



The FFT, Leakage, and Windowing


INTRODUCTION

ME'scopeVES features one-button commands that make it very convenient to look at signals in either the time or frequency domain. You can toggle between these domains at will without losing any signal characteristics.


 **Transform | FFT** converts a block of time-histories into equivalent frequency domain spectra.

 **Transform | Inverse FFT** converts a block of spectra back into the time domain.

A more general command lets you form or augment a block of linear Fourier Spectra, Auto Spectra or Power Spectral Densities from one or more time histories.

 **Transform | Spectrum** creates a new block of spectra of various forms from time histories.

Additionally, ME'scopeVES has a waveform synthesizer with which you can generate a broad variety of sine, periodic, impact and random signals.

 **File | New | Data Block** synthesizes time-histories.

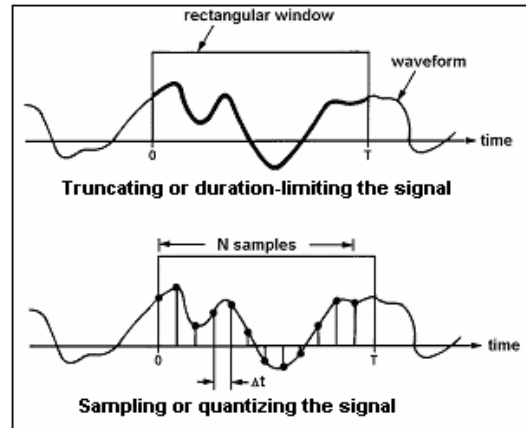
You will use these commands to review some basic properties of the Fast Fourier Transform (FFT) and to learn how common FFT "window leakage errors" can be avoided.

Steps in his Application Note can be duplicated using any package that includes VES-300 Signal Processing.

BASIC TIME & FREQUENCY RELATIONSHIPS

The FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a digitally measured time domain signal such as a time-history. There are three basic formulas that govern all DFT calculations.

Time Domain Equation



Time Domain Windowing & Sampling.

Firstly, the DFT assumes that the time domain signal is represented by N uniformly spaced samples of data, which satisfy the equation,

$$T = N\Delta t \tag{1}$$

where:

- T = total time of the sampled signal (in seconds)
- N = number of samples (block size)
- Δt = time step (increment) between samples

The schematic figure above depicts the mathematics of digitally characterizing a continuous (analog) time-history. This consists of two steps:

- 1) Multiply the signal by a rectangular observation window (also called a *boxcar* 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 observation times. This *quantizes* the measurement in time.

Note that the total observation time, T , of the sampled signal includes the time *up to but not including* sample $N+1$ (which may be considered as the first sample point of a subsequent observation block).

Frequency Domain Equation

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

$$f_{max} = \Delta f (N/2) \tag{2}$$

where:

- f_{max} = maximum frequency of the signal (in Hz)
- Δf = frequency step (increment) between samples (also called Lines)
- N/2** = frequency domain block size

f_{max} is termed the *Nyquist frequency* of the data. Note that a resulting spectrum has **N/2 + 1** samples; **DC** plus **N/2** frequencies. Note further that the **DC** value and the value at the Nyquist Frequency, f_{max} , contain no *phase* information. These **Lines** are *real valued*; all other **Lines** are *complex-valued*.

Nyquist sampling

Thirdly, the maximum frequency of the spectrum, f_{max} , is related to the time-domain sampling rate, f_s , specifically:

$$f_{max} = 2f_s/2 \tag{3}$$

where:

- $f_s = 1/\Delta t$ = the *sample rate* (samples/second) at which data is sampled.

Equation (3) is a statement of Shannon’s sampling theorem, namely: “The maximum frequency of a spectrum is one half the sampling rate of its corresponding time domain signal”.

Solving the frequency domain equation for Δf , and substituting for $2f_{max}$ gives a fourth important relationship:

$$\Delta f = 2f_{max} / N = 1 / N\Delta t = 1/T \tag{4}$$

This final formula tells us that the frequency resolution, Δf , of a digital spectrum is always equal to the reciprocal of the observation time, **T**, of the signal “snapshot”. In other words, to obtain finer frequency resolution (smaller Δf), a signal must be sampled over a longer period of time, *not at a higher sampling rate, f_s* .

SYNTHESIS OF A PERIODIC SIGNAL

Let’s start our study by synthesizing a periodic signal and analyzing its characteristics. The Fourier Series of a symmetric square-wave of amplitude ± 1 and frequency, f_0 , is given by:

$$y(t) = \frac{4}{\pi} \sum_{k=0}^K \frac{1}{(2k+1)} \text{Sin}[2\pi(2k+1)f_0t] \tag{5}$$

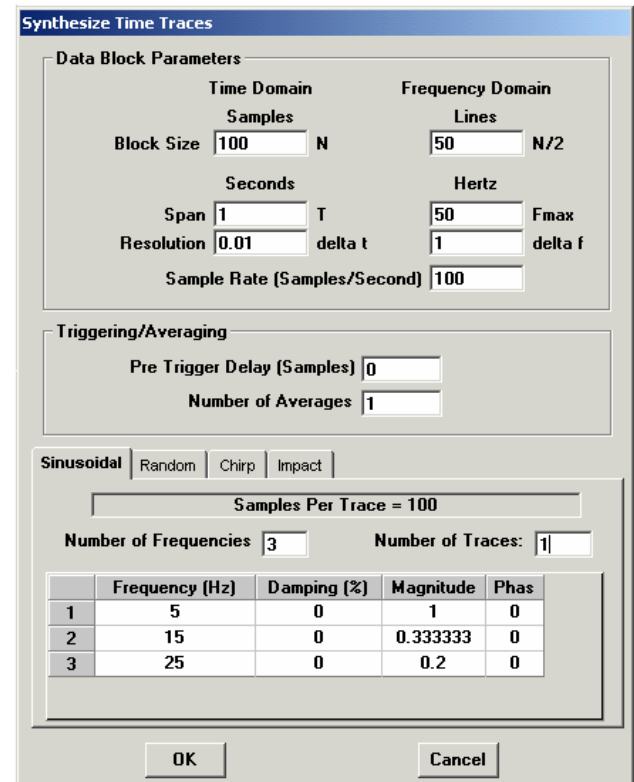
Hence the first three sine waves of this series are:

$$y(t) = \text{Sin}(2\pi f_0t) + \frac{1}{3} \text{Sin}(6\pi f_0t) + \frac{1}{5} \text{Sin}(2\pi f_0t) \tag{6}$$

They approximate a square-wave of $\pm \pi/4 = 0.795$ amplitude with a fundamental frequency, f_0 , of 5 Hz.

 Execute: **File | New | Data Block**.

A **Synthesize Time Traces** dialog (below) will open.



	Frequency (Hz)	Damping (%)	Magnitude	Phas
1	5	0	1	0
2	15	0	0.333333	0
3	25	0	0.2	0

Synthesis dialog box showing entry of 5 Hz squarewave.

- In the **Data Block Parameters** section, make the following entries:

Block Size (N) = 100 Points

Sample Rate (fs) = 100 Samples/Second

Note that entry of *any two* of these **Data Block Parameters** establishes the remaining four parameters. Your entries have established:

T = 1 Sec

ΔT = 0.01 Sec

f_{max} = 50.0 Hz

Δf = 1.00 Hz

- Accept the default value for **Starting Time = 0**.
- Accept the default values in the **Trigger / Averaging Parameters** section:

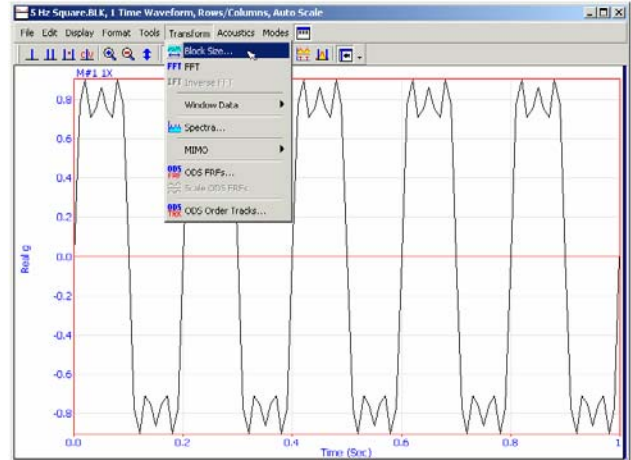
Pre-Trigger Delay = 0 Samples

Number of Averages = 1

On the **Sinusoidal** tab at the dialog's bottom:

- Select **Number of Frequencies = 3**.
- **Select Number of Traces = 1**.
- Enter **Frequency (Hz)** values of **5, 15** and **25** Hz.
- Enter **Magnitude** values of **1, 1/3** and **1/5** g.
- Verify that all **Phase** and **Damping** values remain at the default **0** value.
- Press **OK**.

A new Data Block (**BLK**) Window will open, displaying the 5 Hz square-wave approximated by the three sine waves just defined. Note that the amplitude (in the center of the "ringing") approximates **0.795 g** as anticipated. Observe that *exactly* five cycles of this periodic waveform are captured in the **1.00** second **T** duration of the window.

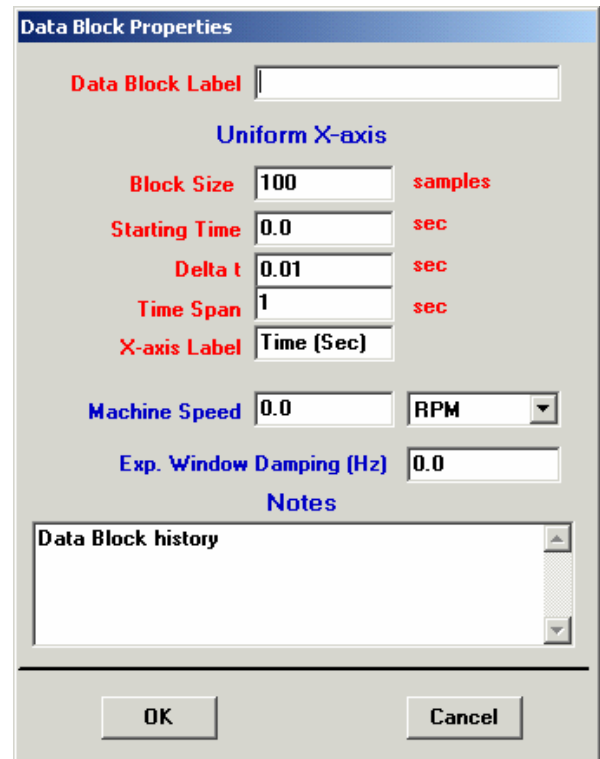


5 Hz square-wave approximated by harmonics 1, 3 & 5.

To verify the **Data Block Parameters** assigned to this synthesized measurement:



Execute: **File | Properties**.



Data Block Properties for synthesized 5 Hz square-wave.

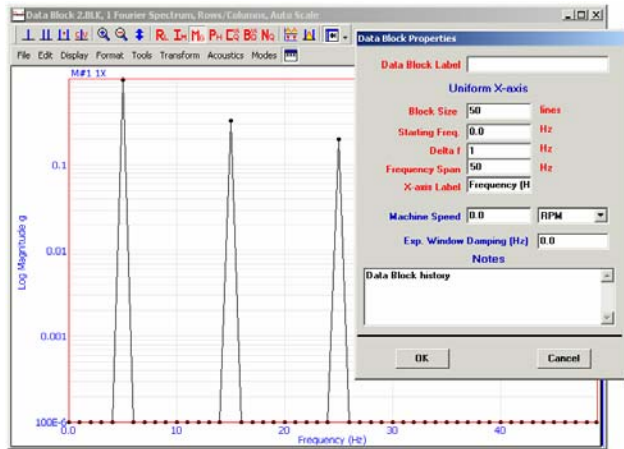
The corresponding **Data Block Properties** Window will open. Verify that the tabulations agree with your prior **N**, **T**, **ΔT** and **Starting Time** entries to **Synthesize Time Traces**.

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

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

 Execute: **Transform | FFT**.

The signal is transformed to a complex spectrum and the **Data Block Properties** reflect this. The block size now reflects number of frequency **Lines** ($N/2$) as well as Δf and the f_{max} .



FFT spectrum of synthesized 5 Hz square-wave.

Note the sharp spikes at 5, 15 and 25 Hz. Use the cursor to precisely measure the amplitude and frequency at each peak.

 Select: **Display | Cursor | Line Cursor**.

 Select: **Display | Cursor | Cursor Values**.

Note that the spectrum faithfully captures the 1, 1/3 and 1/5 g amplitudes synthesized for the first, third and fifth harmonics of this periodic waveform.

 Execute: **File | Save**.

- Save your work as **5 Hz Square.BLK** and close its Window.


LEAKAGE, A FUNDAMENTAL DFT PROBLEM

In the preceding study, we synthesized three components at 5, 15 and 25 Hz using a data block with a T of 1.00 second. Each of the square-wave components went through *exactly an integer number of cycles* (5, 15 and 25 respectively) during 1 Sec (T) of time. As a result, each resulting spectral peak fell *exactly* upon an integer multiple of Δf (1.00 Hz).

When T coincides exactly with N cycles of a signal being analyzed, everything works perfectly. Amplitudes and frequencies are properly identified with high precision, as we have just seen.

Truncated or Aperiodic Signal

Let's repeat our square-wave experiment, but with a single change. We will simply move the fundamental (f_0) frequency from 5 Hz to $4\frac{2}{3}$ Hz, leaving all other processing parameters as they were.

 Execute: **File | New | Data Block** and make the revised entries.

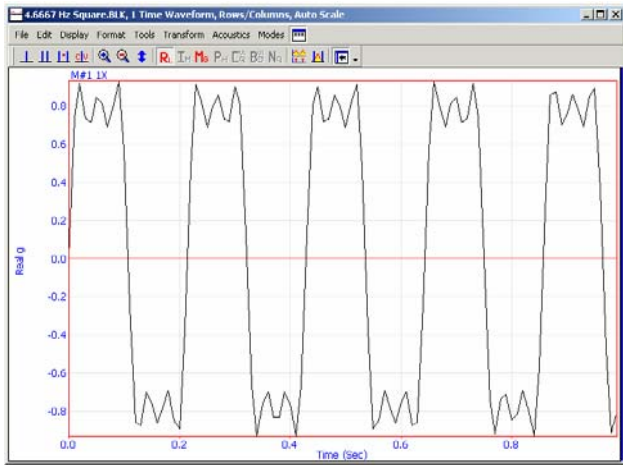
	Frequency (Hz)	Damping (%)	Magnitude	Phas
1	4.666667	0	1	0
2	14	0	0.333333	0
3	23.333333	0	0.2	0

Synthesis dialog entry for the $4\frac{2}{3}$ Hz square-wave.

The *only* difference between this synthesis and the first one are the **Frequency (Hz)** entries to the **Sinusoidal** tab.

- Change the frequencies to **4.666667, 14** and **23.333333 Hz** (1, 3 and 5 times $4\frac{2}{3}$ respectively).

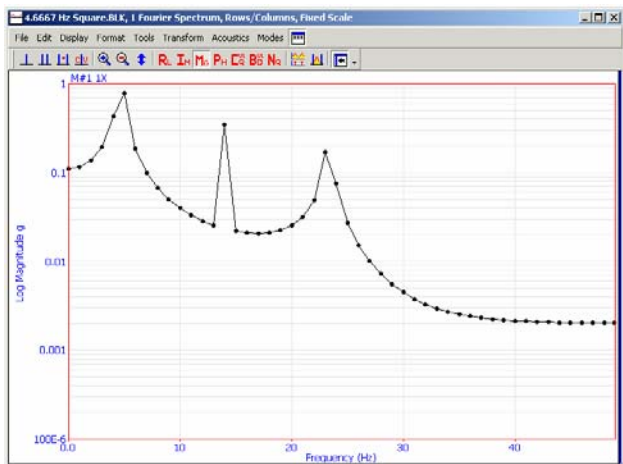
Note that the 3rd harmonic (14 Hz) remains an *exact integer multiple* of Δf and that it will have *exactly* 14 cycles within the 1.00 second T. However, the first and fifth harmonics are not periodic in the window.



$4\frac{2}{3}$ Hz square-wave synthesized in 1 second block.

The $4\frac{2}{3}$ Hz synthesis produces no surprises. The waveform exhibits the same shape and peak amplitude seen in the 5 Hz case. The sole difference is that the **1.00** second window captures $4\frac{2}{3}$ cycles instead of **5**. On the surface, this seems like a small point of differentiation. You are now qualified to be surprised!

FFT Execute: **Transform | FFT**.



Spectrum of synthesized $4\frac{2}{3}$ Hz square-wave.

Clearly, the resulting FFT spectrum is disappointing. While three peaks may still be seen, the general character of the spectrum is far from intuitively obvious. There is significant amplitude at *all* frequencies from Δf to f_{max} , even though we know that only *three* discrete and well-separated tones were used to create the signal.

Note further that the *magnitude* of the 1X and 5X terms are less than the correct values. Only the **14** Hz “cell-centered” 3X term has a peak of correct magnitude. You have just been introduced to *spectral leakage*.

WHAT CAUSES LEAKAGE?

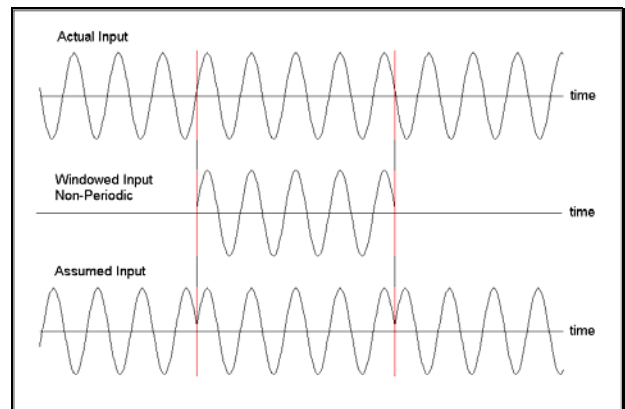
Whenever we use the FFT on a signal that is *not exactly periodic in the window*, leakage will occur. It is the result of multiplying two signals together in one domain, which is equivalent to *convolving* their transforms in the other domain. Convolution is a process of shifting and adding together 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, and the resultant frequency spectrum was the convolution of the spectrum of the sine wave with the spectrum of the rectangular window.

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.



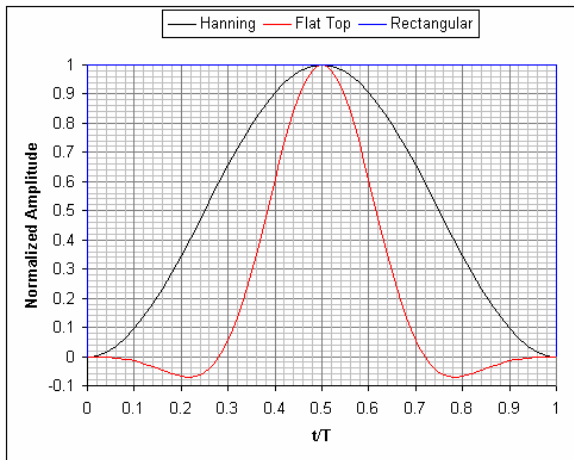
Repetition of a Non-Periodic Signal.

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 sine wave 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 the sampling window, leakage cannot be completely eliminated. However, it can be reduced significantly so that the resulting spectrum presents an accurate representation of the signal. This is accomplished by *changing the shape* of the capture window, which is multiplied by the time-history.



Window shapes: Rectangular, Hanning and Flat Top.

ME'scopeVES has three built-in capture windows for processing data: **Rectangular**, **Hanning**, and **Flat Top**. So far, you have used the Rectangular window, which is *always used by default* when an FFT is performed.

The **Rectangular** window is the proper choice for analyzing transients that are completely captured within the window's T duration. It is also the proper choice when analyzing periodic signals using *order-normalized* sampling, a process by which f_s is varied in proportion to the signal frequency. This is a common technique for the analysis of rotating machinery.

In virtually all other circumstances, the **Rectangular** choice is inappropriate and the resulting spectrum will contain leakage. The **Hanning** window is the normal choice when analyzing broad-band random signals, cyclic signals, or a mixture of the two. It provides excellent suppression of truncation artifacts while retaining good *selectivity*, the ability to separate closely spaced tones or to detect a tone buried in a noisy background.

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** shape. Hence its use is application specific.

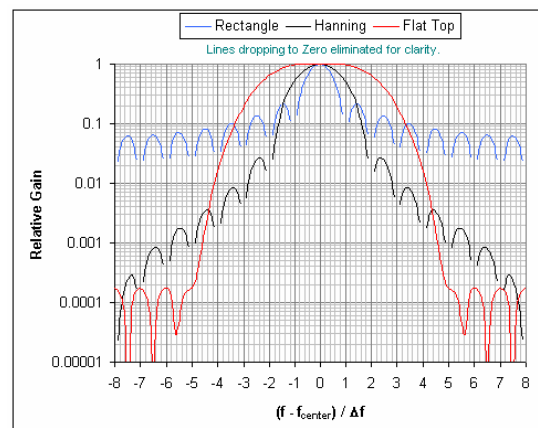
Both the **Hanning** and **Flat Top** windows are designed to optimize the FFT analysis of continuous (never transient!) signals. They do this by symmetrically "tapering" the leading and trailing edges of the observation to (essentially) zero. This has the effect of forcing the time-history to have the appearance of being periodic in T .

While the **1** and **0** nature of the **Rectangular** window merely acts as a time gate, *multiplying* the measured (or synthesized) signal by a **Hanning** or **Flat Top** window changes the wave-shape of the signal prior to application of the FFT. Yes, applying such a window does distort the shape subjected to transformation. On balance, however, the effect is highly beneficial.

FFT Filters

We might view this from yet another perspective. The FFT numerically implements a bank of parallel detected filters. Each filter is of the same constant bandwidth (nominally Δf) and the 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, nature has not blessed us with such ideal filters (either physical or mathematical). Instead, the transfer function of each FFT filter is a complicated shape. That shape is, in fact, the Fourier Transform of the sample window. The FFT filter shapes using **Rectangular**, **Hanning** and **Flat Top** windows are presented below.



Magnitude of FFT filter transfer functions.

Note that the FFT filters for *any* window are far broader than Δf , have a lobed shape and exhibit multiple zeros. Each exhibits maximum gain (sensitivity) at the *center frequency*, f_{center} . Adjacent filter center frequencies are spaced Δf apart, so considerable *overlap* exists between filters.

The **Rectangular** filter has the narrowest center lobe, (a desirable attribute) spanning $2 \Delta f$ between the adjacent zeros. Outside of the center lobe, the **Rectangular** window exhibits zero gain at every multiple of Δf . It also has the highest sidelobe amplitudes and the greatest curvature across the center lobe. These are detracting characteristics.

Consider, for a moment, why this window works so well with sine tones that are periodic in the capture window. In this situation, a signal component is presented at *exactly* the center frequency of one of the bank's filters. That filter responds enthusiastically as the component is coincident with its most sensitive frequency. More importantly, *none of the other filters respond at all because this frequency corresponds to a zero gain point for all other filters.*

Now we can understand why a single truncated or "non-centered" sine excites spectral activity at all points in the spectrum, causing leakage. When the tone moves slightly ($\pm \Delta f / 2$ or less) off of one filter's center, it *no longer coincides with the zeros of the adjacent filters.*

The **Hanning** window improves upon this situation by reducing the height of the first sidelobes and increasing the rate at which subsequent sidelobes decay in amplitude with frequency (from f_{center}). The reduced sidelobe gain *minimizes leakage* of signal energy to inappropriate frequency locations. This comes at the expense of a broader center lobe with $4 \Delta f$'s between the bounding zeros, which reduces the effective resolution.

The **Flat Top** window has an extremely wide center lobe, spanning more than $8 \Delta f$. As the name implies, the top of the center lobe is extremely flat (within $\pm 0.1 \%$), providing accurate amplitude information even when a tone varies as much as $\pm \Delta f / 2$ from f_{center} . While the first sidelobes are well suppressed, the subsequent ones do not roll off with increasing frequency. They are all of nearly equal amplitude.

The *noise bandwidth*, β_n , is the standard measure of a window's random noise performance. It is the bandwidth of an ideal rectangular filter that passes the same broad-band power when both are subjected to white noise. β_n is the "per Hz" frequency of PSD amplitude.


	β_n	Flatness %	1 st Sidelobe dB	Roll-off dB/decade
Rectangular	Δf	36.24	-13.2	-20
Hanning	$1.5 \Delta f$	15.12	-32.1	-60
Flat Top	$3.77 \Delta f$	0.11	-93.0	none

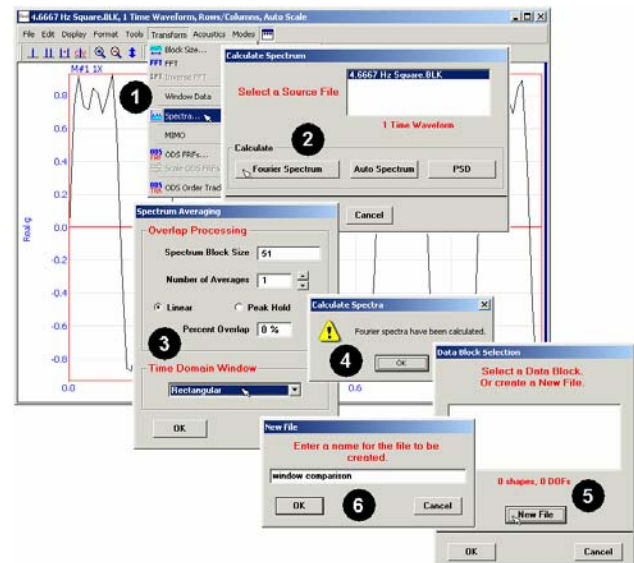
PROOF OF PERFORMANCE

Let's reduce leakage in our $4 \frac{2}{3}$ Hz square-wave spectrum by applying a more appropriate window. Start by transforming the last spectrum measured back into a time history.

 Execute: **Transform | Inverse FFT.**

Now we will build a new **Data Block** containing three different spectra, all from the same synthesized data. These spectra will differ by the window function applied.

-  Execute: **Transform | Spectrum.**
- Press **Fourier Spectrum** in the **Calculate Spectrum** dialog.
- Select the **Rectangular** window in the **Spectrum Averaging** dialog, then click **OK**.
- A **Calculate Spectra** box will open informing you that the calculation is completed. Click on **OK**.
- Click **New File** in the **Data Block Selection** dialog and then press **OK**.
- Enter a file name such as **window comparison** in the **New File** dialog and click **OK**.



Creating a new Spectrum Data Block.

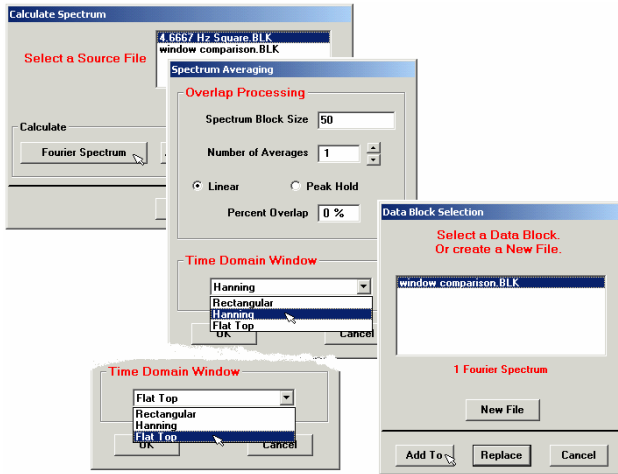
The new **window comparison.BLK** now holds the first spectrum (that using a **Rectangular** window) as **M#1**.

- Repeat preceding steps *twice* with the following small changes.

In step 3 select **Hanning** for the first repetition and **Flat Top** for the second.

In step 5 do not click **New File**. Instead, verify that **window comparison.BLK** is the active choice and press **Add To**.

This adds the spectrum made using **Hanning** as **M#2** and the **Flat Top** spectrum as **M#3**.



Augmenting an existing Spectrum Data Block.

Execute: **File | Save**.

- Save your work as **4.667 Hz Square.BLK** and close its Window.

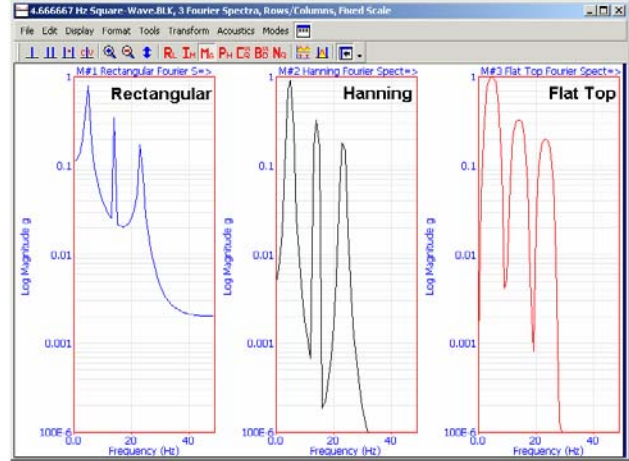
Shift your attention to the **window comparison.BLK**. Certain graphic commands simplify comparing the three traces.

Format | Overlay Traces allows all selected traces to be overlaid in a single plot Window.

Format | Rows/Columns facilitates graphic layout of multiple plots, each with one trace.

Format | Vertical Axis controls the vertical axis format and scaling

Format | Horizontal Axis controls horizontal axis format and scale.



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

This comparison plot clearly illustrates that either the **Hanning** or **Flat Top** window will eliminate most of the leakage, making the proper “3 peak” shape of the spectrum clearly evident. The **Flat Top** window provides the best amplitude estimates compared to the synthesis entries, as evidenced by the following table.

Note that the “cell-centered” 14 Hz component did *not* contribute leakage to the 4.6667 Hz and 23.3333 Hz terms, but the converse is not true. The amplitude at 14 Hz was affected by the leakage from the other two “off-center” tones.

Peak @	Rectangular	Hanning	Flat Top
5 Hz	-19.50	-6.95	0.00
14 Hz	7.01	-0.07	0.00
23 Hz	-13.85	-6.95	0.00

Percent Amplitude Error using 3 different windows.

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.

ME'scopeVES window functions comply with ISO 18431-2