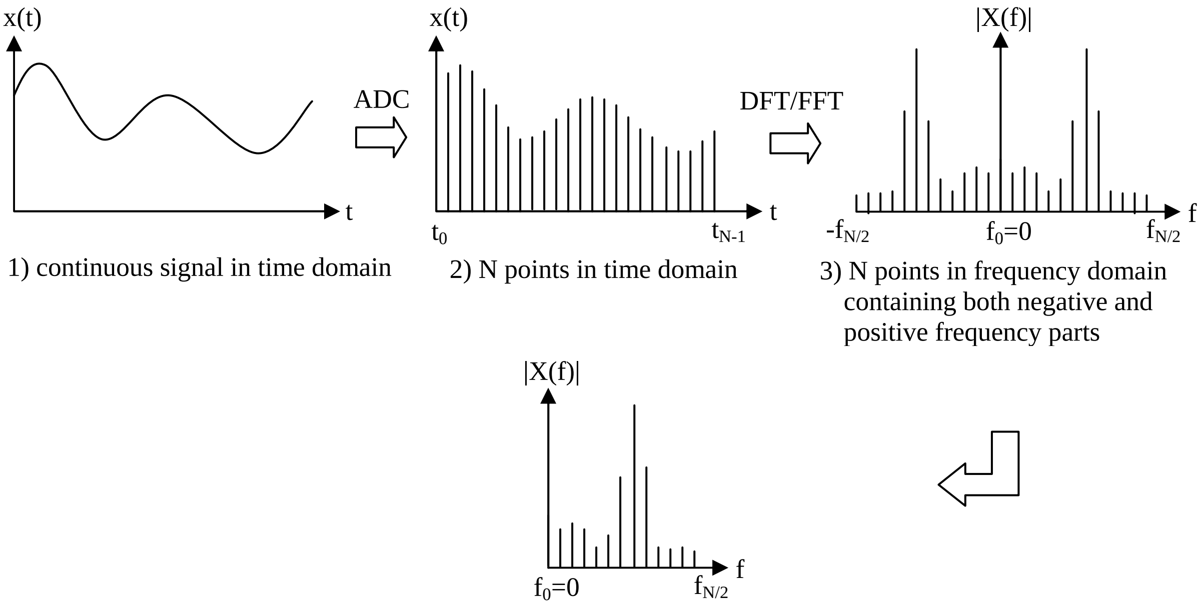
REV: 01
May 1, 2009

The objective of this article is to help FFT software users to understand the basics of FFT so that they can set the relevant FFT parameters correctly and optimally to achieve the best measurement quality in spectrum analysis. Sophisticated mathematics is intentionally avoided in this article in order to make it easily understood by most FFT software users.

TABLE OF CONTENTS

1. Sampling and FFT

A signal in the time domain can be converted to its counterpart in the frequency domain by means of Fourier Transform (FT). The signal must be sampled at discrete time by an A/D converter before it can be analyzed by a computer. Discrete Fourier Transform (DFT) can be used to convert the discrete signal (discrete in time) in the time domain to its counterpart (discrete in frequency) in the frequency domain. DFT can be computed efficiently in practice using a Fast Fourier Transform (FFT) algorithm, which is generally N/log(N)-1 times faster than DFT, where N is called DFT or FFT size, which is the number of data points used in the computation. To achieve maximum efficiency of computation in FFT, N is generally constrained to an integer power of two, e.g. 1024, 2048, 4096, 8192, etc..



1) continuous signal in time domain

2) N points in time domain

3) N points in frequency domain containing both negative and positive frequency parts

4) N/2+1 points in amplitude/power spectrum

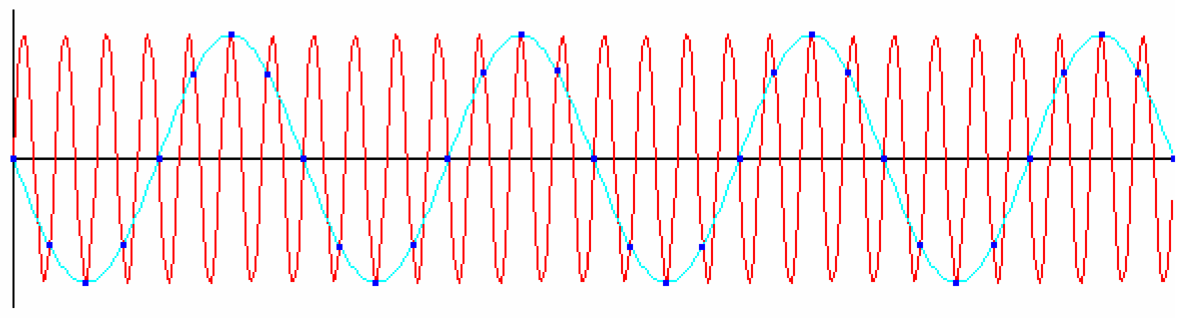
The above figure illustrates the aforementioned process. An N-point time record [$x(t_0)$, $x(t_1)$, …, $x(t_{N-1})$] will generate N points [$X(-f_{N/2})$, …$X(f_0)$, …, $X(f_{N/2})$] in the frequency domain containing both negative and positive frequency parts. The positive and negative frequency parts can be combined to produce N/2+1 points [$X(f_0)$, $X(f_1)$…, $X(f_{N/2})$] at real frequencies in the amplitude/power spectrum. These points are located at frequencies: 0, $\Delta f \times 1$, …, $\Delta f \times N/2$, where $\Delta f = f_s/N$, where f_s is the sampling frequency. The highest frequency measurable is $f_s/2$ and is called Nyquist frequency.

An important principle in digital signal processing is the "Nyquist-Shannon Sampling Theorem" which states that an analog signal that has been sampled can be perfecty reconstructed from the samples if the sampling frequency is greater than twice the highest frequency in the original signal.

There are three possible issues inherent in DFT or FFT, which may result in errors if no proper precautions are taken. They are aliasing, leakage, and picket fence effect.

2. Aliasing

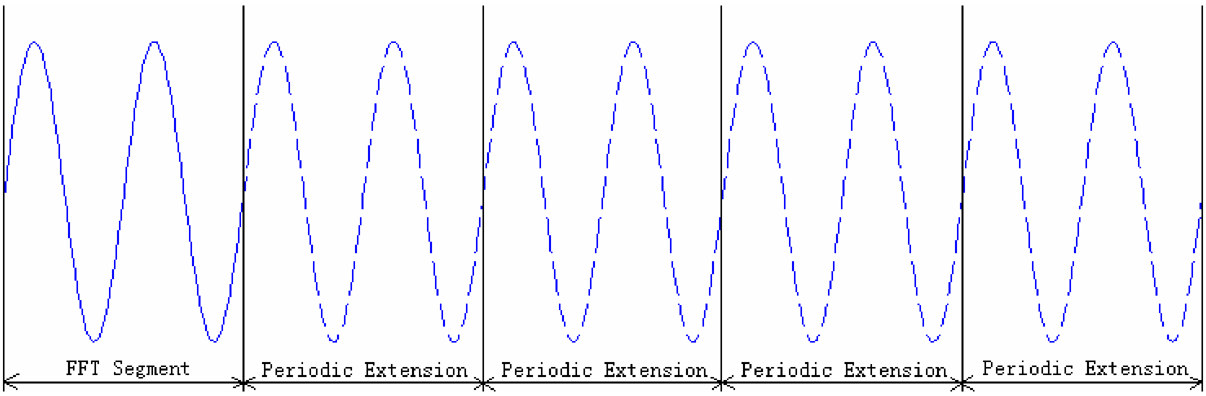
Aliasing occurs when a signal is sampled at less than twice of the highest frequency present in the signal. It causes frequency components that are higher than half of the sampling frequency to "fold over" and overlap with the lower frequency components. For example, an aliased frequency f in the range of $f_s/2$~ f_s becomes f' = f_s – f. That is, if a 7 kHz sinusoidal signal is sampled at 8 kHz, then it will be shown as 1 kHz in the spectrum due to aliasing. From the figure below, it can be seen that a 7 kHz sinusoidal signal (red) and a 1 kHz sinusoidal signal (cyan) share exactly the same set of sampled data points if sampled at 8 kHz, thus they cannot be distinguished with each other after being sampled.



The only solution to the aliasing problem is to ensure that the sampling rate is higher than twice of the highest frequency present in the signal. If that is not possible, then use an anti-aliasing filter to screen out those frequencies higher than ½ of the sampling frequency before sampling, assuming the removed frequencies are not of importance.

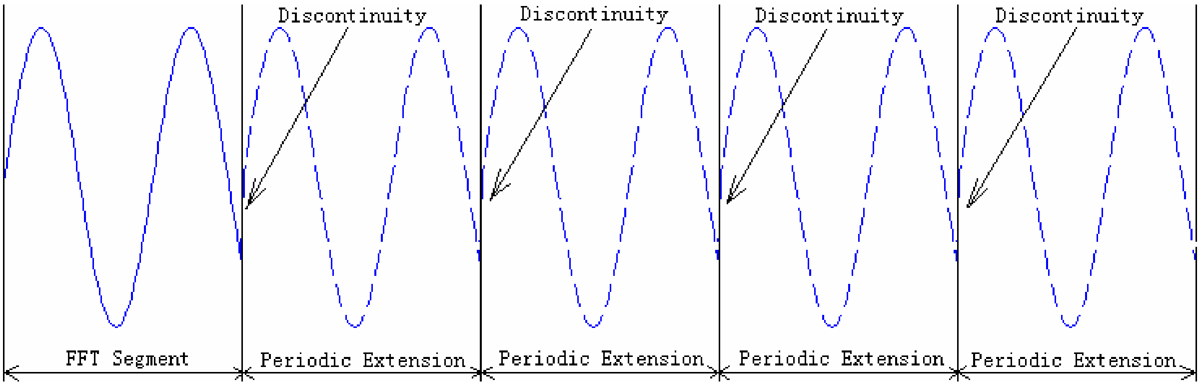
3. Leakage

Spectral leakage is the result of the inherent assumption in the DFT/FFT algorithm that the time record in a DFT/FFT segment is exactly repeated throughout all time and that signals contained in a DFT/FFT segment are thus periodic at intervals that correspond to the length of the DFT/FFT segment. If the time record in a DFT/FFT segment has a non-integer number of cycles, this assumption is violated and spectral leakage occurs.



An example of a FFT segment containing an integer number of cycles

In the above figure, a FFT segment contains exactly an integer number of cycles of the original signal, thus after periodic extension, which is assumed in DFT/FFT, the resulted signal is exactly the same as the original one, and no distortion and spectrum leakage occurs.



An example of a FFT segment containing a non-integer number of cycles

In the above figure, a FFT segment contains a non-integer number of cycles of the original signal, thus after periodic extension, the resulted signal is quite different from the original one in that there are discontinuities between successive segments. These artificial discontinuities turn up as very high frequencies in the spectrum of the signal and are not present in the original signal. These frequencies could be much higher than the Nyquist frequency (i.e. $f_s/2$) and will be aliased somewhere between 0 and $f_s/2$. The obtained spectrum is thus smeared as if the energy at the frequencies present in the original signal leaks out into all the other frequencies.

The only way to completely avoid the spectral leakage problem is to set the sampling frequency and FFT size properly such that a FFT segment contains exactly an integer number of cycles of the signal under test. However, this is not possible in most of the practical applications. When the spectral leakage is inevitable, the best approach to minimize its effect is to multiply the time record by a suitable window function before performing FFT. A window function generally has a unit gain at the center of the FFT segment, decrease gradually to zero or a small value at both ends of the FFT segment, and becomes zero outside the FFT segment. Thus it greatly suppresses the discontinuity of the time record between FFT segments. The multiplication operation in the time domain would means a convolution of their respective spectra in the frequency domain. A FFT without a window function is actually a FFT with a rectangle window function, which has a unit gain inside the FFT segment and becomes zero outside. Different window functions have different characteristics and can be adopted in different situations. A comprehensive study of various window functions with the help of Multi-Instrument* can be found at:

http://www.virtins.com/doc/D1003/Evaluation_of_Various_Window_Functions_using_Multi-Instrument_D1003.pdf

** Multi-Instrument is a powerful multi-function virtual instrument software. It supports a variety of hardware ranging from sound cards which are available in almost all computers to proprietary ADC and DAC hardware such as NI DAQmx cards, VT DSO and so on. The software can be downloaded at: www.virtins.com/MIsetup.exe and www.multi-instrument.com/MIsetup.exe for 21-day fully functional FREE trial.*

4. Picket Fence Effect

As indicated previously, FFT can only produce spectral analysis results at discrete frequencies: 0, $\Delta f \times 1$, $\Delta f \times 2$, …, $\Delta f \times N/2$, where N is the FFT size and $\Delta f = f_s/N$. But the real spectrum of the signal can be continuous or discrete but different from those discrete frequencies calculated by FFT. Observation of the spectrum with FFT is analogous to looking at it through a sort of "picket-fence". If a frequency peak of the signal falls exactly to a FFT spectral line, then it will be observed with correct magnitude, otherwise, its energy will be shared by the its adjacent FFT spectral lines, which in effect smears the peak or even causes it undetectable.

One way for reducing the picket-fence effect is to changing the number of points (N) in a time record by adding zeros at the end of the original record, while keeping the original record intact. This will change locations of the FFT spectral line and reduce the interval between these lines. In this manner, spectral components originally hidden from FFT view can be shifted to points where they can be observed.

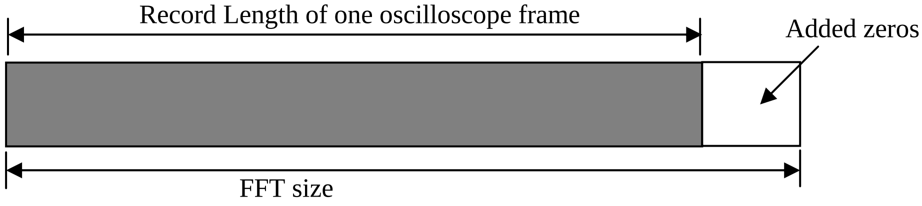
5. Zero Padding

FFT generally requires the number of data points N to be an integer power of two, e.g. 1024, 2048, 4096, 8192, etc.. If not, zero can be added at the end of the original record. This is called zero padding.

Although zero padding increases the apparent FFT frequency resolution, as dictated by $\Delta f = [f_s]/N$, the real frequency resolution is still determined by: $[f_s]$/[Number of Real Data Points in a FFT segment], that is 1/T, where T is the sampling duration. This is because zero padding increases the number of spectral lines simply by interpolation. The sampling duration is important because it determines the lowest frequency that FFT can measure. For example, if the frequency component of 1 Hz in the signal is of interest, then the sampling duration must be greater than 1 s.

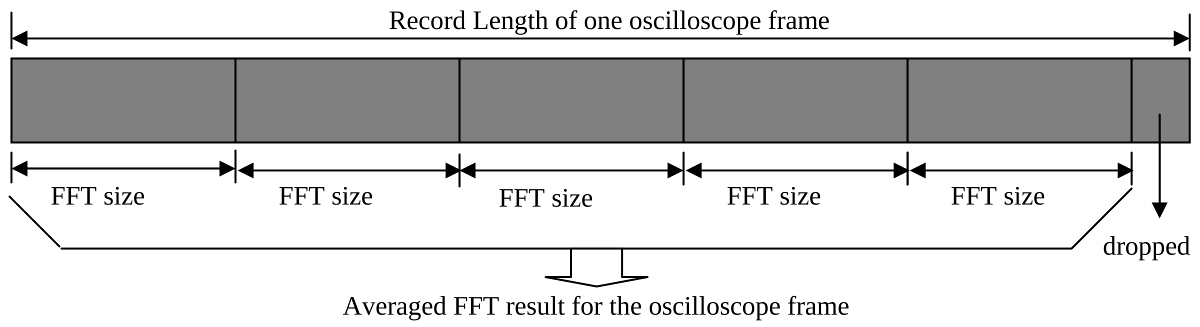
6. Record length, FFT Size, Window Overlap

Unlike many other softwares, Record Length and FFT size can be selected independently in Multi-instrument. If the FFT size is greater than the Record Length, then zeros will be padded at the end of the actual measurement data during FFT computation.



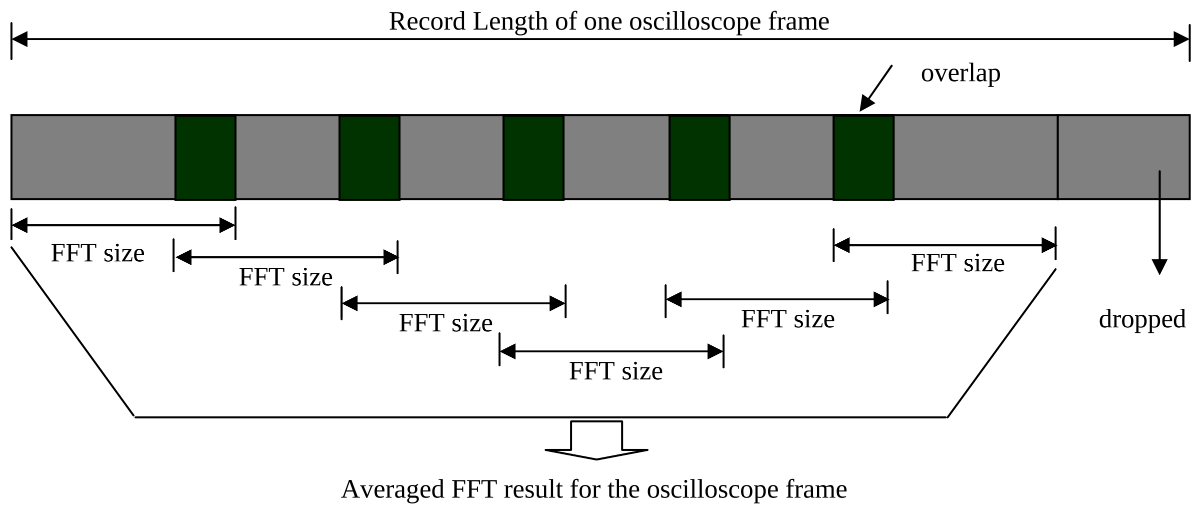
FFT size>Record Length

If the FFT size is less than the Record Length, then the measurement data will be split into different FFT segments with the size of each segment equal to the FFT size. The last segment of data will be dropped if its size is not equal to the FFT size (see figure below). The final result will be obtained by averaging the FFT results from all segments. The average method here is referred to as Intra-Frame Average, as it is done within one oscilloscope frame. This is in contrast to the Inter-Frame Average set through [Setting]>[Spectrum Analyzer Processing]>[Inter-Frame Processing] in Multi-Instrument, which is performed among successive oscilloscope frames.



FFT Size<Record Length, no overlap of FFT segments

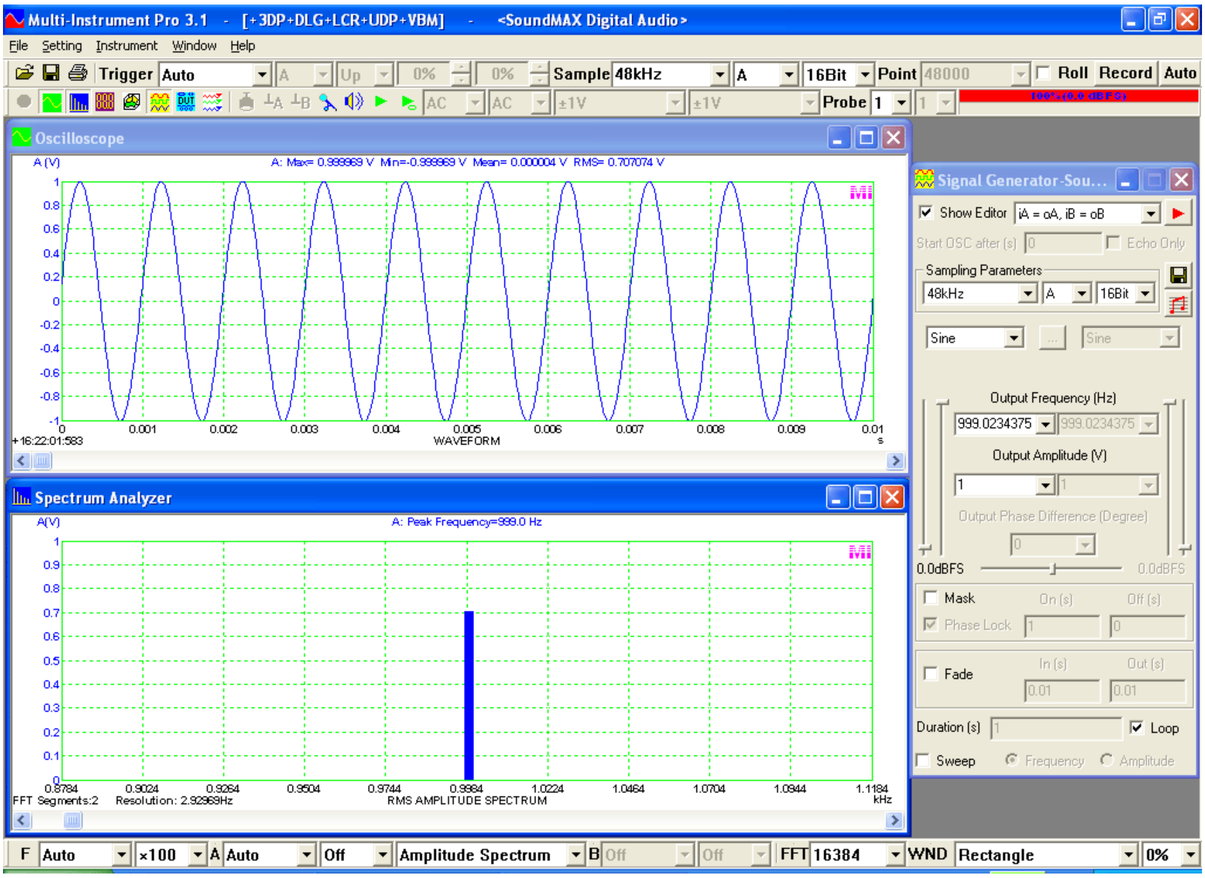
The use of window function suppresses greatly the original data at the edges of the window. As a result, these data contribute much less to the analysis result than the data at the center of the window. To make full use of the acquired data, FFT segments can be overlapped when the Record Length is greater than the FFT size.



FFT Size<Record Length, FFT segments overlapped

7. Case Study of Aliasing, Leakage, Picket Fence Effect

7.1 A FFT segment containing an integer number of cycles



Parameters:

In the above figure, fs = 48000 Hz, FFT size = 16384, Record Length = 48000, window = Rectangle, Overlap = 0%. The signal is sinusoidal with f = 999.0234575 Hz, Amplitude = 1 V. It is generated using the signal generator function under the "iA=oA, iB=oB" mode. The Y scale of the spectrum analyzer is set to Vrms.

Results and Analysis:

1) No Aliasing

The signal frequency is well below fs/2, thus no aliasing occurs.

1) No Leakage

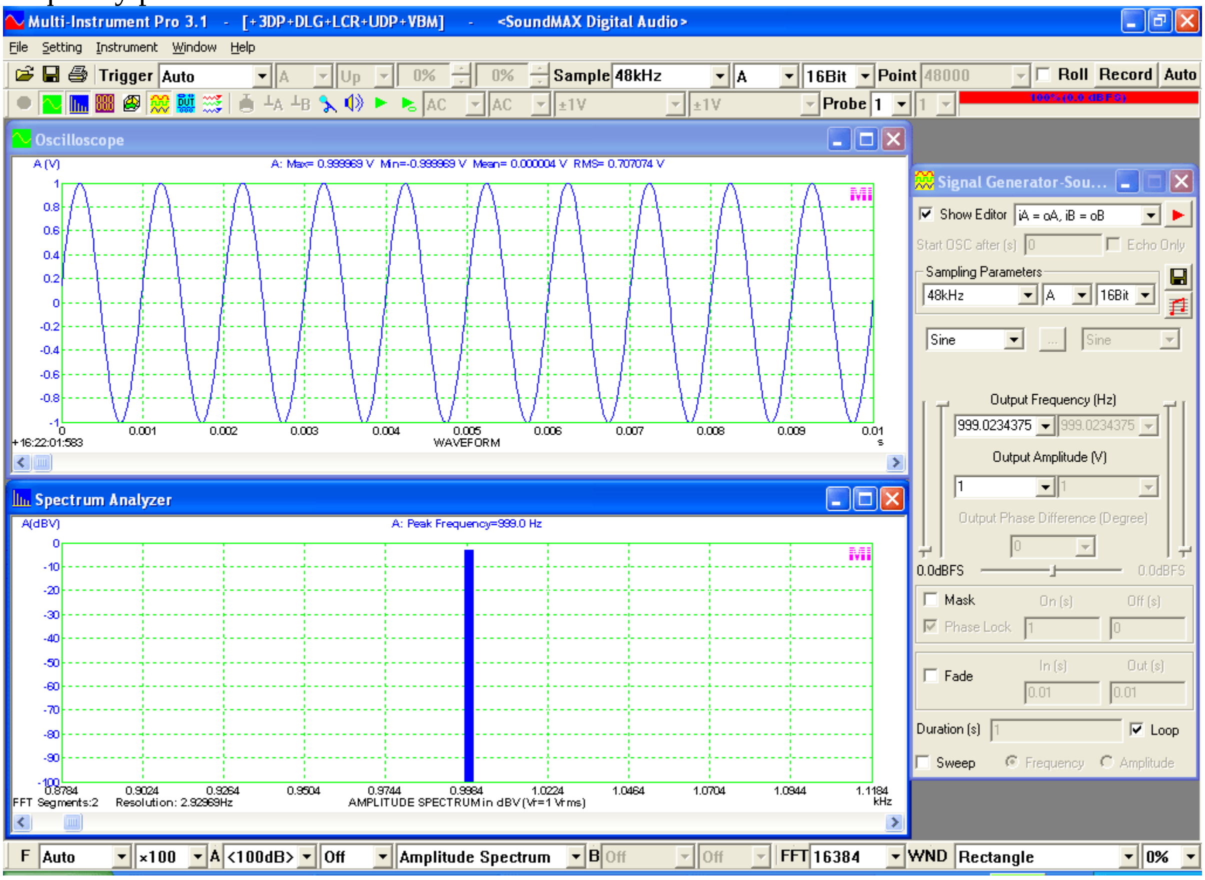
One FFT segment contains [FFT size] × f/fs = 16384 × 999.0234575 / 48000 = 341 cycles. As this is an integer number, no spectral leakage occurs.

2) No Picket Fence Effect

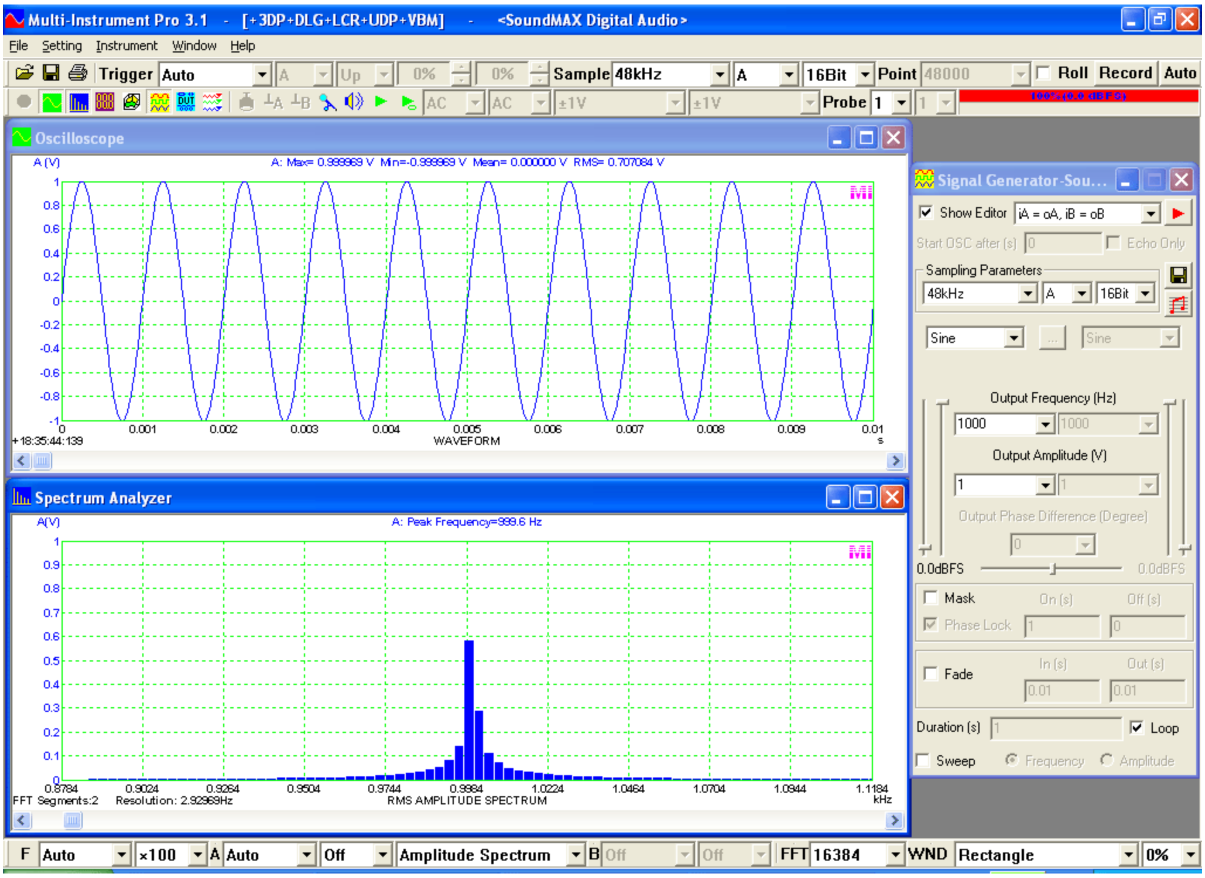
The 342nd spectra line f_{341} is located at $\Delta f \times 341$ = fs/[FFT size] × 341 = 999.0234575 Hz, which is the exactly same as the signal frequency. This means no picket fence effect occurs.

As a result, the single frequency peak in the spectrum analyzer contains all energy of the original signal, and its magnitude is 0.707 Vrms if measured using the cursor reader provided in Multi-Instrument. It is equal to the RMS value of the signal. In other words, the energy in the time domain is conserved in the frequency domain. This is often called Parseval's theorem.

The following figure is the same as the above one except that the Y scale of the spectrum analyzer is set to dBV in an attempt to make those small-magnitude spectral lines prominent. It shows that there are no visible frequency components above –100dBV except the frequency peak itself.



7.2 A FFT segment containing a non-integer number of cycles with no window function



Parameters:

In the above figure, fs = 48000Hz, FFT size = 16384, Record Length = 48000, window = Rectangle, Overlap = 0%. The signal is sinusoidal with f = 1000 Hz, Amplitude = 1 V. It is generated using the signal generator function under the "iA=oA, iB=oB" mode. The Y scale of the spectrum analyzer is set to Vrms.

Results and Analysis:

1) No Aliasing

The signal frequency is well below fs/2, thus no aliasing occurs.

2) Leakage

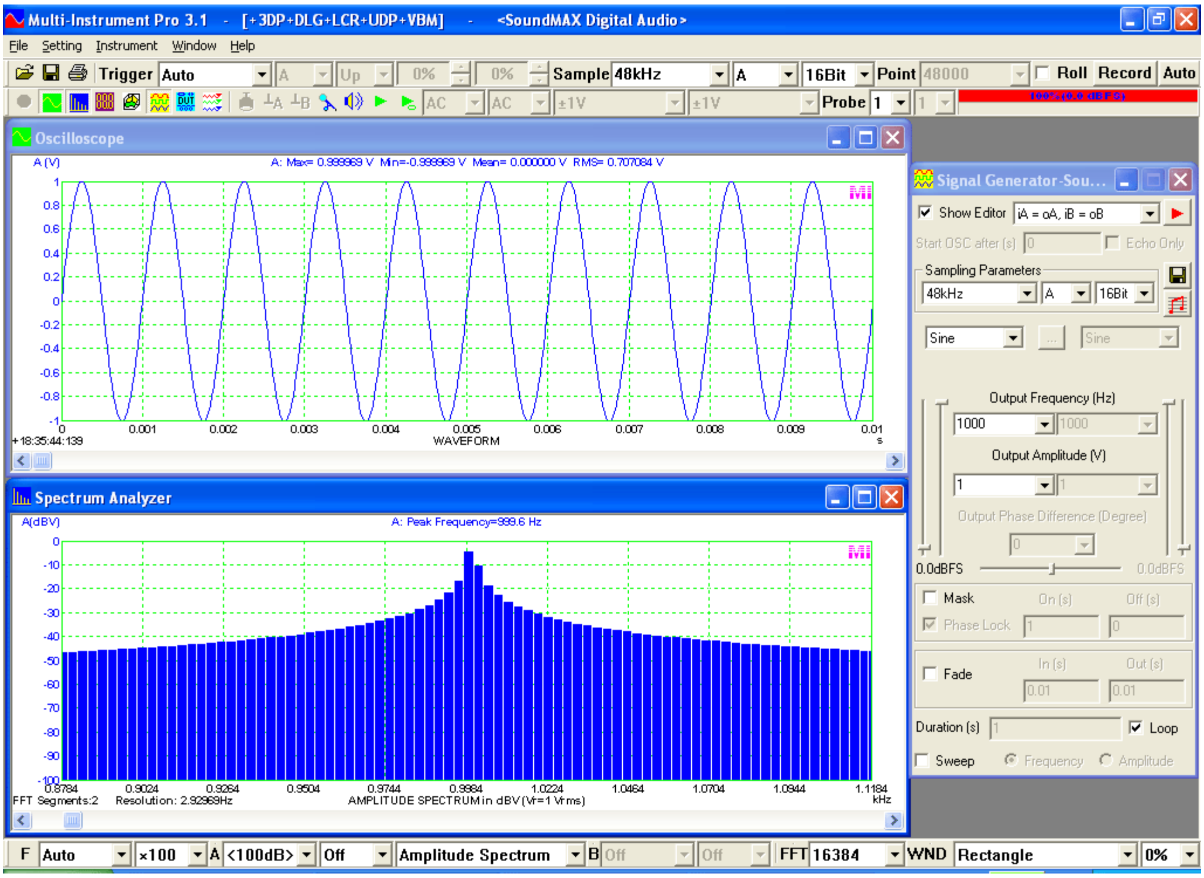
One FFT segment contains [FFT size] × f/fs = 16384 × 1000 / 48000 = 341.3333 cycles. As this is a non-integer number, spectral leakage occurs.

3) Picket Fence Effect

The signal frequency is located between the 342^{nd} spectra line f_{341} (located at $\Delta f \times 341$ = fs/[FFT size] × 341 = 999.0234575 Hz) and the 343^{rd} spectra line f_{342} (located at $\Delta f \times 342$ = fs/[FFT size] × 342 = 1001.953125 Hz), and thus picket fence effect takes place.

As a result, the highest spectral line in the spectrum analyzer is f_{341}, because it is the closest to the original signal frequency. Its magnitude is 0.585 Vrms if measured using the cursor reader provided in Multi-Instrument. Thus it contains only partial energy of the signal. The second highest spectral line is f_{342} with a magnitude of 0.292 Vrms. These two lines account for $(0.585^2 \times 0.292^2)/0.707^2 = 85.5\%$ of the total energy of the signal. The rest of energy spreads to all the rest of spectral lines. One thing to note is that the frequency peak displayed as text at the top of the spectrum analyzer is 999.6 Hz rather than 999.0234575 Hz where the highest spectral line is located. The displayed value is closer to the signal frequency 1000 Hz. The better accuracy is achieved via the unique sub-FFT-bin-size peak frequency detection algorithm used in Multi-Instrument. It will be shown later that this algorithm works better when a window function is used.

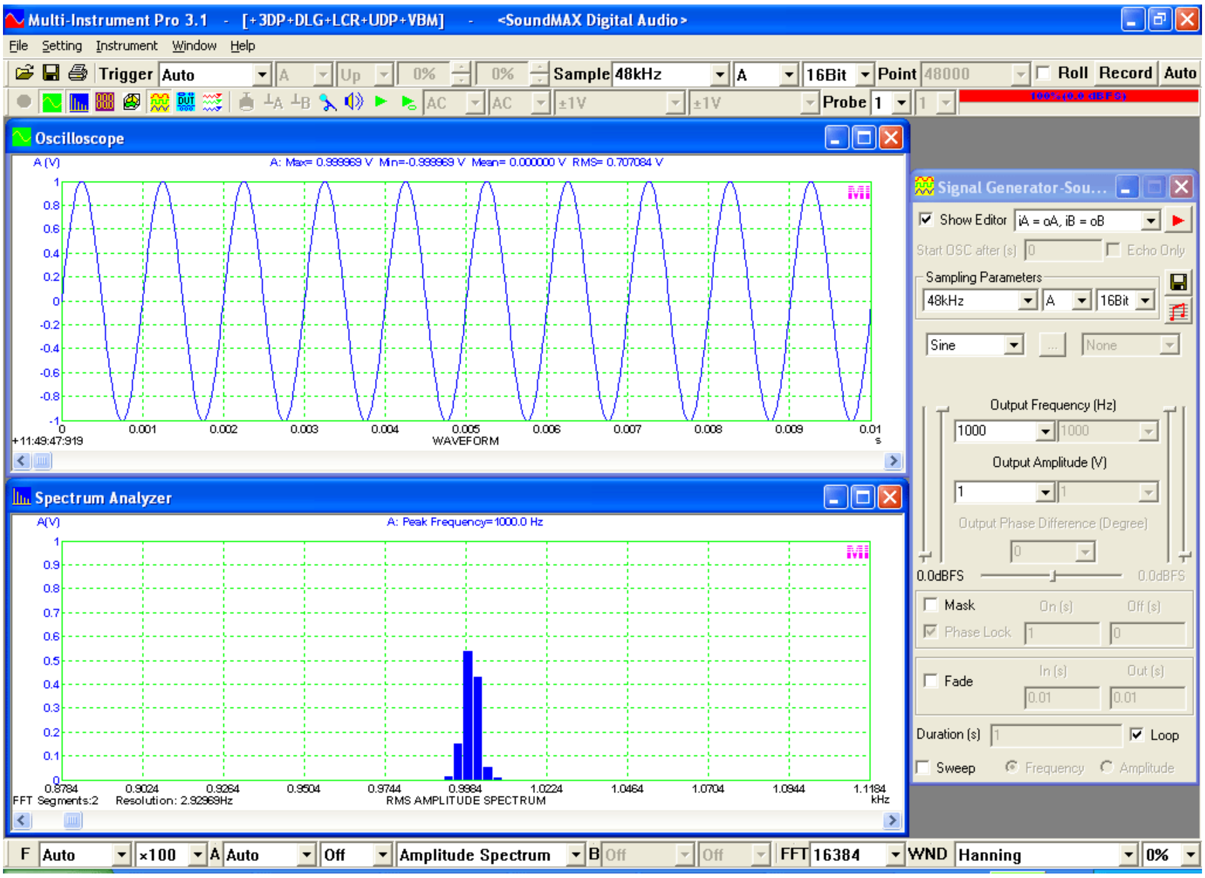
The following figure is the same as the above one except that the Y scale of the spectrum analyzer is set to dBV in an attempt to make those small-magnitude spectral lines prominent. It shows that all the spectral lines are above –100dBV, an indication of severe spectral leakage.



7.3 A FFT segment containing a non-integer number of cycles with a Hanning window

Parameters:

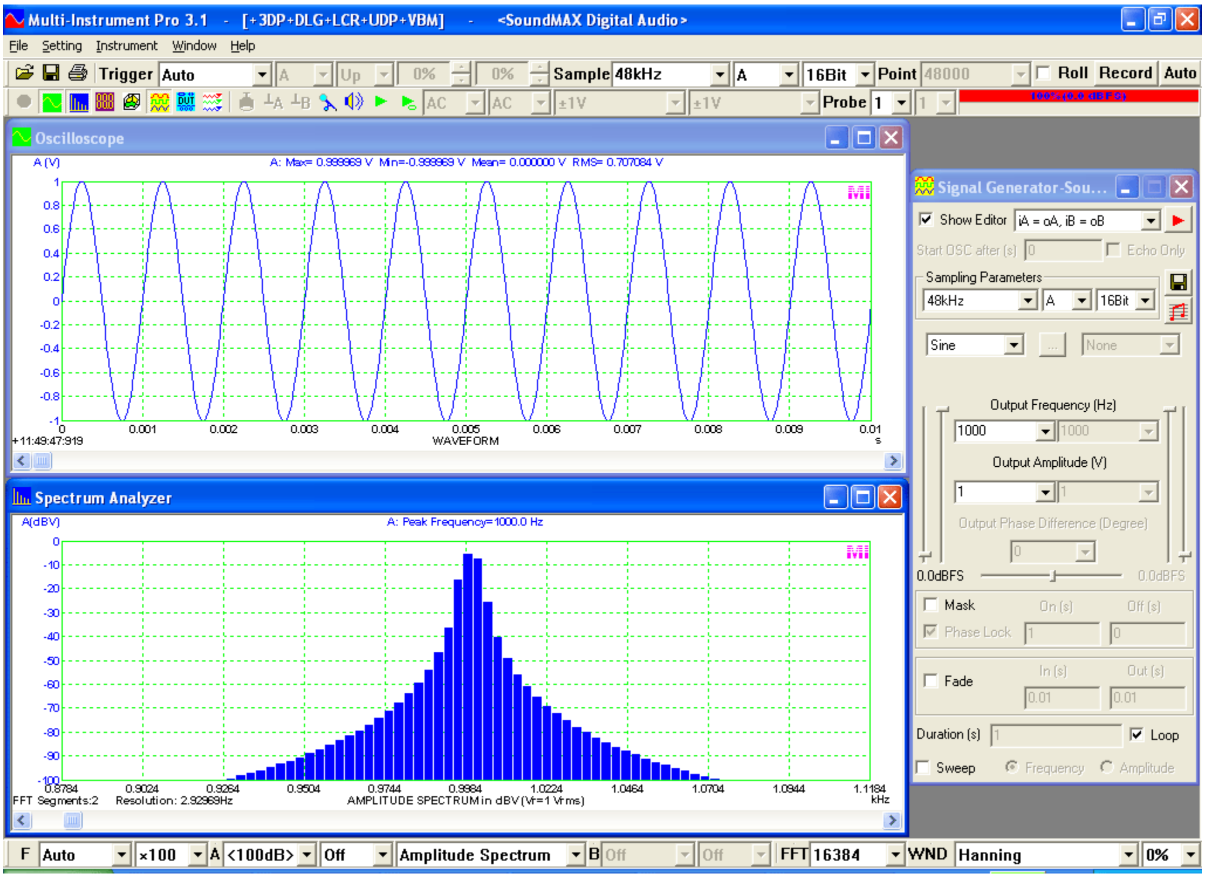
All parameters are the same as the previous section except that a Hanning window is used in the spectrum analysis.



Results and Analysis:

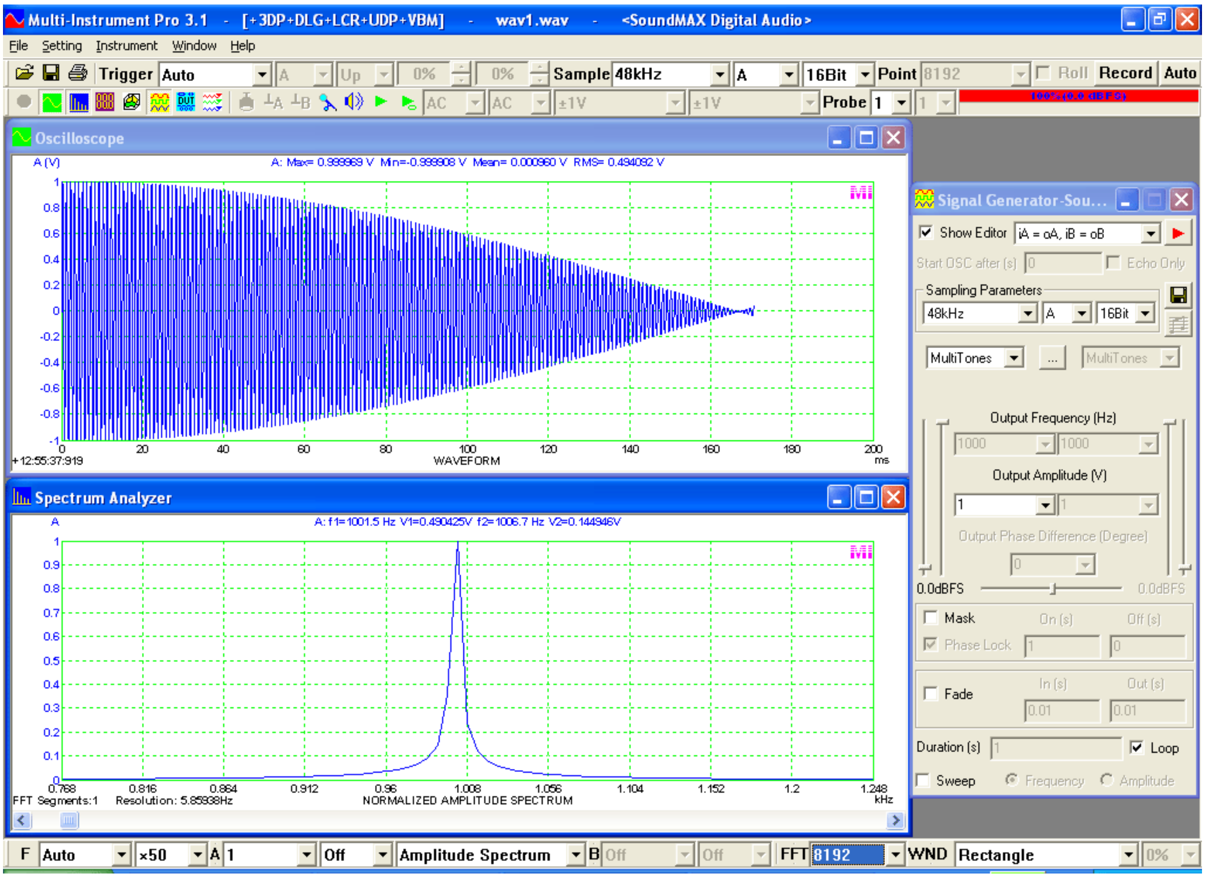
The Y scale of the spectrum analyzer in the above figure is set to Vrms. Compared with the first figure in the previous section, the energy of the spectrum converges more to the few spectral lines around the frequency peak, although the magnitude of the frequency peak itself drop a bit. The spreading of the signal energy to those spectral lines away from the frequency peak is much reduced. Same conclusion can be drawn from the comparison between the following figure (The Y scale of the spectrum analyzer is set to dBV) and the second figure in the previous section.

One thing to note is that the frequency peak displayed as text at the top of the spectrum analyzer is 1000.0 Hz rather than 999.0234575 Hz where the highest spectral line is located. The displayed value is precisely the same as the signal frequency 1000 Hz. The better accuracy is achieved via the unique sub-FFT-bin-size peak frequency detection algorithm used in Multi-Instrument. Compared with the result of the previous section, it can be seen that the sub-FFT-bin-size peak frequency detection algorithm works better when a window function is used.


Multi-Instrument Pro 3.1 - [+3DP+DLG+LCR+UDP+VBM] - <SoundMAX Digital Audio>
Oscilloscope
A: Max= 0.999969 V Min=-0.999969 V Mean= 0.000000 V RMS= 0.707084 V
WAVEFORM
Spectrum Analyzer
A: Peak Frequency=1000.0 Hz
AMPLITUDE SPECTRUM in dBV (Vr=1 Vrms)
FFT Segments:2 Resolution: 2.92969Hz
Signal Generator-Sou...
Output Frequency (Hz)
Output Amplitude (V)
Amplitude Spectrum
FFT 16384
WND Hanning

8. Case Study of Zero Padding and Frequency Resolution

8.1 No Zero Padding



Parameters:

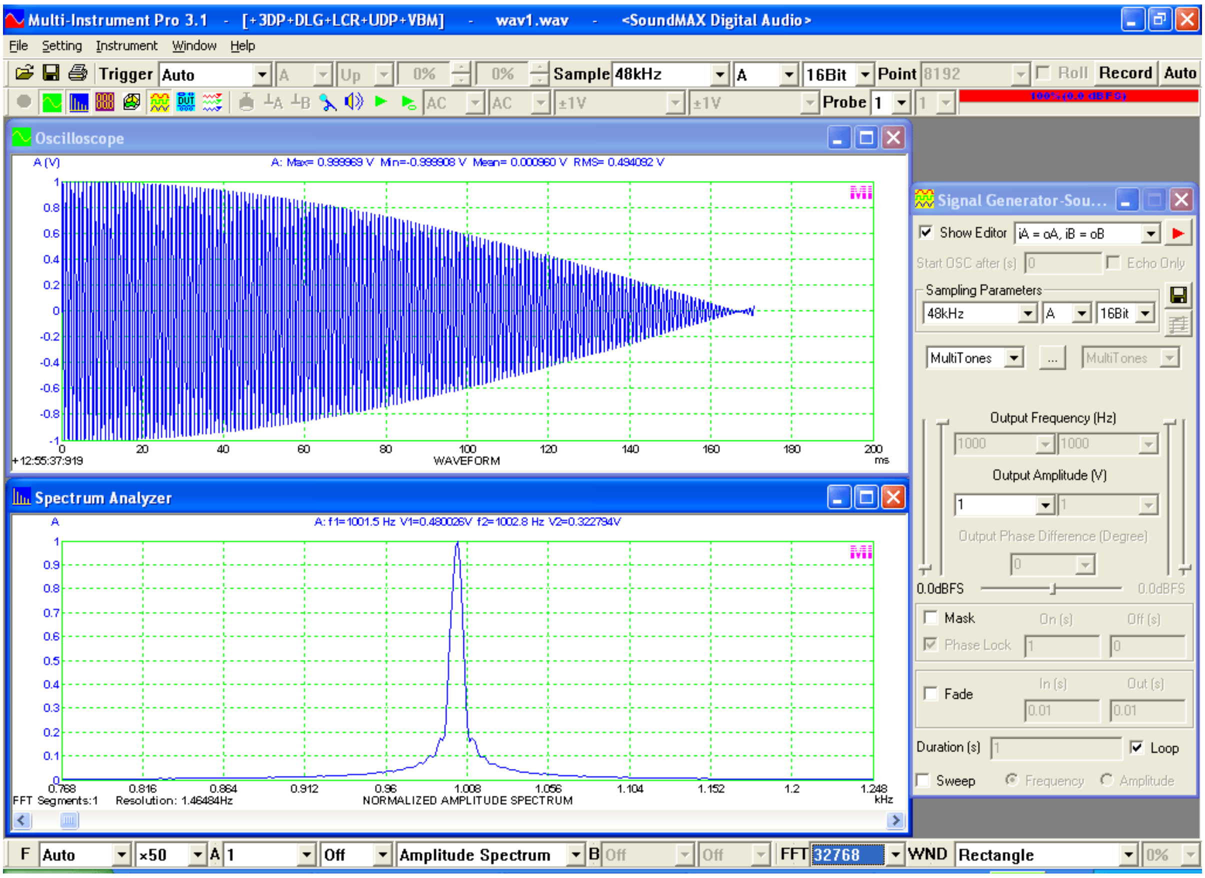
In the above figure, fs = 48000Hz, FFT size = 8192, Record Length = 8192, window = Rectangle, Overlap = 0%. The signal is a sum of two sinusoidal waves: f1 = 1000 Hz, Amplitude1 = 1 V; f2 = 1003 Hz, Amplitude2 = 1 V. It is generated using the MultiTone function of the signal generator under the “iA=oA, iB=oB” mode.

Results and Analysis:

1) No Zero Padding
As [FFT Size] = [Record Length], no zero padding.

2) Frequency Resolution
The FFT frequency resolution is $\Delta f = f_s/[\text{FFT Size}] = 48000/8192 = 5.859$ Hz. Thus it cannot resolve the two frequency peaks which are only 3 Hz apart as shown in the above figure.

8.2 Zero Padding, Frequency Resolution not Increased



Parameters:

All parameters are the same as the previous section except that the FFT size is changed to 32768 while keeping the Record Length = 8192 unchanged.

Results and Analysis:

1) Zero Padding
As [FFT Size]>[Record Length], zero padding is used.

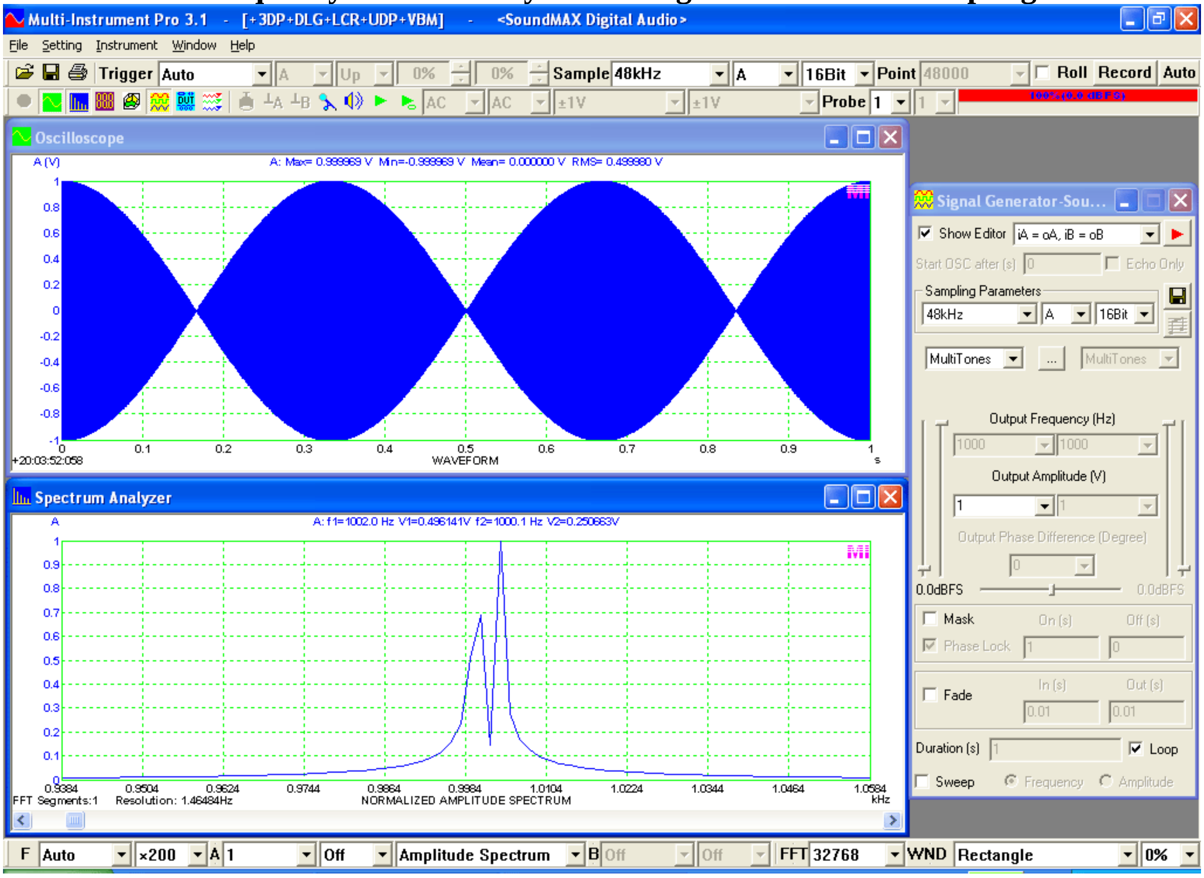
2) Frequency Resolution
The apparent FFT frequency resolution is $\Delta f = f_s$/[FFT Size] = 48000/32768 = 1.46484 Hz. Although Δf seems to be small enough to resolve the two frequency peaks which are 3 Hz apart, it does not, as shown in the above figure. This is because the real frequency resolution is: f_s/[Number of Real Data Points in a FFT segment] = 48000/8192 = 5.859 Hz, which remains unchanged.

It illustrates that zero padding does not improve the real frequency resolution although it does provide more spectral lines via interpolation.

8.3 Increase Frequency Resolution by Increaseing Sampling Duration

The real frequency resolution can be increased by increasing the sampling duration. There are two ways to increase the sampling duration, one is to increase the number of sampling points, the other is to decrease the sampling frequency.

8.3.1 Increase Frequency Resolution by Increasing the Number of Sampling Points



Parameters:

All parameters are the same as Section 8.1 except that: Record Length = 48000, FFT Size = 32768.

Results and Analysis:

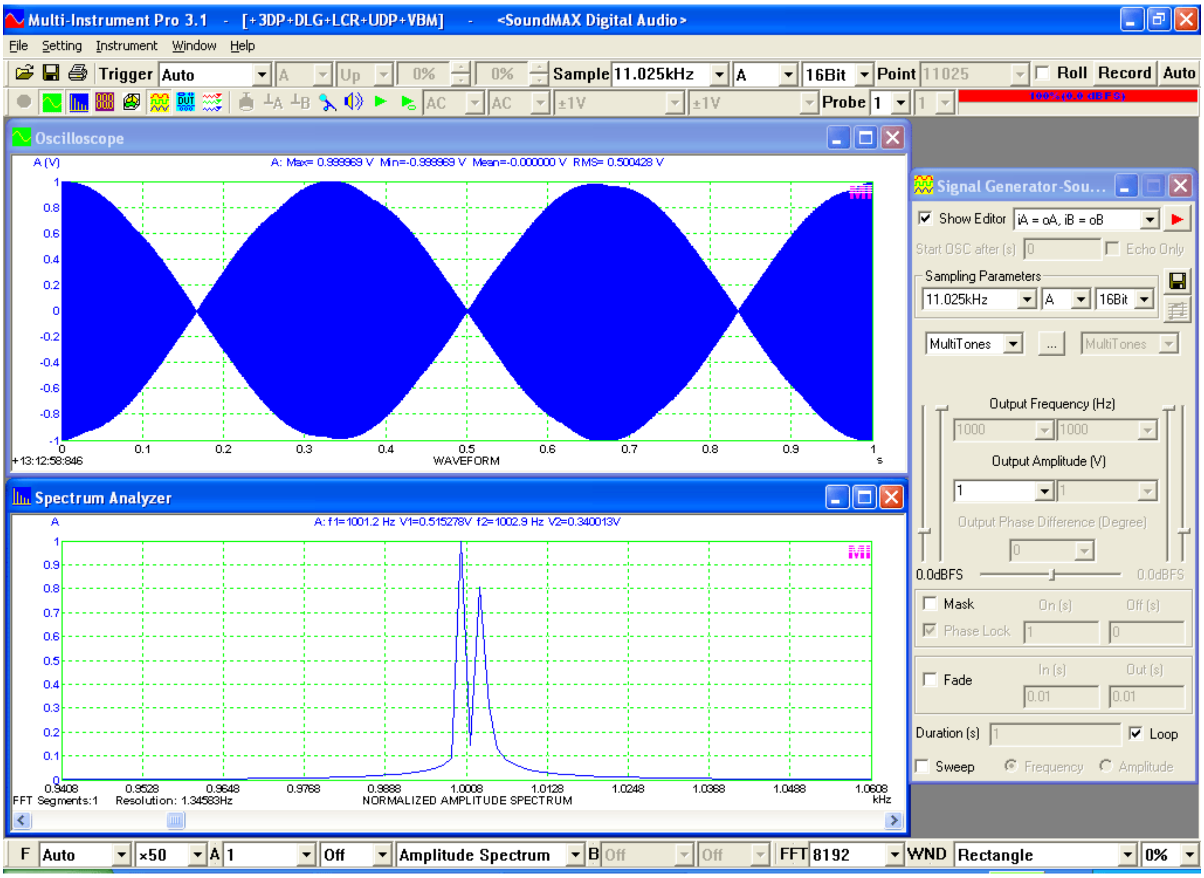
1) Zero Padding

As [FFT Size]<[Record Length], no zero padding is used. Note that the last 48000-32768 = 15232 pints is dropped and not used in the spectrum analysis.

2) Frequency Resolution

The FFT frequency resolution is $\Delta f = f_s/[\text{FFT Size}] = 48000/32768 = 1.46484$ Hz. Thus it is just enough to resolve the two frequency peaks which are 3 Hz apart, as shown in the above figure.

8.3.2 Increase Frequency Resolution by Decreasing Sampling Frequency



Parameters:

All parameters are the same as Section 8.1 except that: Record Length = 11025, Sampling Frequency = 11025 Hz. The FFT size remains unchanged at 8196.

Results and Analysis:

1) Zero Padding

As the [FFT Size]<[Record Length], no zero padding is used. Note that the last 11025-8192= 2833 pints is dropped and not used in the spectrum analysis.

2) Frequency Resolution

The FFT frequency resolution is $\Delta f = f_s/[\text{FFT Size}] = 11025/8192 = 1.3458$ Hz. Thus it is just enough to resolve the two frequency peaks which are 3 Hz apart, as shown in the above figure.

