Vibrant Technology, Inc.

ME'scopeVES Application Note #1

The FFT, Leakage, and Windowing

INTRODUCTION

The **Transform | FFT** and **Transform | Inverse FFT** commands in ME'scopeVES make it very convenient to look at signals in either the time or frequency domains. Also, ME'scopeVES has a waveform synthesizer in the **File | New | Data Block** command with which you can synthesize damped sine wave signals. This command will be used to illustrate the properties of the FFT, and show how errors can occur if it is not used properly.

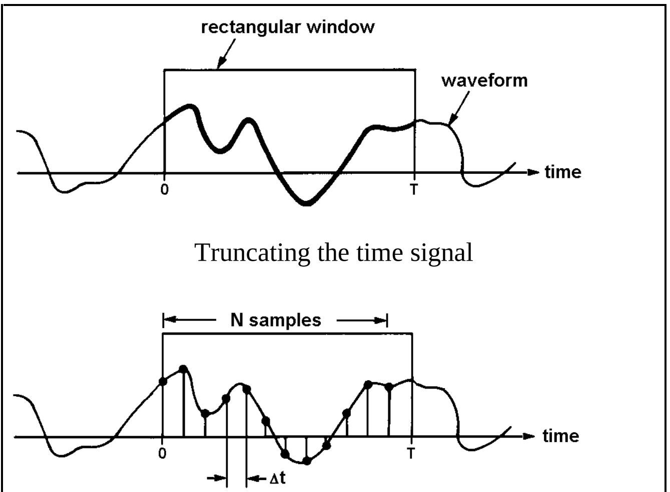
FFT Formulas

The FFT is an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a sampled time domain signal. There are three basic formulas that govern all FFT calculations. The FFT assumes that the time domain signal is represented by N uniformly spaced samples of data, which satisfy the equation,

Time domain equation: $T=N\Delta t$

where:

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



Truncating the time signal

Sampling the time signal

Figure 1. Time Domain Windowing & Sampling

Figure 1 depicts the process of sampling a continuous time signal over a finite period of time. It consists of two separate steps: 1) multiplying the signal by a rectangular window, also called a boxcar function; and 2) sampling the truncated signal. This window of sampled data is also called the ***sampling window.***

Notice that the total time length (T) of the sampled signal also includes the time step ***up to but not including*** the $(N+1)^{st}$ sample.

> **NOTE:** In ME'scopeVES, the ending time of a time domain Trace is the time corresponding to the N^{th} sample, which is ***one Δt less than the total time T***.

The FFT also assumes that the frequency domain spectrum of the signal is represented by $N/2$ uniformly spaced complex valued samples (magnitudes and phases), which satisfy the equation,

Frequency domain equation: $F_{max} = \Delta f\ (N/2)$

where:

F_{max} = maximum frequency of the signal (in Hz)
Δf = frequency step (increment) between samples
$N/2$ = frequency domain block size

> **NOTE:** In ME'scopeVES, the ending frequency of a frequency domain Trace is the frequency corresponding to sample $N/2$, which is ***one Δf less than the maximum frequency F_{max}***.

Finally, the maximum frequency of the frequency domain spectrum is related to the time domain sampling rate, referred to as the Nyquist sampling rate. This formula is also 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***",

Nyquist sampling: $f_s = 1/\Delta t = 2F_{max}$

Solving the frequency domain equation for Δf, and substituting for $2F_{max}$ gives,

$$\Delta f = 2F_{max}/N = 1/N\Delta t = 1/T$$

This final formula says that the frequency resolution (Δf) of a digital spectrum is always equal to the inverse of the total time length (T) of the sampled signal. In other words, to obtain finer frequency resolution (a smaller Δf), a signal must be sampled over a longer period of time.

SYNTHESIZING A PERIODIC SIGNAL

We will now look at the FFT of a ***periodic*** versus a ***non periodic*** sine wave signal. A sine wave signal is said to be ***periodic in the window*** if an integer number of complete cycles of the sine wave are contained within the sampling window. Figure 3 shows an example of a sine wave that is periodic in the window. We will also see that any sine wave that is periodic in the window will also have a frequency that ***coincides exactly*** with one of the frequency samples in its spectrum.

- Execute the **File | New | Data Block** command in the ME'scopeVES window. Name the Data Block file **3125.BLK**, and click **OK** to save it to disk. The **Synthesize a Time Trace** dialog box will open.

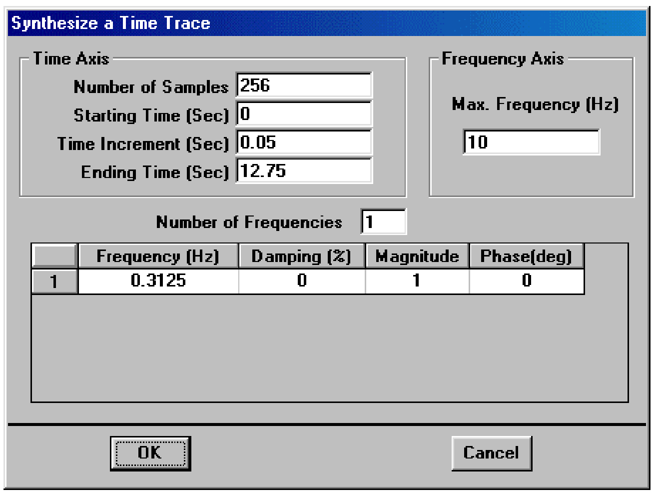


Figure 2. Dialog Box for Periodic Sine.

- Enter the following numbers into the dialog box, shown in Figure 2

  Number of Samples: 256
  Max. Frequency: 10 Hz
  Number of Frequencies: 1
  Frequency: 0.3125 Hz
  Damping (%): 0
  Magnitude: 1

These parameters will synthesize a **0.3125** Hz sine wave with an amplitude of **1.0**, and no damping.

- When all of the parameters are entered, press the **OK** button to synthesize the sine wave. The Data Block window will open with the sine wave in it.

Notice that the Trace has ***exactly 4 cycles*** of the sine wave in it. This signal is ***periodic in the window***, since an integer number of cycles has been sampled (in this case synthesized) within the sampling window.

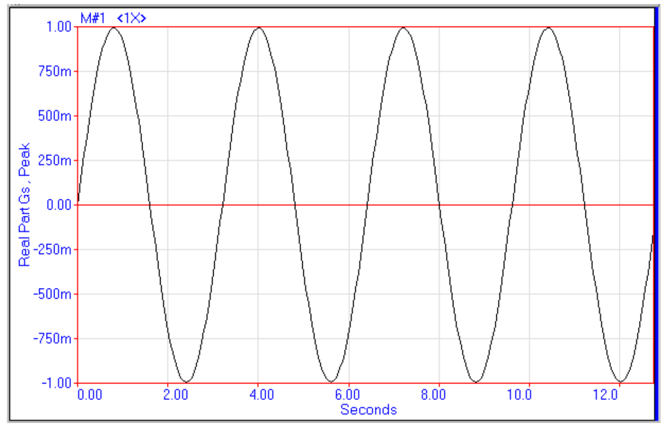


Figure 3. 0.3125 Hz Sine Wave.

- Execute the **Transform | FFT** command.
- Execute **Display | Magnitude | Linear** to display the magnitude, and **Zoom** in around the peak in the resulting spectrum, as shown in Figure 4.
- Turn on the line cursor and move it to the peak.

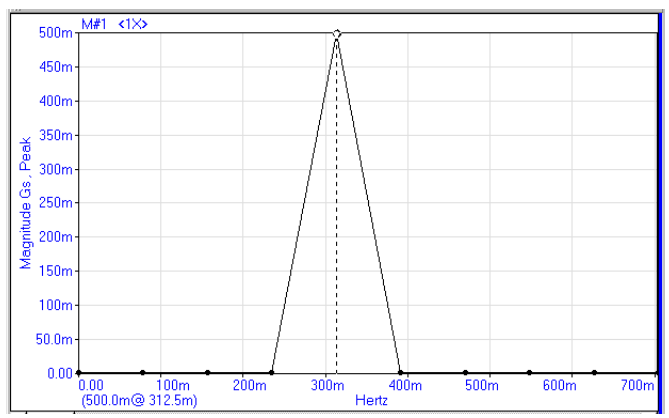


Figure 4. Spectrum of 0.3125 Hz Sine Wave.

Notice that the spectrum contains a peak at **0.3125** Hz, and that its amplitude is **0.5**. This is so because the Fourier transform represents the spectrum of all real valued time signals with ***all frequencies, both positive and negative***. However, the negative frequency portion of the spectrum is never displayed because it looks exactly like the positive frequency spectrum. That is, the Fourier spectrum is symmetric about the origin (zero frequency).

Therefore, the other half of the sine wave spectrum is represented by a negative frequency peak at **-0.3125** Hz, also with an amplitude of **0.5**. (Notice also that the peak occurs at the 5^{th} sample.)

- Execute the **File | Properties** command. The Data Block Properties dialog box will open, as shown in Figure 5.

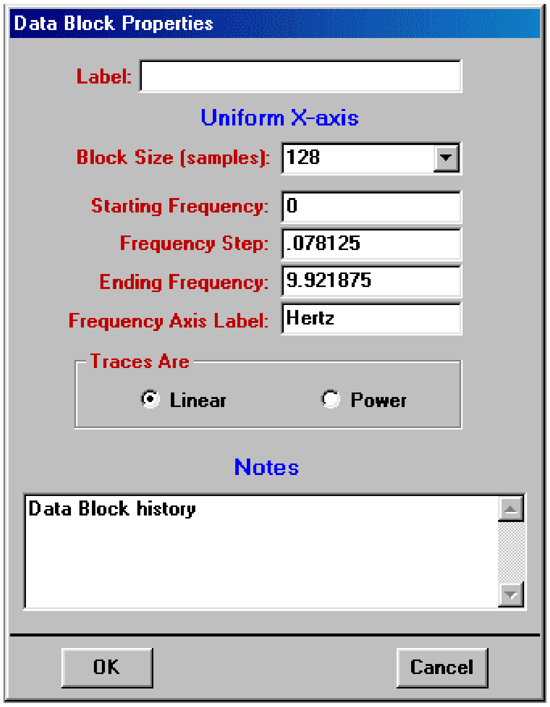


Figure 5. Data Block Info Dialog Box.

Let's check the numbers.

- The Block Size is 1/2 of the time domain block size, or 128 as expected.
- The frequency step (Δf) should be the inverse of the total time of the sampling window,

  Δf= 1/ T= 1/(12.75+0.05)= 1/12.8= 0.078125 Hz.

- The ending frequency is 10 Hz minus one frequency step, or **10.0 - 0.078125 = 9.921875 Hz.**

To synthesize a sine wave with a frequency that lies exactly on one of the frequency samples, we picked a percentage of the frequency domain block size that yielded an integer number. We chose **3.125**%, so we could expect the FFT of the sine wave to yield a peak at 4 Δfs, (the 5[th] sample), which is exactly where it lies.

No. of Δfs= 0.03125 x 128 = 4

An ideal result like this can always be expected when any time domain signal that is periodic in the window is FFT'd to obtain its DFT. But, we cannot expect most real world signals to be perfectly periodic in the sampling window.

SYNTHESIZING A NON PERIODIC SIGNAL

Now, let's synthesize another sine wave with a frequency that lies half way between sample nos. **5** and **6** (or **4.5 Δfs)** in the frequency domain. We'll enter a percentage of Fmax equal to,

% of Max. Frequency = 4.5/128 => 3.515%

Frequency = (3.515%) (10 Hz) => 0.3515 Hz

- Execute the **File | New | Data Block** command in the ME'scopeVES window again. Name the Data Block file **3515.BLK**, and click **OK** to save it to disk. The **Synthesize a Time Trace** dialog box will open, as shown in Figure 6.

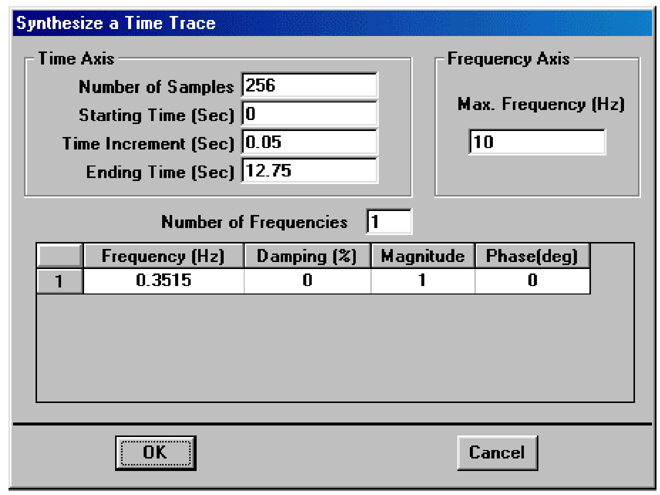


Figure 6. Dialog Box for Non Periodic Sine.

- Enter the following numbers into the dialog box,

  Number of Samples: 256
  Max. Frequency: 10 Hz
  Number of Frequencies: 1
  Frequency: 0.3515
  Damping (%): 0
  Magnitude: 1

These parameters will synthesize a **0.3515** Hz sine wave with an amplitude of **1.0**, and no damping.

- When all of the parameters are entered, press the **OK** button to synthesize the sine wave. The Data Block window will open with the sine wave in it.

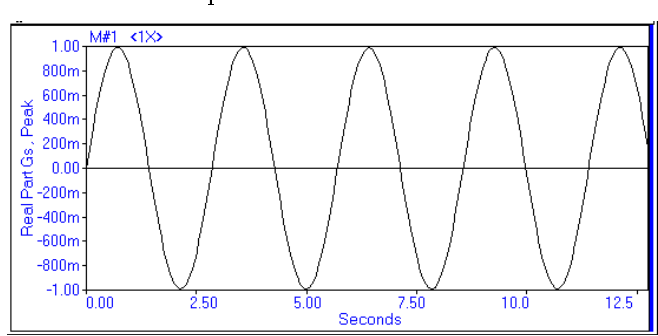


Figure 7. 0.3515 Hz Sine Wave

- Execute the **Transform | FFT** command.
- Zoom in around the peak in the resulting spectrum, as shown in Figure 8.
- Turn on the line cursor and move it to the peak.

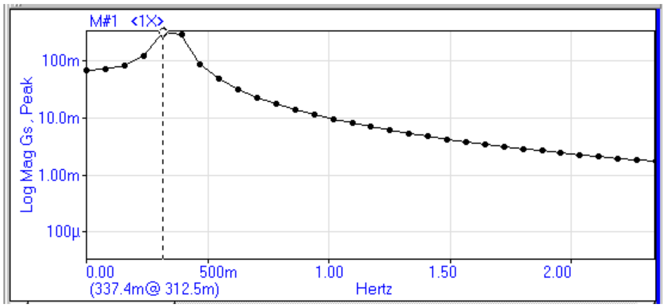


Figure 8. Spectrum of 0.3515 Hz Sine Wave.

The spectrum in Figure 8 still contains a peak at **0.3125** Hz, but the amplitude of the peak is **0.337**, instead of **0.5**. This is a **33% error** in the amplitude.

Notice also that there is a lot more of the signal at frequencies surrounding the peak. In other words, the spectrum has been ***smeared*** in the vicinity of the expected peak. This smearing is called ***leakage***. The signal energy has ***leaked out*** of the frequency bin where it should have been (bin **5.5**, where there is no sample!!) into the surrounding bins.

WHAT CAUSES LEAKAGE?

Whenever we use the FFT on a signal that is ***non 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 the non periodic sine wave, its 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 in Figure 9.

The actual signal is a continuous sine wave. The finite length windowed version is non-periodic in its sampling window. The FFT computes the spectrum of the ***assumed signal***, which when repeated outside of its sampling window, is no longer a continuos sine wave.

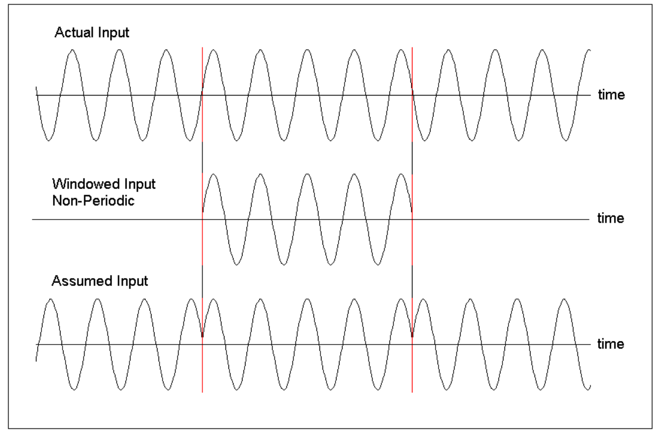


Figure 9. Repetition of a Non-Periodic Signal.

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 signal is non periodic in its sampling window, leakage cannot be completely eliminated, but it can be reduced so that the resultant spectrum is still usable.

ME'scopeVES has three built-in time windows for processing data: Rectangular, Hanning, and Flat Top. We have already used the Rectangular window, which is always used by default when an FFT is performed. Now, let's try the other two windows.

- Execute the **Transform | Inverse FFT** command to recover the non periodic sine wave in Data Block **3515.BLK**
- Execute the **Transform | Spectrum Analysis** command. A dialog box will open. Choose the Data Block **3515.BLK** and execute **Calculate | Linear Spectrum**.
- The Spectrum Averaging dialog box will open, as shown in Figure 10.
- Select the **Flat Top** window from the drop down list, press **OK,** and select a new Data Block name.

Now, a linear spectrum (which is the spectrum computed with the FFT) of the non periodic sine wave is computed and shown in the Data Block window. (See Figure 11.) However, this time, before the FFT was computed, the Flat Top window was applied to the time domain data.

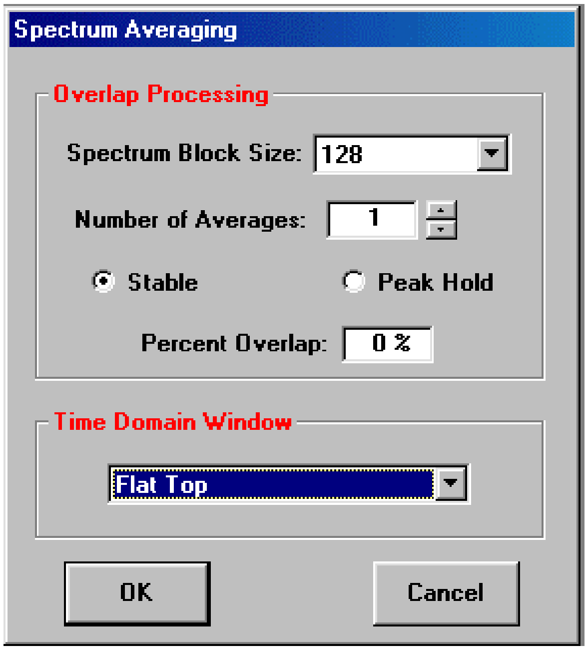


Figure 10. Spectrum Averaging Dialog Box.

The peak is ***between*** samples **5** & **6** (or **0.3125** & **0.3906** Hz), which is where we expect it. The amplitude of the peak is **0.455,** a **9 % error** from the expected amplitude of **0.5**.

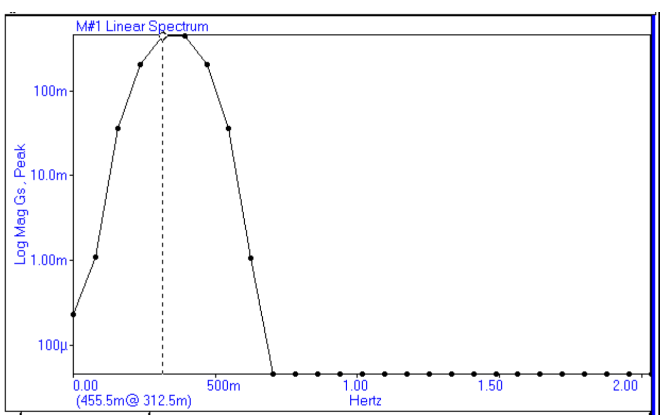


Figure 11. Linear Spectrum after Flat Top Window.

To see the effect of the Flat Top window on the time waveform (shown in Figure 12),

- Execute the **Transform | Inverse FFT** command.

Notice that the Flat top window smoothly reduces the sine wave to zero at both ends of the sampling window. This new signal is now completely contained within the window, making it ***periodic in the window***. However, it is no longer a constant amplitude sine wave function, so we can't expect its spectrum to be a single frequency peak either.

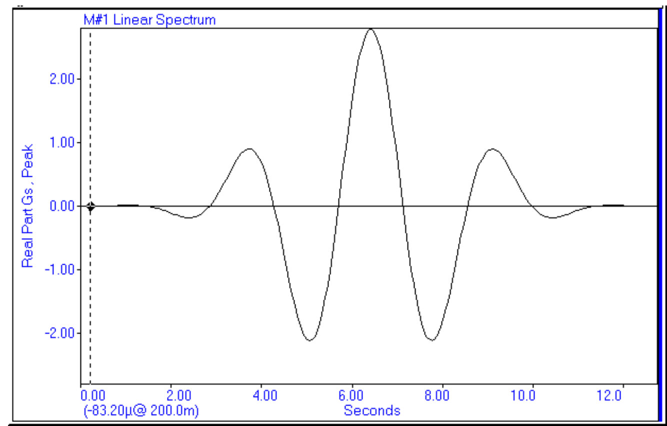


Figure 12. Time Signal After Flat Top Windowing.

Now, let's use the Hanning window on the same sine wave.

- Execute the **File | Add** command in the ME'scopeVES window, and re-open the **3515.BLK** file.
- Execute the **Transform | Spectrum Analysis** command. A dialog box will open. Choose the Data Block **3515.BLK** and execute **Calculate | Linear Spectrum**.
- Select the **Hanning** window from the drop down list, press **OK**, and select a new Data Block name.

The Data Block will show the linear spectrum of the non periodic sine wave, with the Hanning window applied. (See Figure 13.)

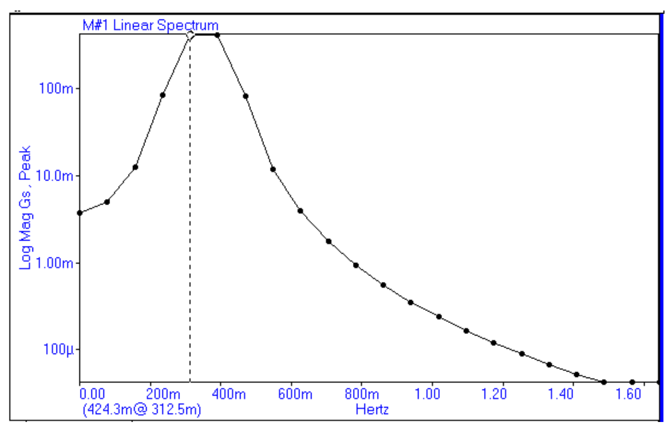


Figure 13. Linear Spectrum after Hanning Window.

Again, the peak is ***between*** samples **5** & **6** (or **0.3125** & **0.3906** Hz), which is where we expect it. The amplitude of the peak is **0.424**, a **15 % error** from the correct amplitude of **0.5,** and not as good as the Flat Top window.

Notice, however, that the Hanning windowed spectrum is narrower (less leakage) than the Flat Top spectrum. To display the time waveform multiplied by the Hanning window, (shown in Figure 14),

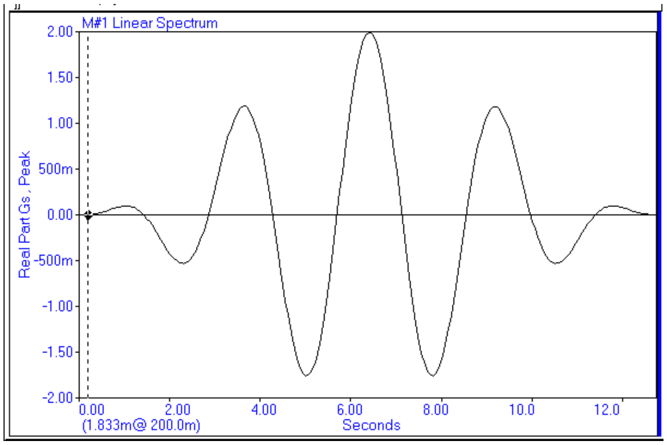


Figure 14. Time Signal After Hanning Windowing.

- Execute the **Transform | Inverse FFT** command.

Both the Hanning and Flat Top windows, though different, did the same thing to the sine wave. They ***smoothly*** reduced it to zero at the beginning and end of the sampling window.

The Flat Top and Hanning windowed sine waves are completely contain in the sampling window, and are therefore ***periodic in the window***. Consequently, their corresponding linear spectra are ***leakage free***. These spectra are only approximations of the spectrum of the original sine wave, but nevertheless, they are still useful for providing relatively accurate frequency and amplitude estimates.

CONCLUSIONS

- If a signal is ***non-periodic in the sampling window***, then a special window (Hanning, Flat Top, etc.) must be used to reduce leakage in its spectrum.
- In general, the Flat Top window is used for sinusoidal (or narrow band) signals, because it gives spectrum amplitudes that are more accurate than the Hanning window.
- In general, the Hanning window is used for random (or wide band) signals, because it preserves narrow peaks better than the Flat Top window.
- If they are completely contained within the sampling window, transient signals don't require any special window other than the default Rectangular window.
- Special types of swept sine (chirp) and random (burst random) excitation signals have been developed for yielding leakage-free measurements. These signals are specially synthesized in an analyzer to ensure that the measured (excitation & response) signals are periodic in the sampling window.

REFERENCES

1. "Fundamentals of the Discrete Fourier Transform", Sound and Vibration Magazine, March 1978.
2. "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.