



MEscope Application Note 02

Waveform Integration & Differentiation

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

[Click here](#) to download the MEscope Demo Project file for this App Note.

INTRODUCTION

The most common type of vibration transducer is an accelerometer, which measures acceleration. However, in order to answer the question **“How much is the machine or structure really moving?”** a common requirement of signal processing is to integrate acceleration (or velocity) signals to displacements.

In this note, we will exercise both the integration & differentiation methods in MEscope. Integration & differentiation can be done on either time domain or frequency domain waveforms. We will see how **DC offsets and leakage** can cause errors when integrating or differentiating waveforms, and how these errors can be effectively dealt with.

MEscope has a built-in Fast Fourier Transform (or FFT). Using the FFT, any Data Block of waveforms can be transformed between the time & frequency domains without loss of information. Therefore, integration & differentiation can be performed on any Data Block of time or frequency domain waveforms.

TIME DOMAIN INTEGRATION & DIFFERENTIATION

Time domain waveforms are integrated & differentiated in MEscope by transforming them to the frequency domain using the FFT, performing frequency domain integration or differentiation on their Digital Fourier Transform (or DFT), and inverse transforming them back to the time domain using the Inverse FFT.

The main difficulty with any time domain integration method is that any DC offset in a signal **must be removed before integration** is performed on it. Otherwise, the integrated DC offset will dominate the result.

Also, if the time waveform is **not periodic (or is completely contained) within its sampling window**, leakage will occur in its spectrum. Like a DC offset, leakage will cause more errors in the integrated or differentiated waveform.

FREQUENCY DOMAIN INTEGRATION

Frequency domain integration is done by **dividing each sample** of the frequency spectrum ($X_i(2\pi f_i)$) by the sample frequency ($j2\pi f_i$). Time domain waveforms are integrated by using the following **equivalent frequency domain operation**.

$$\int x(t)dt \Leftrightarrow \frac{X_i(2\pi f_i)}{(j2\pi f_i)}, i=1,\dots,N/2$$

where:

$X_i(2\pi f_i)$ = Fourier spectrum (DFT) of the signal for the i^{th} sample

$(j2\pi f_i)$ = frequency of the i^{th} sample (in rad./sec), j - denotes the imaginary operator

f_i = frequency of the i^{th} sample (in Hz)

FREQUENCY DOMAIN DIFFERENTIATION

Frequency domain differentiation is done by **multiplying each sample** of the frequency spectrum ($X_i(2\pi f_i)$) by the sample frequency ($j2\pi f_i$). Time domain waveforms are differentiation by using the following **equivalent frequency domain operation**.

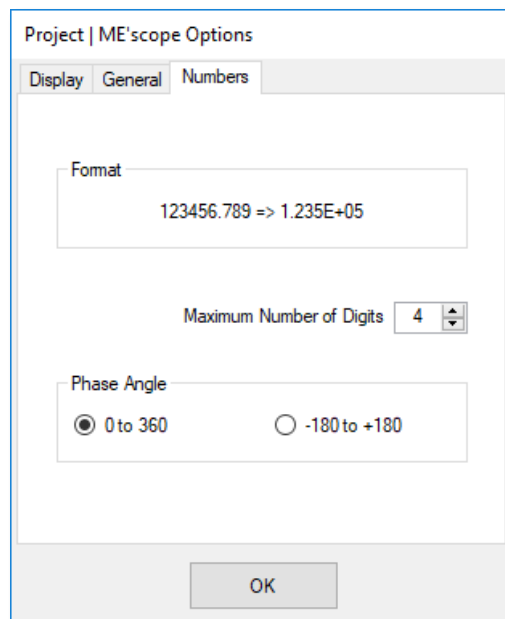
$$\frac{d(x(t))}{dt} \Leftrightarrow (j2\pi f_i)(X_i(2\pi f_i)), i=1,\dots,N/2$$

PERIODIC SIGNAL

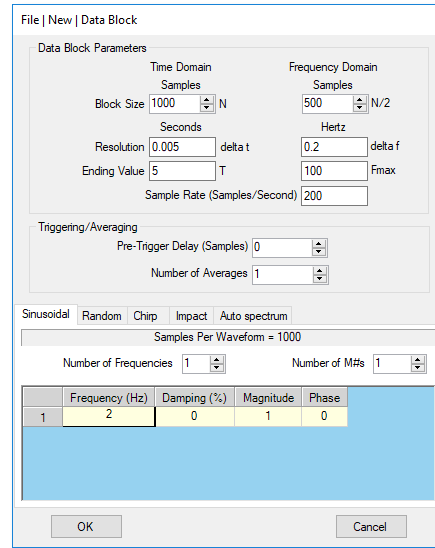
If a time domain waveform is **periodic in its sampling window**, we will see that it can be accurately integrated & differentiated. To demonstrate this, we will synthesize a sine wave that is periodic in its window.

To ensure that numbers are displayed with enough digits in them, MScope should be changed to display numbers with at least 4 digits.

- Execute **File | New | MScope Options** in the MScope window
- Enter **“4”** or greater into the **Maximum Number of Digits** box on the **Numbers** tab in the Options box

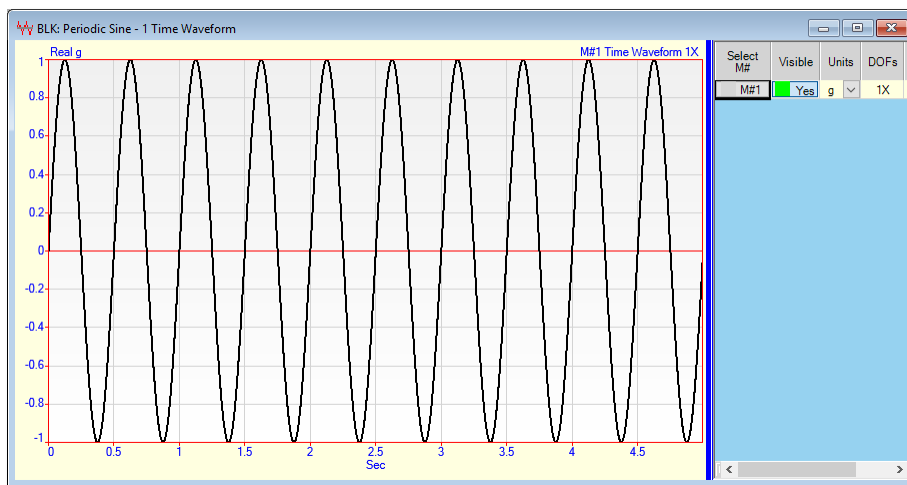


- Execute **File | New | Data Block** in the MScope window
- Enter the following parameters into the dialog box that opens
 - Time Domain Block Size: 1000 Samples**
 - Fmax: 100 Hertz**
- On the **Sinusoidal** tab, enter;
 - Number of Frequencies: 1**
 - Number of M#s: 1**
 - Frequency (Hz): 2.0, Damping (%): 0**
 - Magnitude: 1, Phase: 0**
- **Click** on **OK**, and name the new Data Block file **“Periodic Sine”**.



File | New | Data Block Dialog Box for Periodic Sine.

The new Data Block widow will open with a sine wave in it.



Periodic Acceleration Sine Wave.

Notice that **M#1** has **exactly 10 cycles** of a sine wave in the window. Since an **integer number of cycles** have been sampled in the **BLK: Periodic Sine** window, this signal is **periodic in its sampling window**,

Notice also (**on the upper left**) that the sine wave units are “g” (or gravitational units). The **Tools | Integrate** command in MEscope can be used to convert from acceleration units to velocity units, and from velocity units to displacement units by integrating either time or frequency waveforms.

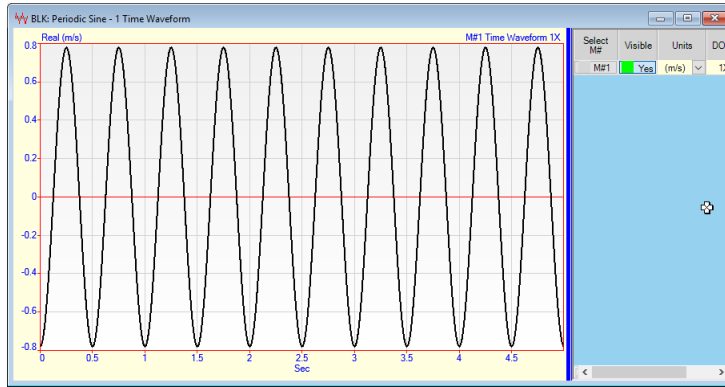
NOTE: g units are automatically converted to **(meters per second-squared)** units by the MEscope integration commands in a Data Block or Shape Table.

Time Domain Integration of a Periodic Signal

To integrate the **periodic** acceleration sine wave,

- Execute **Tools | Integrate** in the **BLK: Periodic Sine** window

After the integration is complete, the acceleration signal will be replaced with the velocity signal shown below. Notice that the units of the signal (**on the upper left**) are **meters/second (m/s)**.



Velocity Sine Wave after Integration.

DC REMOVAL

In general, **integration greatly amplifies the low frequencies** in a waveform, including DC (zero frequency). Dividing each frequency sample of a waveform's spectrum by frequency is the same as multiplying it by the function $(1/\text{frequency})$.

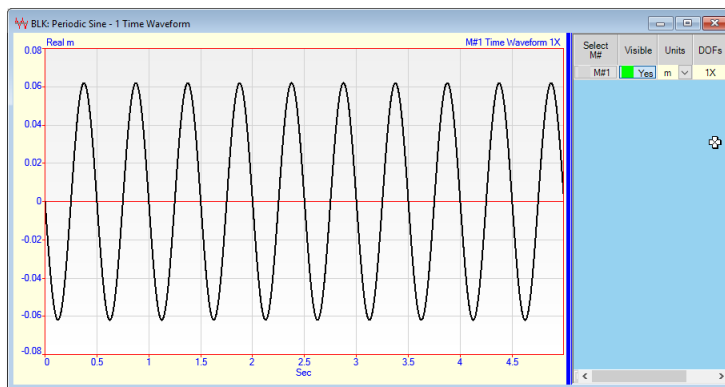
Double integration multiplies a spectrum by $(1/\text{frequency}^2)$.

Most real world signals have **some amount** of DC offset in them, even when DC coupling is used to remove DC from a signal during acquisition. Even a small amount of DC will **dominate** the result, especially when double integration is performed to convert acceleration to displacement signals.

NOTE: When integration is performed on a time waveform in MScope, the DC term is removed from its spectrum before it is divided by frequency.

- Execute **Tools | Integrate** again in the **BLK: Periodic Sine** window

The resulting displacement sine wave, in **meters (m)**, is shown below.



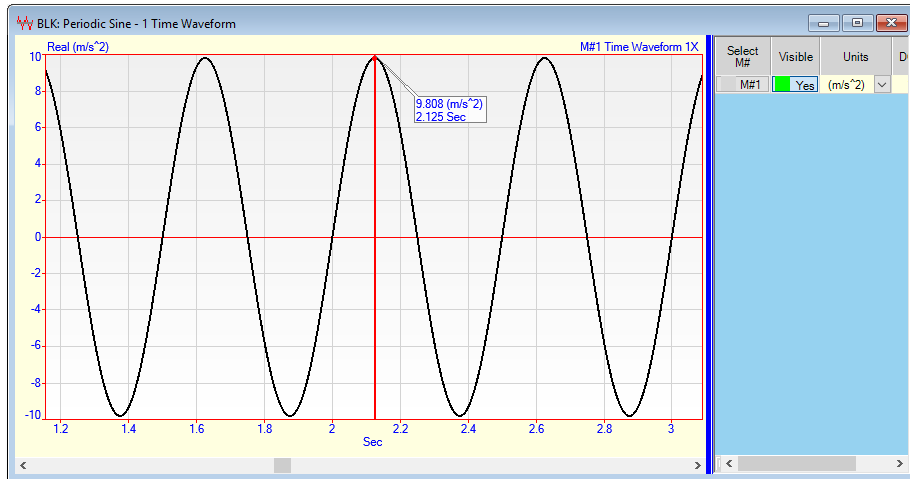
Displacement Sine Wave after Double Integration.

Time Domain Differentiation of a Periodic Signal

Now, let's doubly differentiate the displacement sine wave to recover the original acceleration signal.

- Execute **Tools | Differentiate twice** in the **BLK: Periodic Sine** window

The resulting acceleration signal is shown below. Notice that its values range between $\pm 9.8 \text{ m/s}^2$.



Periodic Sine Wave after Double Differentiation.

Changing Units

To change units from m/s^2 to g 's,

- **Drag** the **Vertical Blue Bar** in the Data Block window to the left, to expose the **M#s spreadsheet**
- **Double click** on the **Units** column heading
- Select "**g**" from the drop down list in the dialog box, and **click** on **Yes** to re-scale the data

The original acceleration signal, with values in the range $\pm 1 g$, should be displayed.

Frequency Domain Integration of a Periodic Signal

So far, we have seen that time domain integration & differentiation can be performed repeatedly on a time domain signal if it is **periodic** in its sampling window.

To doubly integrate the same **periodic** sine wave in the frequency domain

- Execute **Transform | FFT** in the **BLK: Periodic Sine** window
- Execute **Tools | Integrate twice**
- Execute **Transform | Inverse FFT** to obtain the displacement sine wave

Again, the signal has the same values in **meters** as the result from integration of the time domain sine wave.

Frequency Domain Differentiation of a Periodic Signal

To recover the original signal,

1. Transform the signal to the frequency domain
2. Doubly differentiate it
3. Transform it back to the time domain
4. Change the units from m/s^2 to g 's

So far, we have seen that both time and frequency domain integration & differentiation yield the same result when applied to a sine wave that is **periodic in its sampling window**.

NON-PERIODIC SIGNALS

To create a sine wave that is **non-periodic** in its sampling window,

- Execute **File | New | Data Block** in the MEscape window. The following dialog box will open.

File | New | Data Block

Data Block Parameters

Time Domain		Frequency Domain	
Samples	Seconds	Samples	Hertz
Block Size	1000 N	500 N/2	
Resolution	0.005 delta t	0.2 delta f	
Ending Value	5 T	100 Fmax	
Sample Rate (Samples/Second)		200	

Triggering/Averaging

Pre-Trigger Delay (Samples) 0

Number of Averages 1

Sinusoidal Random Chirp Impact Auto spectrum

Samples Per Waveform = 1000

Number of Frequencies 1 Number of M#s 1

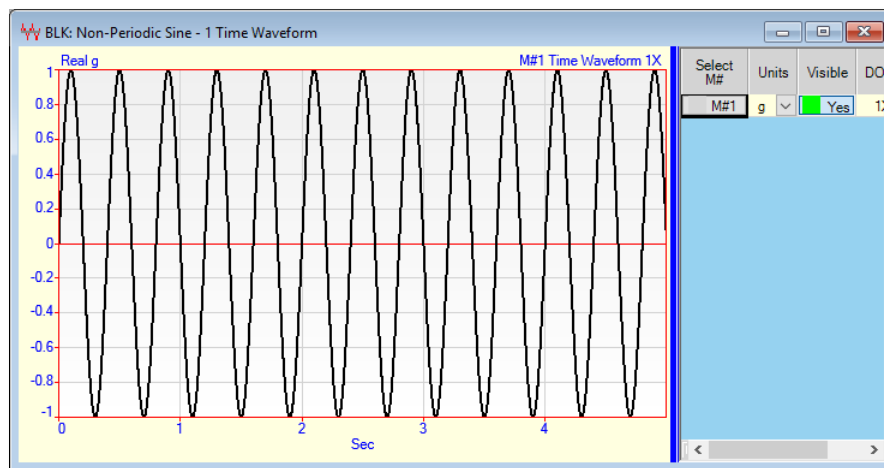
	Frequency (Hz)	Damping (%)	Magnitude	Phase
1	2.5	0	1	0

OK Cancel

Dialog Box to Synthesize a Non-Periodic Sine Wave.

- Enter **Frequency (Hz) = 2.5** into the spreadsheet on the **Sinusoidal** tab, and **click** on **OK**
- **Click** on **OK**, and name the new Data Block file "**Non-Periodic Sine**".

A **2.5 Hz** sine wave with a magnitude of **1.0**, and no damping will be synthesized, and a new Data Block window will open with the sine wave in it, as shown below. Notice that **M#1** has **12.5 cycles** in the window. Since there is **not an integer number of cycles** in the **BLK: Non-Periodic Sine** window, this signal is **non-periodic in its sampling window**.



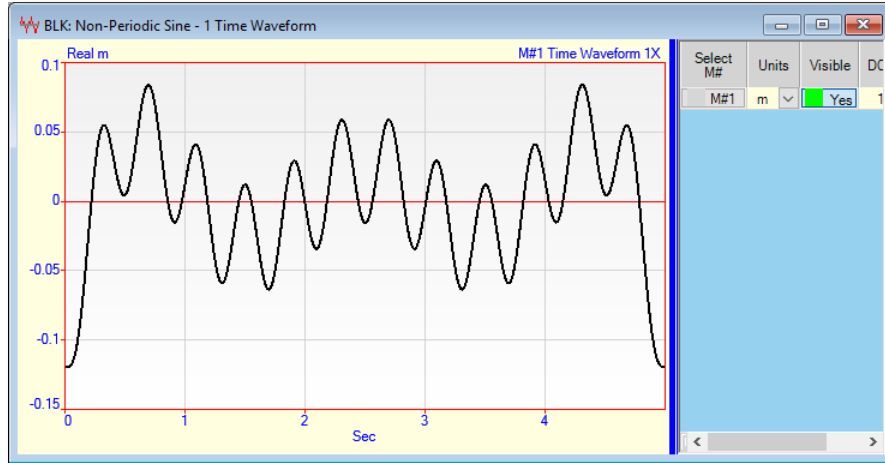
Sine Wave that is Non-Periodic in its Sampling Window.

Time Domain Integration of a Non-Periodic Signal

To doubly integrate this waveform,

- Execute **File | Save Data Block** in the **BLK: Non-Periodic Sine** window
- Execute **Tools | Integrate twice** in the **BLK: Non-Periodic Sine** window

The resulting sinusoidal displacement waveform is shown below.



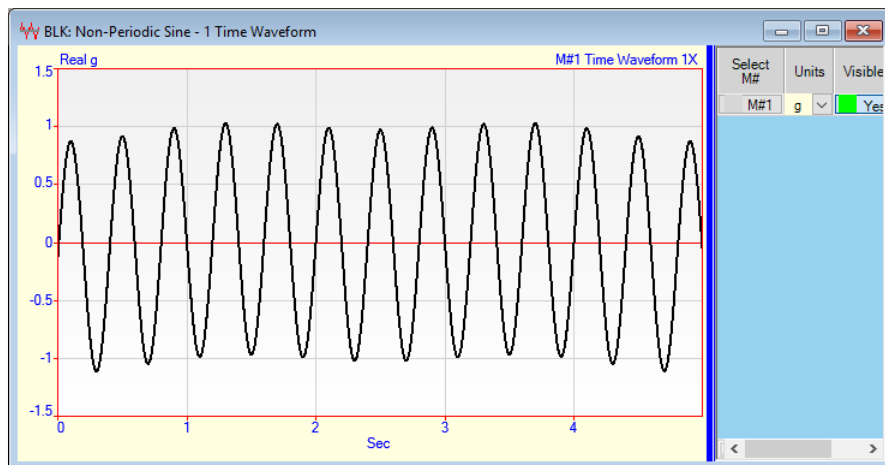
Double Integration of a Non-Periodic Sine Wave.

Doubly integrating the **non-periodic** sine wave gives an unexpected result which is no longer a pure sinusoidal waveform. This error is due to **leakage**, an FFT phenomenon that occurs when **non-periodic** signals are transformed from one domain to the other. (see **App Note #1** for details)

Time Domain Differentiation of Non-Periodic Signal

We will now try to recover the original **non-periodic** sine wave by **doubly differentiating** it and changing its units to **g's**.

- Execute **Tools | Differentiate twice** in the **BLK: Non-Periodic Sine** window
- **Double click** on the **Units** column heading in the **M#s spreadsheet**
- Select **“g”** from the drop down list in the dialog box that opens, and **click** on **Yes** to re-scale the data



Recovered Non-Periodic Sine Wave.

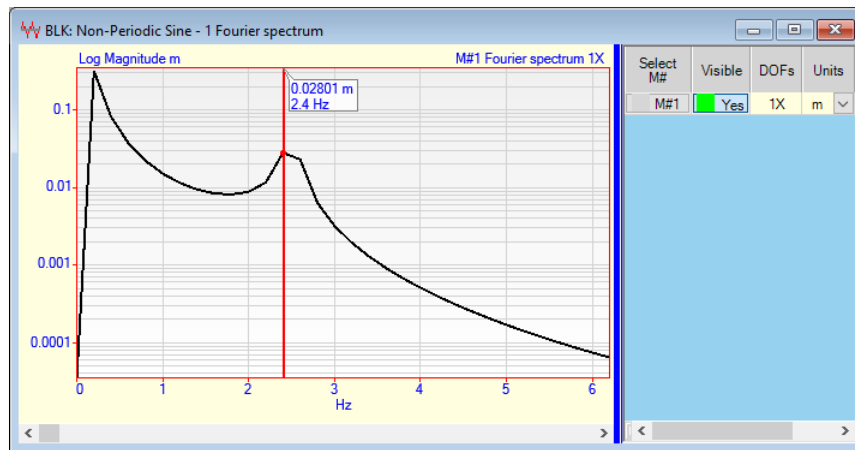
The **non-periodic** acceleration sine wave (in g's) is shown above. The original waveform was **not perfectly recovered** because a small portion of the leakage in the spectrum (near DC) was removed.

Frequency Domain Integration of a Non-Periodic Signal

Now, let's doubly integrate the **non-periodic** acceleration signal using frequency domain integration, and remove even more of the leakage near DC to see if that improves the integration.

- Close the **BLK: Non-Periodic Sine** window and don't save it
- Open **BLK: Non-Periodic Sine** from the Project panel


- Execute **Transform | FFT** in the **BLK: Non-Periodic Sine** window
- Execute **Tools | Integrate twice**
- **Zoom** the display **around 2.5 Hz**, as shown below

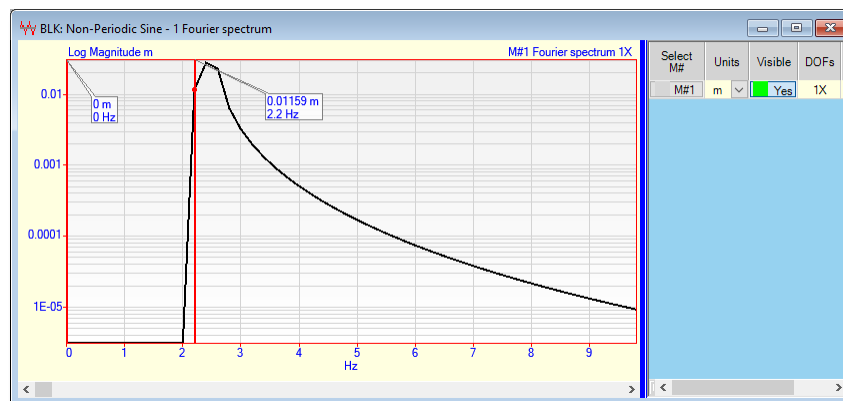


Frequency Domain Double Integration of a Non-Periodic Sine Wave.

Notice that instead of the expected single frequency peak at 2.5 Hz in the spectrum that the signal **has leaked from the 2.5 Hz peak** to all of the surrounding frequencies in the spectrum. In addition, the double integration has **significantly increased the frequencies below 2.5 Hz**

Again, since the signal was **non-periodic** in its sampling window, leakage occurred in its spectrum, and the low frequencies, (which were distorted by leakage), were **amplified by the double integration process**.

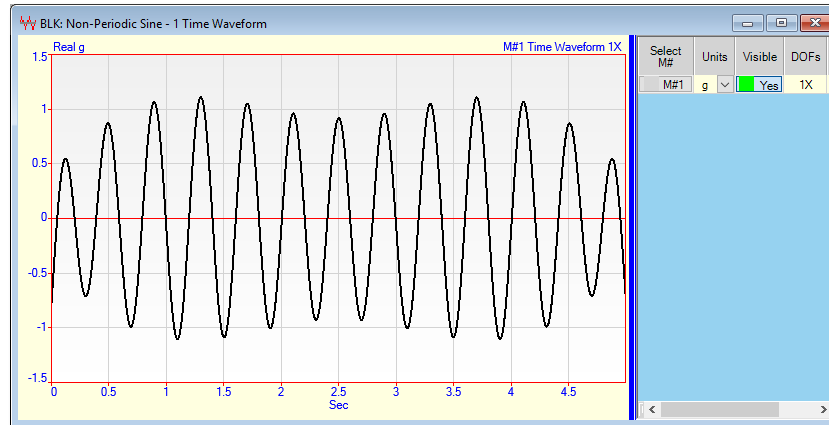
- Display the Band Cursor , and position it as shown below
- Execute **Transform | Window M#s**
- Select **Notch** in the dialog box that opens, and click on **Apply**



Frequency Spectrum After Notch Window Applied.

- Execute **Transform | Inverse FFT** in the **BLK: Non-Periodic Sine** window
- Execute **Tools | Differentiate twice**

The result is shown below. Integration will always amplify the lower frequencies in a spectrum. Removal of some of the low frequencies by applying a notch window will improve the integrated signals, but differentiating them will not restore the original waveform. Leakage combined with notch windowing will change the spectrum.



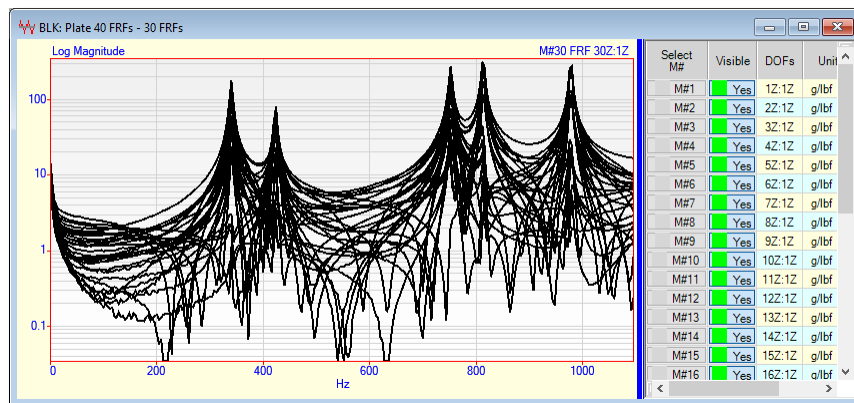
Non-Periodic Sine After Double Differentiation.

Removal of Low Frequencies before Integration

When a time domain signal is **non-periodic** in its sampling window or a frequency domain signal has non-zero low frequency components at or near DC those components **must be removed from its spectrum** before it can be integrated using frequency domain integration.

- Execute **File | Import | Data Block**, navigate to **MEscope\Examples\Plate 30 FRFs.UFF**, and import the file
- Execute **Format | Overlaid**
- Save the **BLK: Plate 30 FRFs** file

The overlaid FRFs are shown below.



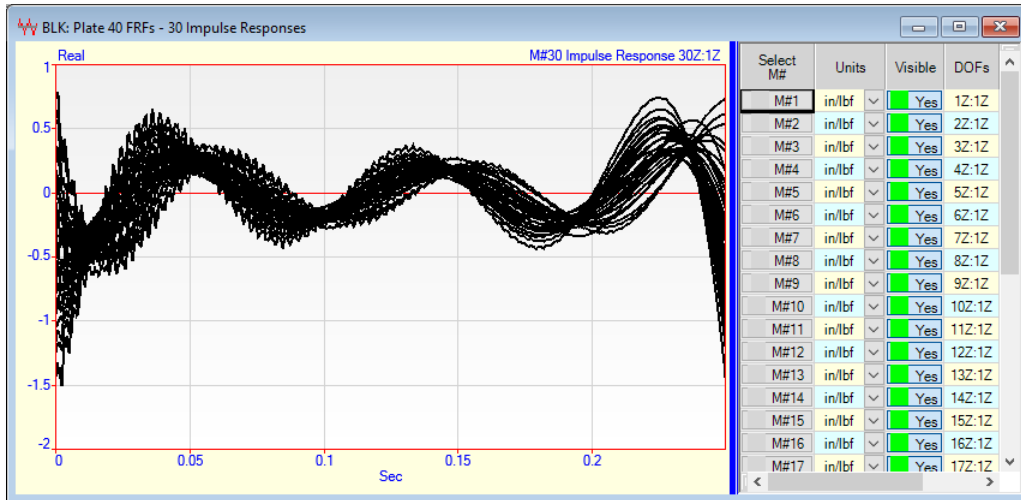
Overlaid FRFs Showing DC & Other Low Frequencies.

Notice that all of the FRFs have **non-zero DC** values plus many other low frequency samples with non-zero values. This low frequency response is the **rigid body motion** of the plate on its soft mountings. The plate was impact-tested while resting on a form rubber pad.

Is these measurements were doubly integrated to obtain displacement responses, the low frequencies would **dominate** the FRFs and their corresponding time domain IRFs (Impulse Response Functions).

- Execute **Transform | Inverse FFT**
- Execute **Tools | Integrate twice**


Notice in the figure below that even though the the DC samples were removed the other low frequencies were still greatly amplified by the double integration.



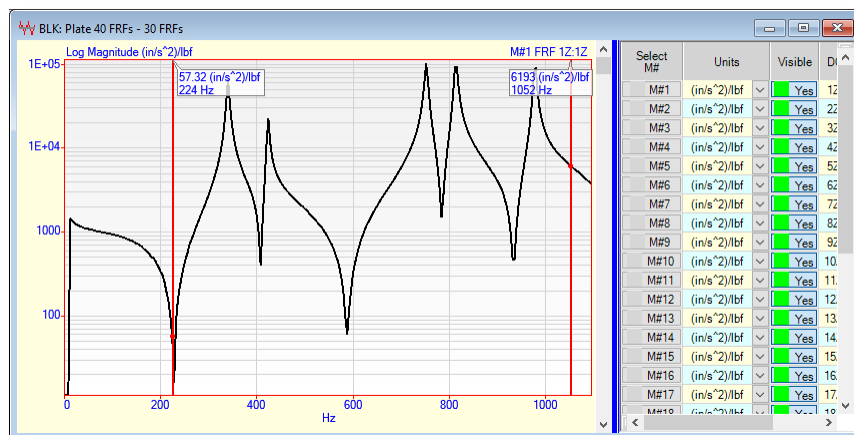
Distorted IRFs Due to Double Integration.

Clearly, this is not the expected result! The IRFs above do not even resemble impulse responses. More of the low frequencies must be removed before integrating the FRFs.

Removing Lower Frequencies

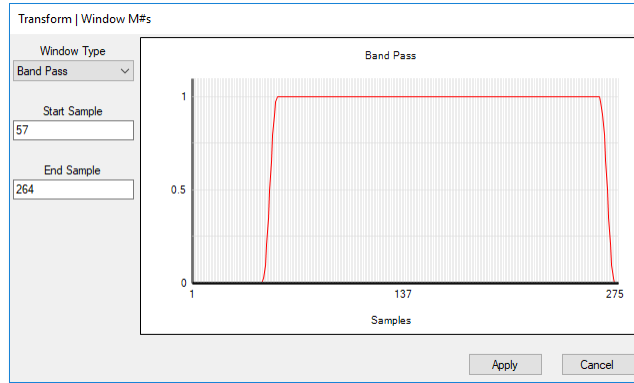
- Close the **BLK: Plate 30 FRFs.UFF** window, and then re-open it
- Display the Band Cursor .
- **Drag** the band cursors to enclose a band of **about 224 to 1052 Hz**, as shown below.

When the **Band Pass** window is applied to the FRFs, all of the data **outside of the cursor band** will be removed, including DC.

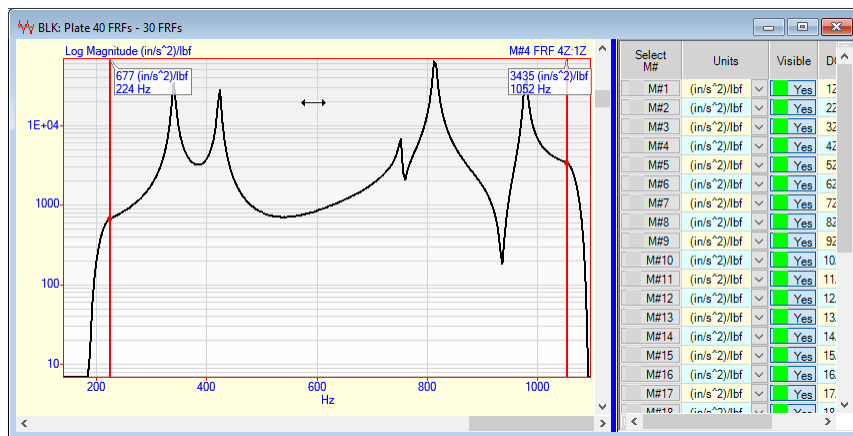


Band Cursors for Band Passing the FRFs

- Execute **Transform | Window M#s**
- Select **Band Pass** from the **Window Type** list, as shown below
- **Click** on the **Apply** button



After the Band Pass window has been applied, the FRFs will be smoothly tapered to zero outside of the Band cursor edges, as shown below.

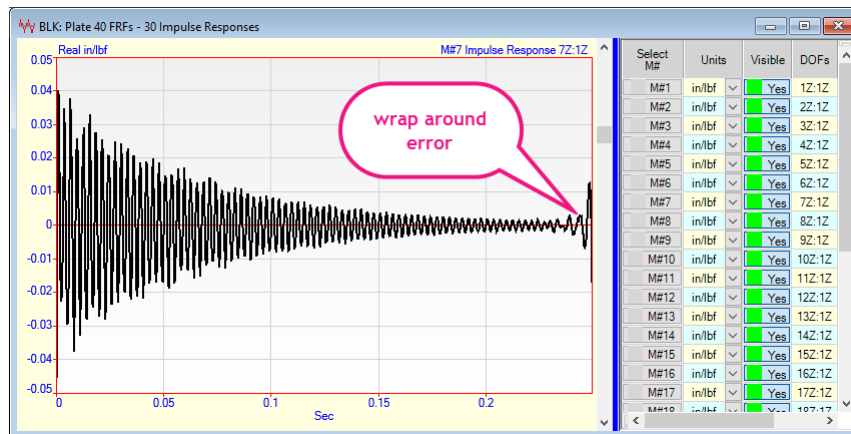


FRFs after Band Pass Windowing.

Now the FRFs are ready for double integration.

- Execute **Transform | Inverse FFT**
- Execute **Tools | Integrate twice**

The resulting M#s are IRFs with displacement response units. Notice that they also *look like* the expected impulse response functions, with one exception.



Displacement Response IRFs.

WRAP AROUND ERROR

Even though DC and the low frequency **rigid body motion** has been removed prior to double integration, the resulting IRFs still exhibit a problem, called **time domain leakage** or **wrap around error**.

All of the IRFs exhibit the characteristic *damped sinusoidal* response, but many of them begin to **grow in amplitude** near the end of the sampling window. This is not realistic, since real vibration does not damp out and then grow again. This too is a signal processing error caused by the FFT. Just as leakage was created in its spectrum when a time domain signal was **non-periodic (or was not completely contained)** in its sampling window, the same *leakage error occurs when a frequency spectrum is not completely contained in its sampling window*.

This time domain error is either called **time domain leakage or wrap around error**. When an FRF is calculated, if resonances outside of its frequency span are excited, the FRF will be **truncated** in frequency just as if the **true FRF** (defined over a larger frequency span) were multiplied by a rectangular sampling window. The result is a time domain **smeared** signal after the Inverse FFT is applied to the FRF. In this case, some of the IRFs exhibited **pronounced wrap around error** at the end of their time signal **due to their truncated FRFs**.

CONCLUSIONS

First, we saw that a sine wave that was **periodic in its sampling window** could be integrated & differentiated using time or frequency domain methods, and the same result was obtained.

Next, we saw the effects of *leakage* errors when a sine wave of **slightly different frequency** became **non-periodic in its sampling window** and could not be integrated & differentiated without some windowing to reduce the leakage effects.

Finally, we looked at the integration of FRFs that had DC and low frequency **rigid body** dynamics in them. We saw that applying a Band Pass window to the FRFs effectively removed the low frequency components as well as high frequency residual components, thus permitting double integration of the FRFs which yielded realistic IRFs.

From these examples, we can make the following conclusions,

- Integration & differentiation can be used on any time domain or frequency domain signal if it is **periodic** (or is completely contained) within its sampling window
- If a signal is **non-periodic (or is not completely contained)** within its sampling window, **leakage effects will occur** when the signal is transformed to the other domain
- Before integrating a **non-periodic** signal, **DC and other low frequency components**, as well as the **residual effects of high frequency components**, should be removed from its frequency spectrum by Band Pass windowing the data
- Leakage usually distorts a waveform in the opposite domain enough so that it is unusable unless a special window is applied to it before applying the FFT