MEscope Application Note 01

The FFT, Leakage, and Windowing

INTRODUCTION

The steps in this Application Note can be duplicated using any Package that includes the VES-3600 Advanced Signal Processing Option.

Click here to download the Demo MEscope Project file for this App Note.

MEscope makes it very convenient to look at signals in either the time or the frequency domain. You can transform you data between these two domains at will without losing any signal characteristics.

- **Transform | FFT** transforms a Data Block of time domain waveforms into their equivalent frequency domain spectra
- **Transform | Inverse FFT** transforms a Data Block of spectra into their equivalent time domain waveforms

Another command in MEscope lets you create Fourier spectra, Auto & Cross spectra, Power Spectral Densities (PSDs), or Energy Spectral Densities (ESDs).

- **Transform | Spectra** creates a new Data Block of spectra from time domain waveforms or other spectra

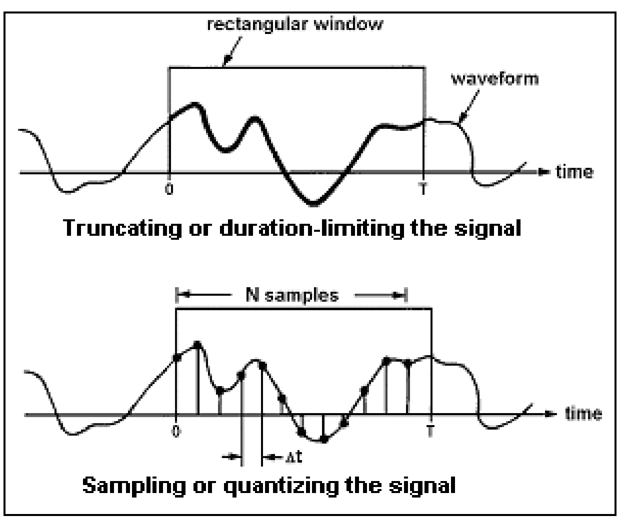
Additionally, MEscope has a command with which you can synthesize a broad variety of sine, random, chirp, or impact signals, plus Auto spectra.

- **File | New | Data Block** synthesizes time waveforms or Auto spectra, and puts them into a new Data Block window

You will use these commands to review some basic properties of the Fast Fourier Transform (FFT) and learn how common FFT ***window leakage errors*** can be minimized.

BASIC TIME & FREQUENCY RELATIONSHIPS

The FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a digital time domain signal, or time-history. Three basic formulas govern all FFT calculations.



Time Domain Windowing & Sampling.

1. Time Domain Equation

Digital signal processing with an FFT assumes that a digitized time domain signal is represented by ***N uniformly spaced samples*** of data, which satisfy the equation,

T = **NΔt** (1)

T = total time of the sampled signal (in seconds)

Δt = time step (or increment) between samples of data

N = number of samples, also called the ***Block Size***

The schematic figure above depicts the digital sampling of a continuous (analog) time signal. This consists of two steps:

1. Multiply the signal by a rectangular observation window (also called a ***boxcar*** or ***uniform*** window) to limit its duration to a practical observation time. This is termed ***time truncation***.
2. Measure the amplitude of the truncated signal at a series of equally spaced discrete times.

Note that the total observation time (**T**) of the sampled signal includes the time ***up to but not including*** sample **N+1**, which is the first sample point of a subsequent sampling window, or block of data.

2. Frequency Domain Equation

Secondly, the FFT assumes that the DFT of the signal is represented by its ***mean value*** (**DC**) plus **N/2** ***uniformly spaced complex-valued*** samples (magnitudes & phases), which satisfy the equation:

$\mathbf{F_{max}} = \mathbf{\Delta f}$ (**N/2**) (2)

$\mathbf{F_{max}}$ = maximum frequency of the signal (in Hz), also called the *Nyquist frequency*

Δf = frequency step (or increment) between samples (or Lines) of data

N/2 = number of frequency domain samples, always *one half* of the *Block Size*

The resulting spectrum has a total of (**N/2 + 1)** samples in it; a **DC** value plus **N/2** frequencies. Furthermore, the **DC** value and the value at the Nyquist frequency ($\mathbf{F_{max}}$) contain *no phase* information. The samples (or **Lines)** at DC and $\mathbf{F_{max}}$ are *real valued*. All other **Lines** are *complex-valued*.

3. Nyquist Sampling

Thirdly, the maximum frequency of the spectrum ($\mathbf{F_{max}}$,) is related to the time domain sampling rate ($\mathbf{f_s}$), specifically:

$\mathbf{F_{max}} = \mathbf{f_s/2}$ (3)

$\mathbf{f_s} = \mathbf{1/\Delta t}$ **=** the *sample rate* (samples/second) at which the time domain data is sampled.

Equation (3) is a statement of **Shannon's sampling theorem**, namely: ***"The maximum frequency of a spectrum is equal to one half the sampling rate of its corresponding time domain signal."***

Solving the frequency domain equation for **Δf**, and substituting for $\mathbf{2F_{max}}$ gives a fourth equation:

$\mathbf{\Delta f = 2F_{max}/N = 1/N\Delta t = 1/T}$ (4)

Equation (4) tells us that the frequency resolution (**Δf**) of a digital spectrum is equal to the reciprocal of the length of the observation time (**T)**. In other words, **"*to obtain finer frequency resolution (smaller Δf), a signal must be sampled over a longer time period (T), not at a higher sampling rate (f_s).*"**

SYNTHESIS OF A PERIODIC SIGNAL

Let's start by synthesizing a periodic signal and analyzing its characteristics. The Fourier series of a *square-wave of amplitude* ***± 1*** and frequency ($\mathbf{f_0}$) is given by:

$$y(t) = \frac{4}{\pi}\sum_{k=0}^{K}\frac{1}{(2k+1)}Sin\left[2\pi(2k+1)f_0 t\right] \qquad (5)$$

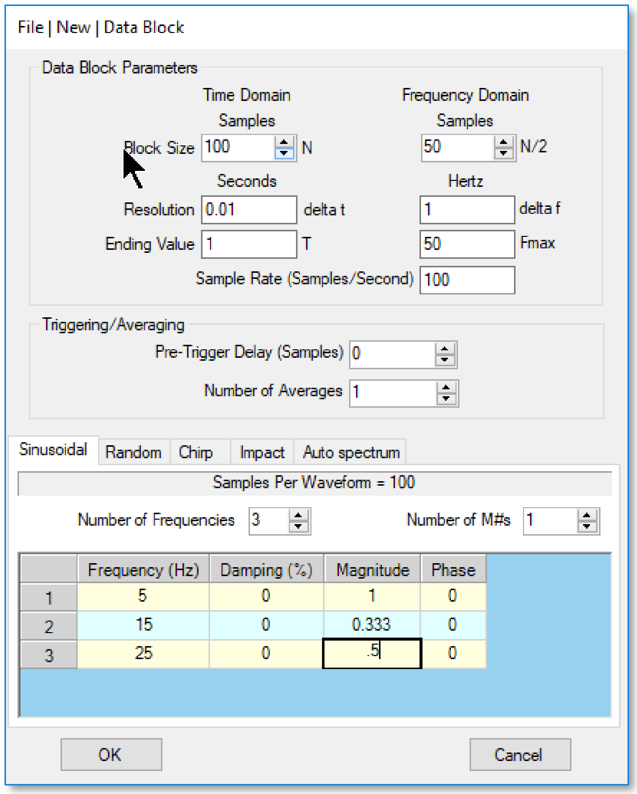
The first three terms of this series are:

$$y(t) = \mathrm{Sin}(2\pi f_0 t) + \frac{1}{3}\mathrm{Sin}(6\pi f_0 t) + \frac{1}{5}\mathrm{Sin}(10\pi f_0 t) \qquad (6)$$

These terms approximate a square-wave of amplitude (**± π/4 = 0.785)** with a fundamental frequency (**f_0**), let's say of **5** Hz.

Execute **File | New | Data Block**, and *click* on **OK** in the dialog box that opens.

The following dialog box will open.



Synthesis Dialog Entries for a 5 Hz Square-Wave.

- In the Data Block Parameters section, enter the following entries:

  Block Size (N) = 100 Samples

  Ending Value (F_{max}) = 50 Hertz

Note that entry of ***any two*** of the **Data Block Parameters** determines the ***remaining four*** parameters.

The remaining four parameters are:

T = 1 Second

ΔT = 0.01 Seconds

N/2 = 50 Samples

Δf = 1.00 Hertz

- ***Verify*** the following default values:

  Pre-Trigger Delay = 0 Samples

  Number of Averages = 1

- *Click* on the **Sinusoidal** tab to display it, and enter the following parameters:

  Number of Frequencies = 3

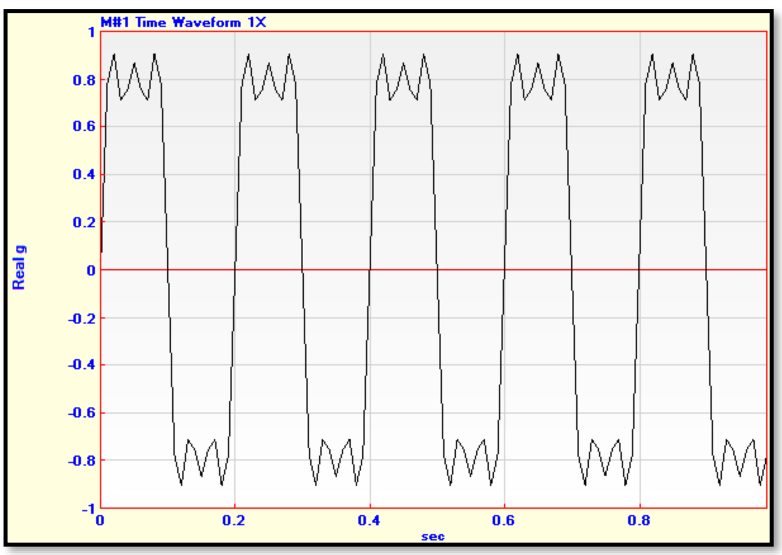
  Number of M#s = 1

  Frequency (Hz) values of **5**, **15** & **25** Hz

  Magnitude values of 1, $^1/_3$ & $^1/_5$

- ***Verify*** that all **Phase** and **Damping** values remain at the default **0** value.
- ***Press*** the **OK** button at the bottom of the dialog box

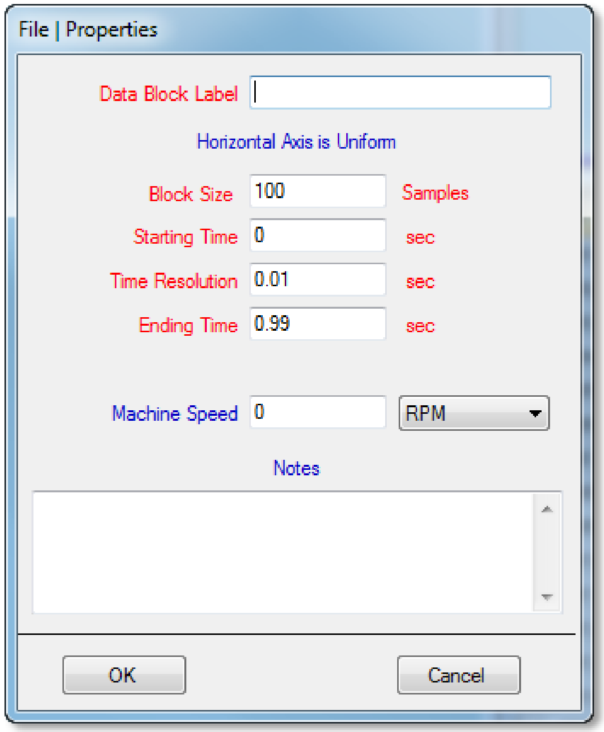
A new Data Block (**BLK**) Window will open, displaying the 5 Hz square-wave approximated by three sine waves. Note that the amplitude (in the center of the *ringing*) is approximately **±0.795 g,** as anticipated. Notice that *exactly* five cycles of this periodic waveform are captured in the **1** second duration of the window.



Synthesized 5 Hz Square-Wave.

To verify the parameters of this synthesized time waveform:

Execute **File | Properties** to open the Data Block Properties dialog box



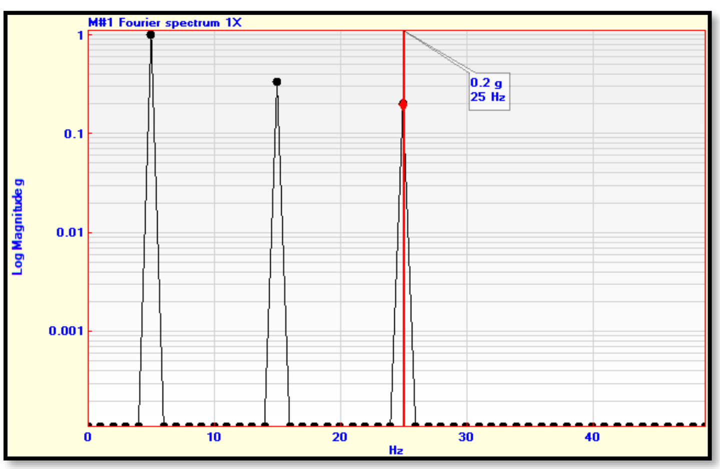
Data Block Properties of Synthesized 5 Hz Square-Wave.

Verify that the properties agree with the **N**, **T**, **ΔT** and **Starting Time** that you entered to synthesize the square-wave.

- ***Press* OK** to close this dialog.

Now let's look at the frequency spectrum of the square-wave.

Execute Transform | FFT



Spectrum of Synthesized 5 Hz Square-Wave.

The frequency spectrum of the signal is shown above.

- Execute **File | Properties** again to verify the number of frequency **Lines** (**N/2**) as well as **Δf** and the $\mathbf{F_{max}}$.

Note that peaks clearly appear at **5**, **15** and **25** Hz. Use the cursor to display the magnitude & frequency at each peak.

Execute Display | Cursor | Line Cursor

Execute Display | Cursor | Cursor Values

- ***Drag*** the cursor to each of the ***three peaks*** in the spectrum

Note that the spectrum contains the **1**, **1/3** and **1/5** magnitudes that were used to synthesize the square-wave.

Execute File | Save As

- Enter "**Periodic Square Wave"** and save the Data Block.

LEAKAGE, A FUNDAMENTAL FFT PROBLEM

In the preceding steps, we synthesized a square-wave using three sinusoidal components at **5**, **15** and **25** Hz. The square-wave was synthesized over a time period (**T)** of **1** second. Each of the square-wave components completed *exactly an integer number of cycles* (5, 15 & 25 respectively) during the **1**-second time period. As a result, each resulting peak in the spectrum fell *exactly* on an integer multiple of □**f** (**1.00 Hz**).

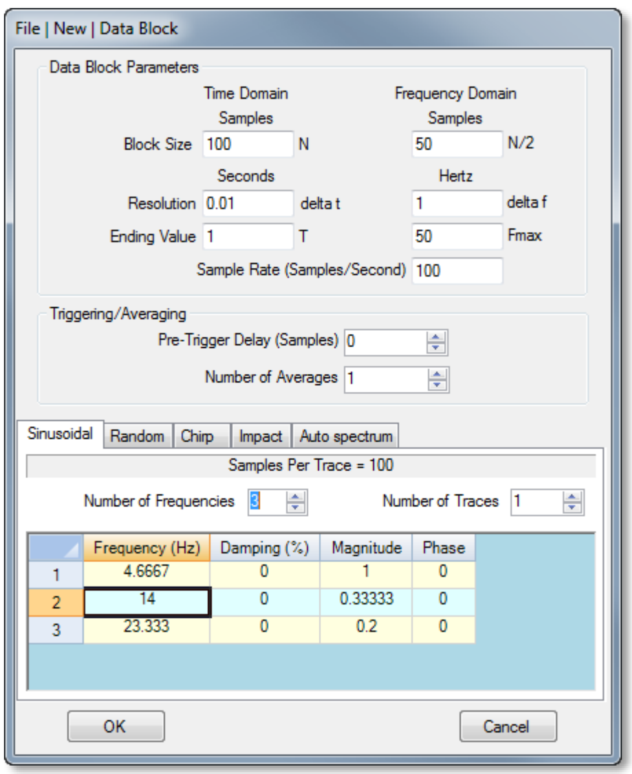
When **T** coincides *exactly* with N cycles of a periodic signal, the signal is said to be ***periodic in the sampling window***. Magnitudes and frequencies of the waveform components can be identified with high precision, as we have just seen.

SYNTHESIS OF A NON-PERIODIC SIGNAL

Let's repeat the square-wave synthesis, but with a small change. We will change the fundamental frequency ($\mathbf{f_0}$) from **5** Hz to **4** $^{2}/_{3}$ Hz, and leave all of the other parameters as they were before.

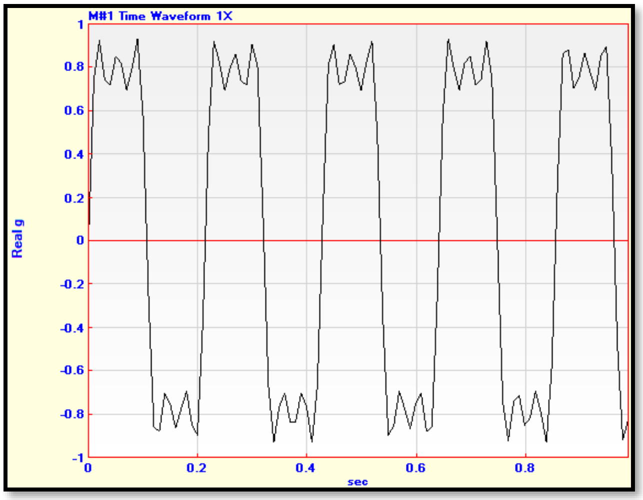
Execute **File | New | Data Block,** and make the following entries;

- Change the frequencies to **4.6667**, **14** & **23.333 Hz,** which are 1, 3 & 5 times 4$^{2}/_{3}$ Hz respectively.



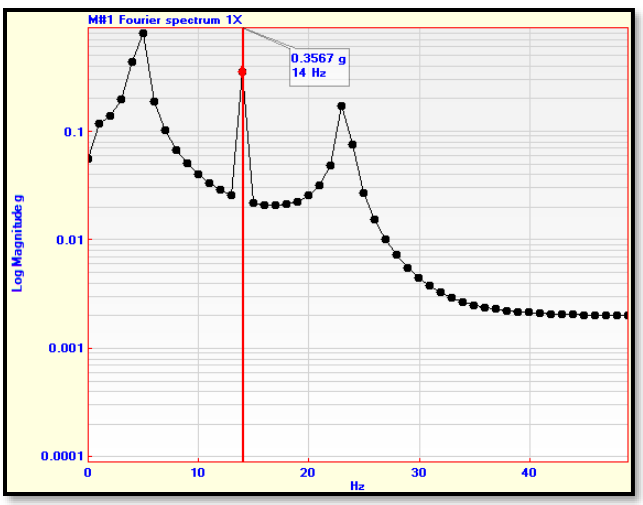
Synthesis Dialog Entries for a 4 $^{2}/_{3}$ Hz Square-Wave.

Notice that the *third* harmonic (**14** Hz) remains an *exact integer multiple* of □**f**, and that it will complete *exactly* 14 cycles within the **1** second sampling window. However, the *first* and *fifth* harmonics are *non-periodic* in the sampling window.



Synthesized $4\ ^{2}/_{3}$ Hz Square-Wave.

The $4\ ^{2}/_{3}$ Hz square-wave synthesis produces no surprises. The waveform exhibits the same shape and peak amplitude as the 5 Hz case. The difference is that **$4\ ^{2}/_{3}$** cycles are contained within the sampling window instead of **5** cycles. This seems like an insignificant difference!



Spectrum of Synthesized $4\ ^{2}/_{3}$ Hz Square-wave.

Execute Transform | FFT

Clearly, the resulting frequency spectrum (shown above) is *quite different* from the spectrum of the periodic signal. While three peaks may still be seen, the rest of the spectrum is not intuitively obvious. There is significant energy at *all frequencies,* even though only *three* discrete and well-separated tones were used to create the signal.

Note further that the *magnitudes* of the 1X and 5X peaks are *less* than the correct values of **1** & **0.2**. Only the **14** Hz *periodic* 3X peak has a magnitude, which is **close to the correct value** (**0.333**). This phenomenon is known as ***spectral leakage***.

WHAT CAUSES LEAKAGE?

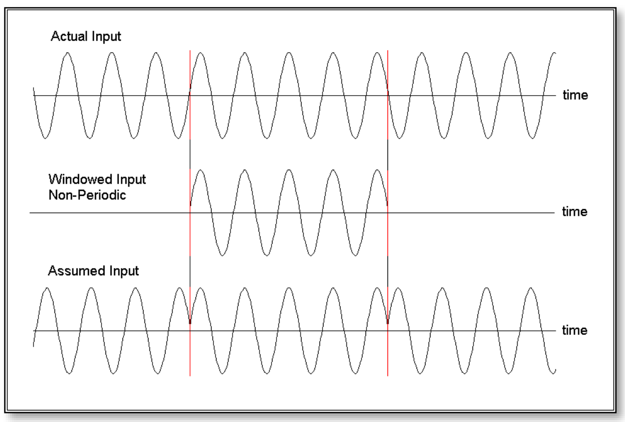
Whenever the FFT is applied to a signal that is *not exactly periodic in its sampling window*, leakage will occur in its spectrum. It is the result of multiplying two signals together in one domain, which is equivalent to *convolving* their Fourier transforms in the other domain. Convolution is a process of *shifting and adding* together two signals, so that the resultant signal is a *smeared version* of the expected result. (See reference [1] for details.)

In the case of a truncated sine wave, its correct spectrum was convolved with the spectrum of the rectangular sampling window. The resultant frequency spectrum was the convolution of the spectrum of the sine wave with the spectrum of the rectangular window, which *smeared* the peaks of expected square-wave spectrum.

Another Interpretation

Another way to understand leakage is to look at what happens to a sampled signal. One of the fundamental rules of the FFT is that ***"sampling of a signal in one domain causes repetition of it in the other domain".*** This phenomenon is illustrated below.

The actual signal is a continuous sine wave. The finite duration captured version is ***truncated*** in the sampling window. The FFT computes the spectrum of the ***assumed signal***, which when repeated outside of the sampling window, is no longer a continuous sine wave.



Repetition of a Non-Periodic Signal.

Instead of getting the FFT of the continuous sine wave (the expected result), we get the FFT of the assumed signal, which is a *smeared* spectrum.

From this example it should be also clear that "any type of signal that is completely contained within the sampling window (like a transient that damps out within the window), is also periodic in the window."

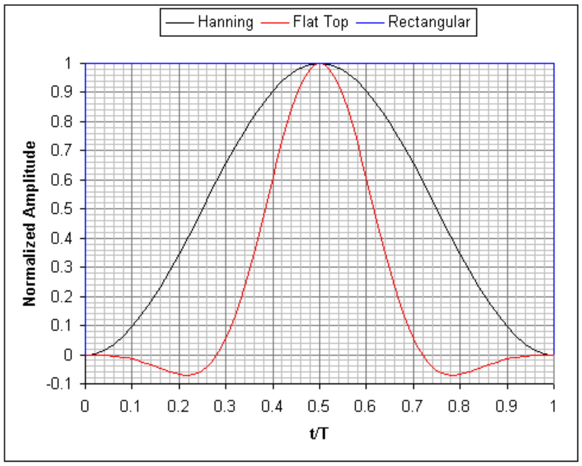
REDUCING LEAKAGE

If a periodic signal is *truncated* by its sampling window, leakage in its spectrum ***cannot be eliminated***. However, it can be ***significantly reduced*** so that the resulting spectrum contains a more accurate representation of the signal. This is accomplished by ***changing the shape*** of the sampling window, which is multiplied by the time waveform.

MEscope has three built-in time domain windows for windowing data; **Rectangular**, **Hanning**, and **Flat Top**. So far, you have used the Rectangular window, which is ***always used by default*** unless another window is specified.

Rectangular Window

The **Rectangular** window is the proper choice for analyzing transients that are ***completely contained*** within the duration **T** of the sampling window.



Window shapes: Rectangular, Hanning and Flat Top.

The Rectangular window is also the proper choice when analyzing periodic signals using ***order-normalized*** sampling, a process by which the sampling rate ($\mathbf{f_s}$) ***is changed*** in proportion to the operating speed of a rotating machine.

In virtually all other circumstances, the **Rectangular** window choice is inappropriate, and the resulting spectrum will contain leakage.

Hanning Window

The **Hanning** window is the best window choice when analyzing ***broad-band random*** signals, ***non-periodic cyclic*** signals, or a mixture of the two. It provides excellent suppression of signal truncation artifacts while retaining the ability to separate closely spaced tones or detect a tone buried in a noisy background.

Flat Top Window

The **Flat Top** window is specifically designed to give ***improved amplitude precision*** when measuring ***cyclic*** or ***narrow-band*** signals such as sine waves. It excels in this capacity, but is not as selective as the **Hanning** window for discerning closely spaced peaks in a spectrum.

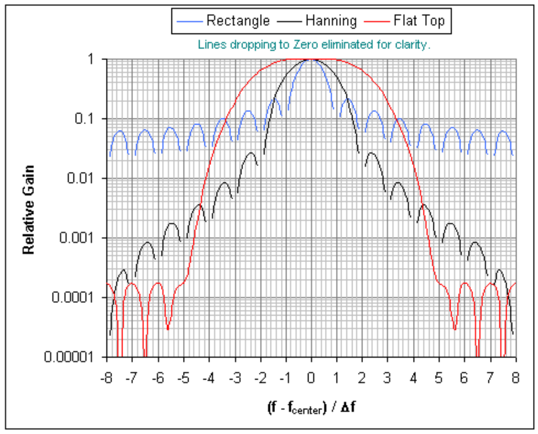
Both the **Hanning** and **Flat Top** windows are designed to optimize the spectrum analysis of *non-transient* signals. Both windows ***symmetrically taper*** the leading and trailing portions of the sampled time domain signal to ***essentially zero***. This has the effect of giving the time signal the *appearance* of being ***periodic*** in the time period **T**.

While the **Rectangular** window merely acts as a time gate (multiplying the signal by **1** or **0**), the **Hanning** or **Flat Top** window ***completely changes the time waveform shape***. Even though these windows distort the time waveform shape, their effect on the resulting spectrum is highly beneficial for reducing the effects of leakage.

FFT Digital Filters

Leakage and the FFT can be viewed from yet another perspective. The FFT numerically implements a bank of parallel digital filters. Each filter is of the ***same constant bandwidth*** (nominally **Δf**), and the filter center-frequencies are spaced **Δf** apart. In an ideal world, the transfer function of each filter would be a ***"brick wall"*** rectangle of width, **Δf**.

However, such ideal filters do not exist, either ***physically*** or ***mathematically***. Instead, the transfer function of each FFT filter has a complicated shape. The filter shapes for the **Rectangular**, **Hanning** and **Flat Top** windows are shown below.



Magnitude of FFT Filter Transfer Functions.

Note that the FFT filter for each of these windows is far broader than one **Δf**. Each one exhibits a maximum gain (sensitivity) at the ***center frequency*** (**f_{center}**). Adjacent filter center frequencies are spaced **Δf** apart, so it is clear that ***considerable overlap exists*** in a bank of filters.

The **Rectangular** filter has the narrowest center lobe, (a desirable attribute) spanning **2Δf's** between adjacent zeros. Outside of the center lobe, the **Rectangular** window exhibits *zero gain* at every multiple of **Δf.** It also has the highest side lobe amplitudes and the greatest curvature across the center lobe, compared to the other two window filters. These are detracting characteristics.

This window works well when applied to signals that are ***periodic*** in the sampling window. In this situation, each ***periodic*** signal component exists at ***exactly the center frequency*** of one of the window's filters. That signal component is coincident with the most sensitive frequency of the Rectangular filter. More importantly, the frequency of a ***periodic*** signal component ***corresponds to a zero gain point for all other filters***.

Alternatively, a ***truncated*** or ***non-periodic*** **waveform** has ***non-zero spectral components*** **at *all* frequencies** in the spectrum, resulting in ***leakage*** of the signal into all of the filter sideband frequencies. When a single frequency ***tone*** moves slightly by (±**Δf / 2** or less) off of the center frequency of a filter, it ***no longer coincides with the zeros of the adjacent filters***.

The **Hanning** window improves upon the Rectangular window because the ***height of its side lobes*** is much less in amplitude. This reduced side lobe gain ***minimizing the leakage*** of a signal into other frequencies. However, the Hanning window has a broader center lobe with **4 Δf**'s between its nearest bounding zeros. Thus, the effective frequency resolution in a spectrum is reduced when a Hanning window is used.

The **Flat Top** window has an extremely wide center lobe, spanning more than **8 Δf's**. However, as the name implies, the top of the center lobe is ***extremely flat*** (within ± 0.1 %), providing an accurate amplitude even when a tone varies as much as ±**Δf / 2** from the center **f_{center}** of the digital filter. The side lobes of the **Flat Top** filter are well suppressed, but they are all of nearly equal amplitude.

Some of the characteristics of these three windows are listed below. The ***noise bandwidth***, **Δ_n**, is a standard measure of a window's ***broadband*** random noise performance. It is the bandwidth of the broadband power passed by the filter when it is subjected to white (or bandwidth) noise.

WINDOW COMPARISON

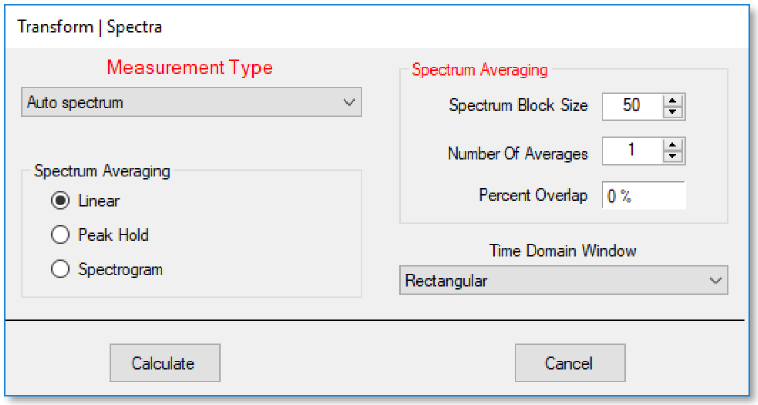
The three windows will be compared by applying them to the 4 $^2/_3$ Hz ***non-periodic*** square-wave. Start by transforming the Fourier spectrum of the synthesized 4 $^2/_3$ Hz square-wave back into the time domain.

IFT Execute Transform | Inverse FFT

Now we will create a new Data Block containing three different spectra, all from the same synthesized 4 $^2/_3$ Hz square-wave, but with each different time window applied to it.

- Execute **Transform | Spectra** in the **BLK: Non-Periodic Square Wave** window

- *Select* **Fourier spectrum** in the **Transform | Spectra** dialog box
- Select the **Rectangular** window in the dialog box that opens, and ***click*** on **Calculate**

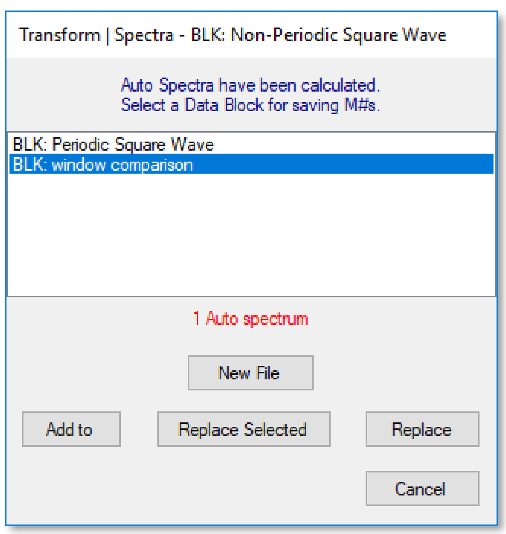


When the calculation is complete,

- ***Click*** on **New File** in the Data Block selection dialog box
- Enter a file name **"window comparison"** in the dialog box that opens, and ***click*** on **OK**

The new **BLK: window comparison** now holds the first spectrum, that was calculated using a **Rectangular** window.

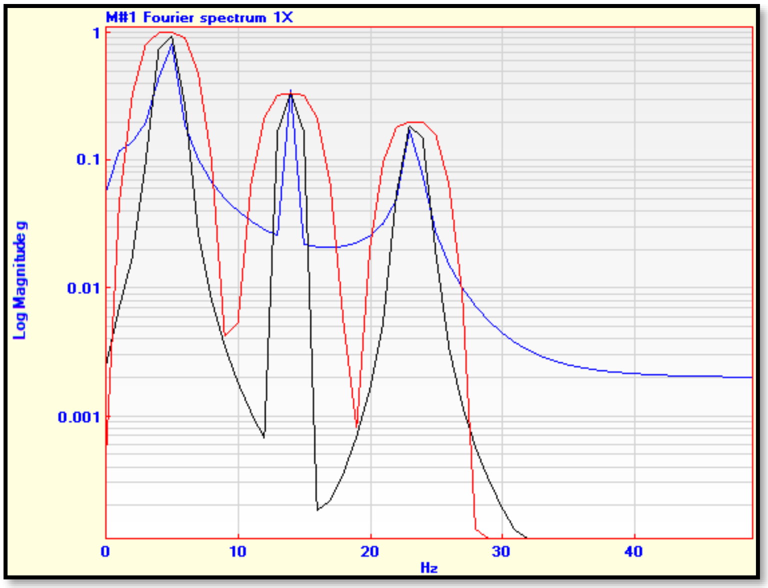
- Execute **Transform | Spectra again** in the **BLK: Non-Periodic Square Wave** window
- Select the **Hanning** window in the dialog box, and ***click*** on **Calculate**
- When the calculation is complete, select **BLK: window comparison** in the dialog box that opens, and ***press*** the **Add To** button



This adds the spectrum that was created using the **Hanning** window as **M#2** to **BLK: window comparison**.

- Execute **Transform | Spectra** again in the **BLK: Non-Periodic Square Wave** window.
- Select the **Flat Top** window in the dialog.box, and ***click*** on **Calculate**
- When the calculation is complete, select **BLK: window comparison** in the dialog box that opens, and ***press*** the **Add To** button

This adds the spectrum that was created using the **Flat Top** window as **M#3** to **BLK: window comparison**.



4 $^{2}/_{3}$ Hz square-wave analyzed using 3 different windows.

- Execute **Format | Overlaid**

This comparison plot clearly illustrates that either the **Hanning** or **Flat Top** window ***will eliminate most of the leakage***, making the ***three peaks*** **clearly** evident in the spectrum.

The **Flat Top** window provides the best magnitude estimates compared to the other two windows, as shown in the following table.

The ***periodic*** **14** Hz component ***did not contribute*** **leakage** to the **4.6667** Hz or **23.333** Hz peaks, but the magnitude at **14** Hz was affected by the leakage from the other two frequency components.

Peak	Rectangular	Hanning	Flat Top
4.66 Hz	-19.50	-6.95	0.00
14 Hz	7.01	-0.07	0.00
23.33 Hz	-13.85	-6.95	0.00

Percent Amplitude Error using 3 different windows.

The MEscope time window functions comply with ISO standard 18431-2

REFERENCES

Richardson, M., *Fundamentals of the Discrete Fourier Transform*, Sound and Vibration Magazine, March 1978.

Richardson, M., *Modal Analysis Using Digital Test Systems*, Seminar on Understanding Digital Control and Analysis in Vibration Test Systems, Shock and Vibration Information Center Publication, Naval Research Laboratory, Washington, D.C., May 1975.