DATA TRANSLATION®

Using MATLAB for Vibration Measurements

Vibration measurements are critical in predictive maintenance and diagnostic fault testing applications for quality assurance. This white paper describes the following topics that are important when choosing a vibration measurement solution:

- Sensor Considerations
- Data Acquisition Hardware Considerations

  Sigma-Delta A/D Converters

  Simultaneous Inputs

  AC Coupling
- Data Acquisition Software Considerations

  Configuring your External Hardware

  Acquiring Data from two Triaxial Accelerometers

  Reading Data into MATLAB for Immediate Analysis

  Generating Reports and Applications

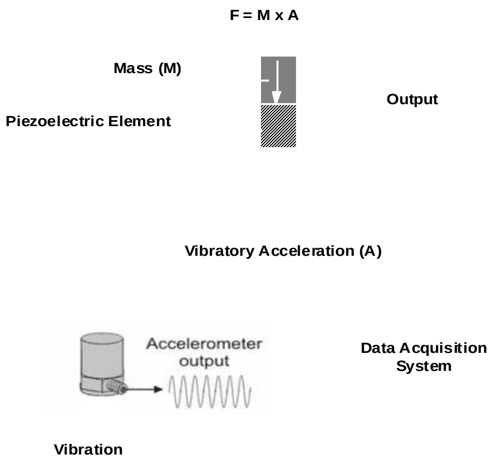
Sensor Considerations

Typical vibration applications use IEPE (ICPTM) accelerometers to measure vibration. An accelerometer consists of a piezoelectric element connected to a known mass. When the accelerometer is vibrated, the mass applies force to the piezoelectric element, generating an electrical charge that is proportional to the applied force. You can then measure this charge to determine vibration characteristics.

A wide range of IEPE accelerometers are available, including those that measure along only one axis and those that measure up to three axes simultaneously, called triaxial accelerometers. Different mounting options and measurement ranges are available to suit your application.

When choosing a vibration measurement solution, ensure that your data acquisition system supports direct connection of IEPE inputs, and that the measurement range of your sensor is within the input range of your data acquisition system. For example, if you want to measure a 6 V input signal, the full-scale input range of your system should exceed 6 V. In addition, if you are using a triaxial accelerometer, ensure that your data acquisition supports simultaneous acquisition of up to three IEPE inputs so that you can correlate measurements on all three axes.

Most accelerometers require a current source of 4 mA and a compliance voltage of at least 18 V to drive their internal circuitry. Other accelerometers require a 2 mA current source, but have limitations in cable length and bandwidth. Ensure that your data acquisition system supports the requirements of your sensor.



How Accelerometers Work

Data Acquisition Hardware Considerations

When choosing data acquisition hardware for vibration applications, the following hardware considerations are important:

- Sigma-Delta A/Ds
- Simultaneous A/Ds
- AC Coupling

Sigma-Delta A/Ds

Most vibration measurement applications require a data acquisition system with an antialiasing filter in the A/D circuitry to eliminate unwanted frequencies in the measurements. Sigma-Delta A/Ds have anti-aliasing filters built-in. Sigma-Delta converters offer the following advantages, making them ideal for vibration measurement applications:

• Reduce noise and improve accuracy by oversampling each input.

• Eliminate errors that result from aliasing and high frequency noise by using a built-in decimation filter.

• Provide excellent low-level signal-to noise performance, which improves

dynamic accuracy on low-level signals.

• Provide excellent differential linearity, which ensures consistently accurate data conversion across the full input range of the signal.

For precise measurements, 24-bit Sigma-Delta A/Ds are also desirable since they allow you to measure the full dynamic range of the input sensors.

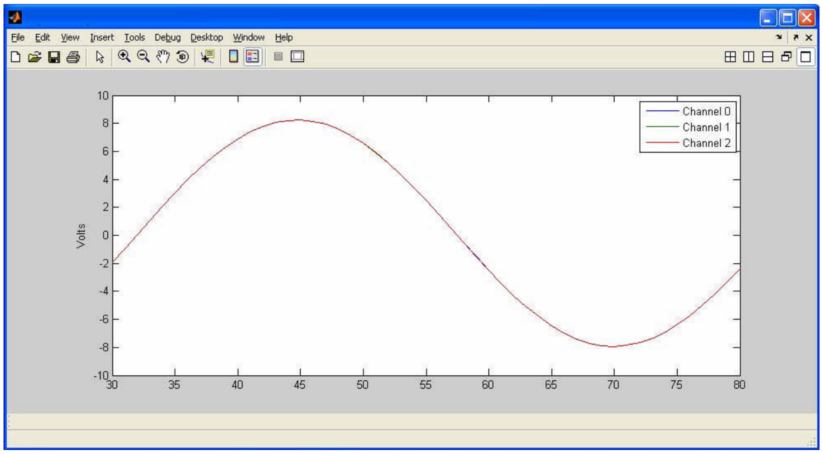
Simultaneous A/Ds

Simultaneous sampling increases the bandwidth of signals that you can accurately measure, while eliminating several sources of error, including time skew and cross-talk between channels.

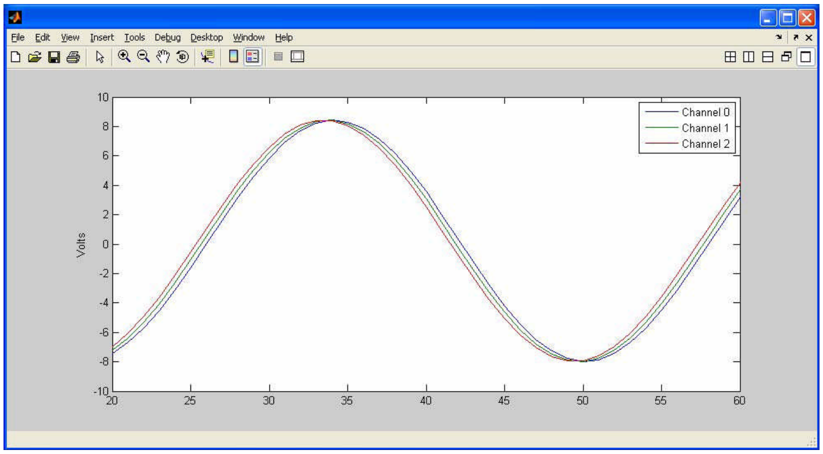
One A/D converter per channel increases the module's overall sampling frequency (when compared to multiplexed architectures) and increases the signal bandwidth that you can acquire. For example, if your module provides a sampling frequency of 100 kHz per channel, then according to Nyquist, you can accurately measure up to 50 kHz on each channel when using a simultaneous A/D architecture. In contrast, in multiplexed architectures, the usable signal bandwidth decreases with every additional channel you measure.

With a simultaneous A/D architecture, all signal inputs are sampled at the exact same instant in time, therefore, any channel skew, cross-talk, and settling time problems are also virtually eliminated, allowing you to measure highly accurate data at high speeds.

Simultaneous Sampling eliminates time skew between channels and simplifies both time and frequency based analysis techniques.



Sequentially (Multiplexed) Sampling may require software correction for detecting certain patterns

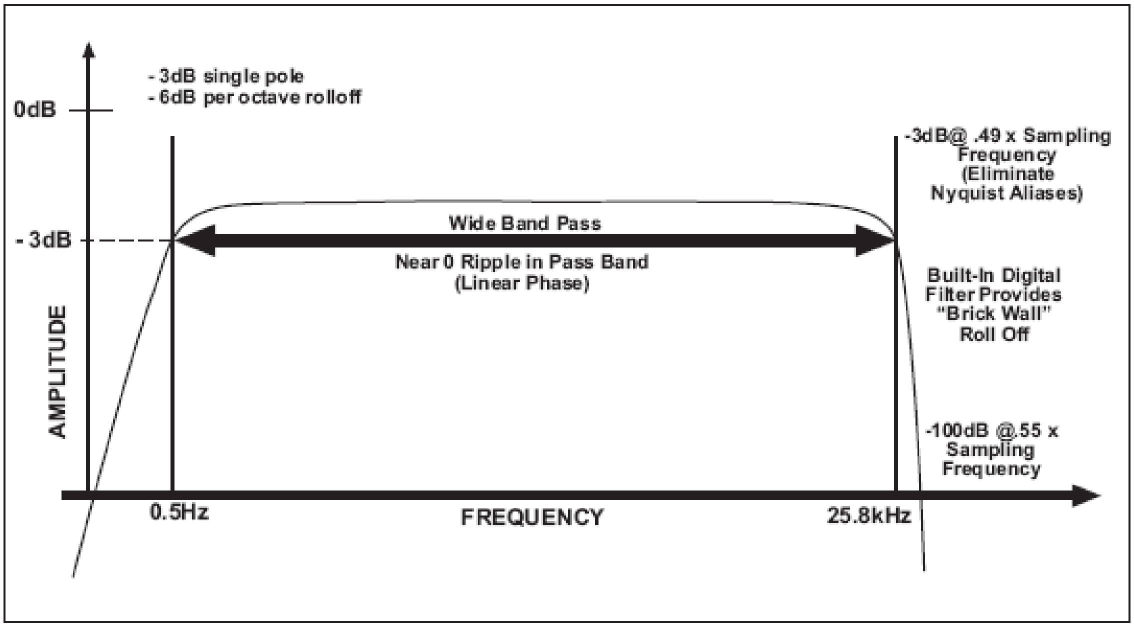


The DT9837 and DT9841-VIB USB module from Data Translation® provide up to 8, 24-bit Sigma-Delta converters for precise, simultaneous analog input measurements.

AC Coupling

To measure low frequency signals accurately at the Nyquist sampling rate, it is important that your anti-aliasing filter supports a wide passband to eliminate unwanted high frequency components.

An ideal anti-aliasing filter passes all signals in the band of interest and blocks all signals outside of that band. However, in practical use, the roll-off characteristics of the antialiasing filter allow some signals to pass above the filter's cutoff frequency. By using AC coupling, you can eliminate any DC that may pass through the anti-aliasing filter, thereby, maintaining the integrity of your signal. As shown below, the low frequency-breakpoint of 0.5 Hz allows very low frequency measurements.


- 3dB single pole
- 6dB per octave rolloff
0dB
-3dB@ .49 x Sampling Frequency (Eliminate Nyquist Aliases)
Wide Band Pass
- 3dB
Near 0 Ripple in Pass Band (Linear Phase)
Built-In Digital Filter Provides "Brick Wall" Roll Off
AMPLITUDE
-100dB @.55 x Sampling Frequency
0.5Hz
FREQUENCY
25.8kHz

Data Translation's DT9837 and DT9841-VIB modules provide software- selectable AC and DC coupling. When AC coupling is selected, the modules eliminate any DC that may pass through the anti-aliasing filter. The excellent "brick wall" anti-alias filter eliminates unwanted high frequency interference.

Data Acquisition Software Considerations

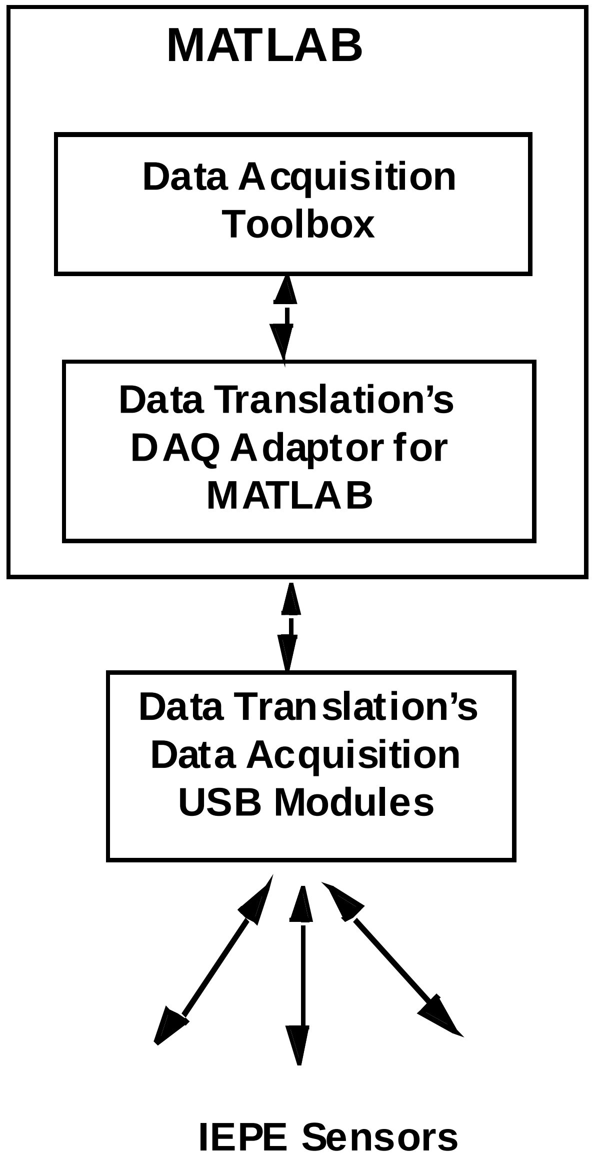
When choosing software for data acquisition and analysis, the following considerations are important:

- Support for a wide array of data acquisition hardware
- Strong data analysis libraries
- Scripting that is easy to read, modify, and debug
- Application development support
- Report generation to share results

MATLAB® is a well known interactive software environment for data acquisition and analysis, report generation, and test system development. MATLAB provides a complete set of tools for acquiring and analyzing analog and digital I/O signals from a variety of PC-compatible data acquisition hardware. The MATLAB Data Acquisition Toolbox lets you configure your external hardware devices, read data into MATLAB and Simulink for immediate analysis, and send out data for controlling your system.

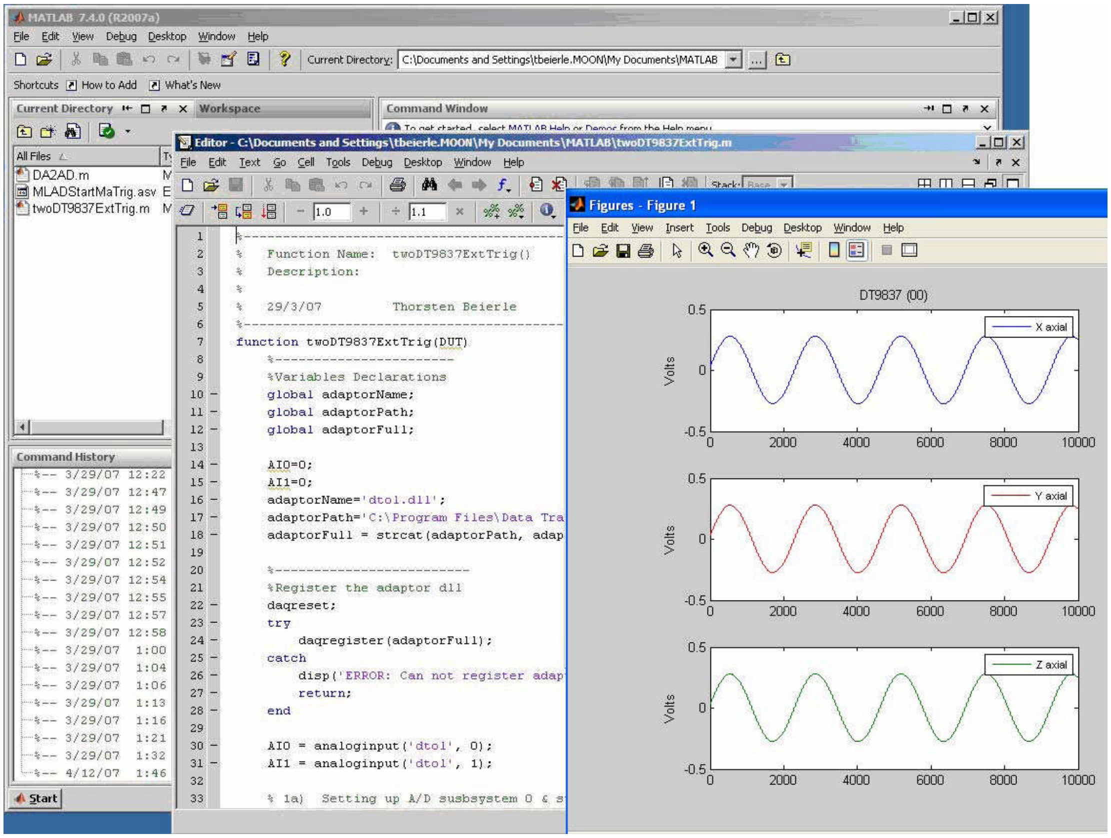
The diagram below depicts an example using MATLAB and the MATLAB Data Acquisition Toolbox with Data Translation's DT9837 to acquire vibration data from USB modules. Notice that the Data Translation provides an interface layer, called the DAQ

Adaptor for MATLAB, which allows the MATLAB Data Acquisition Toolbox to communicate with Data Translation's hardware. While the Data Acquisition Toolbox is collecting data, MATLAB can analyze and visualize the data.


MATLAB
Data Acquisition Toolbox
Data Translation's DAQ Adaptor for MATLAB
Data Translation's Data Acquisition USB Modules
IEPE Sensors

While the Data Acquisition Toolbox is collecting vibration data from the USB module, MATLAB can analyze and visualize the data.

The following sections describe in detail how you can use MATLAB to measure vibration data from two triaxial accelerometers using two Data Translation DT9837 USB data acquisition modules, and display the results.



MATLAB 7.4.0 (R2007a)
File Edit View Debug Desktop Window Help
Current Directory: C:\Documents and Settings\tbeierle.MOON\My Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Command Window
All Files
DA2AD.m
MLADStartMaTrig.asv
twoDT9837ExtTrig.m
Command History
Start
Editor - C:\Documents and Settings\tbeierle.MOON\My Documents\MATLAB\twoDT9837ExtTrig.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Figures - Figure 1
File Edit View Insert Tools Debug Desktop Window Help
DT9837 (00)
X axial
Y axial
Z axial
Volts
0.5
0
-0.5
0
2000
4000
6000
8000
10000

```
%-- 3/29/07 12:22
%-- 3/29/07 12:47
%-- 3/29/07 12:49
%-- 3/29/07 12:50
%-- 3/29/07 12:51
%-- 3/29/07 12:52
%-- 3/29/07 12:54
%-- 3/29/07 12:55
%-- 3/29/07 12:57
%-- 3/29/07 12:58
%-- 3/29/07  1:00
%-- 3/29/07  1:04
%-- 3/29/07  1:06
%-- 3/29/07  1:13
%-- 3/29/07  1:16
%-- 3/29/07  1:21
%-- 3/29/07  1:32
%-- 4/12/07  1:46
```

```
%---------------------------------------------
%   Function Name:  twoDT9837ExtTrig()
%   Description:
%
%   29/3/07         Thorsten Beierle
%---------------------------------------------
function twoDT9837ExtTrig(DUT)
    %----------------------
    %Variables Declarations
    global adaptorName;
    global adaptorPath;
    global adaptorFull;

    AI0=0;
    AI1=0;
    adaptorName='dtol.dll';
    adaptorPath='C:\Program Files\Data Tra
    adaptorFull = strcat(adaptorPath, adap

    %----------------------
    %Register the adaptor dll
    daqreset;
    try
        daqregister(adaptorFull);
    catch
        disp('ERROR: Can not register adap
        return;
    end

    AI0 = analoginput('dtol', 0);
    AI1 = analoginput('dtol', 1);

    % 1a)  Setting up A/D susbsystem 0 & s
```

This example uses the MATLAB Data Acquisition Toolbox with Data Translation's USB modules to measure and analyze vibration data in one integrated environment.

Configuring Your External Hardware

To configure your external hardware, first connect the X, Y, and Z outputs of one accelerometer to analog input channels 0, 1, and 2 of the first DT9837 module. Then, connect the X, Y, and Z outputs of the second accelerometer to analog input channels 0, 1, and 2 of the second DT9837 module. If you want to start acquisition on both modules simultaneously, connect a shared external digital TTL trigger input to both modules. A rising-edge on the external trigger input will start the acquisition on both modules.

In MATLAB, set up your script as follows to finish configuring your system:

```
% Register Data Translation’s DAQ Adaptor for MATLAB
daqregister (‘dtol’);
% Create an analog input object to communicate with each
% DT9837 module (AI0 corresponds to the first module; AI1
% corresponds to the second module)
AI0 = analoginput(‘dtol’, 0);
AI1 = analoginput(‘dtol’, 1);
% Add A/D channels 0, 1, and 2 – one channel for each axis of the accelerometer –
% to each analog input object
addchannel(AI0, 0:2);
addchannel(AI1, 0:2);
% For each DT9837 module, configure A/D channels 0, 1, and 2 for IEPE inputs;
% each channel uses a 4 mA current source and AC coupling
AI0.channel.ExcitationCurrentSource = ‘Internal’;
AI0.channel.Coupling = ‘AC’;
AI1.channel.ExcitationCurrentSource = ‘Internal’;
AI1.channel.Coupling = ‘AC’;
% Configure the external digital trigger (rising edge) for each DT9837 module
set([AI0 AI1], ‘TriggerType’, ‘HwDigital’);
set([AI0 AI1], ‘TriggerCondition’, ‘Rising’);
% When a trigger is detected, continuously acquire data until a stop function is
% issued or until an error occurs
set([AI0 AI1], ‘SamplesPerTrigger’, inf);
% Configure the clock for the each DT9837 module, setting the clock rate to the
% maximum clock rate supported
maxRate = daqhwinfo(AI0, MaxSampleRate);
set([AI0 AI1], ‘SampleRate’, maxRate);
% Allocate memory to store the data
```

```
set([AI0 AI1], ‘BufferingConfig’, [1000 5]);
% Define the number of samples (10000, in this case) to acquire for
each channel
set([AI0 AI1], ‘SamplesAcquiredFcnCount’, 10000);
% When 10000 samples have been acquired for each channel,
% define the function to call to plot the data
set(AI0, ‘SamplesAcquiredFcn’, {@plotData0});
set(AI1, ‘SamplesAcquiredFcn’, {@plotData1});
```

Acquiring Data from Two Triaxial Accelerometers

Once you have wired your external hardware and configured it in MATLAB, start the analog input objects using the following script in MATLAB:

```
% Start the analog input objects on both modules
start([AI0 AI1]);
```

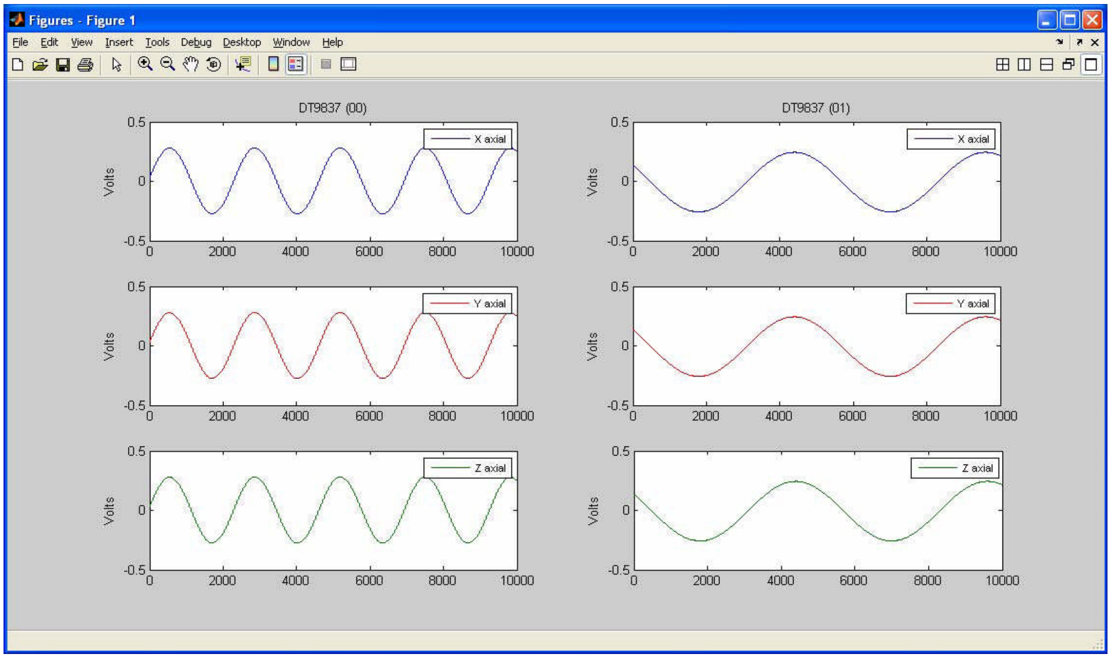
When a rising edge of the shared digital trigger is detected on each module, acquisition begins. The data is acquired from all six IEPE inputs at the maximum clock rate of the device, and the data is stored in the memory you allocated.

Reading Data into MATLAB for Immediate Analysis

When 10,000 samples are acquired on each input channel, the callback functions you referenced previously (plotData0 and plotData1) are called. These functions read the data into MATLAB and plot it for immediate analysis.

Callback functions plotData0 and plotData1 are defined as follows:

```
% Get the data from the first DT9837 module into MATLAB and plot
all
% three channels separately.
function plotData0(Obj, event)
AI0 = Obj;
data = getdata (AI0, 10000);
ch0a_data = data(:,1);
subplot(3,2,1); plot(ch0a_data)
```



Data acquired and plotted in MATLAB for immediate analysis.

Generating Reports and Applications

A report can be automatically generated from the MATLAB script if it is written in the MATLAB Editor. To generate the report, press the *Publish to HTML* button in the editor. (Other formats are available.) Pressing this button will execute the MATLAB script and embed the script's contents along with all plots generated in a single HTML file that can be viewed and shared.

MATLAB provides a GUI development tool called GUIDE that allows you to incorporate the MATLAB script in a graphical application. Press the *GUIDE* button in MATLAB to launch this tool.

Conclusion

When choosing a solution to measure vibration, consider the IEPE sensors you need and their requirements, the capabilities of your USB data acquisition hardware, and the analysis software you'll need to accomplish your goals. This paper described an example that used the MATLAB Data Acquisition Toolbox with Data Translation's DT9837 USB modules and DAQ Adaptor for MATLAB to provide a complete solution for measuring and analyzing sound and vibration.

For More Information

1. Using DT9837 hardware: http://www.datx.com/products/dataacquisition/usb/ prod_dt9837.htm

2. Using MATLAB: http://www.mathworks.com/products/matlab

3. Using MATLAB Data Acquisition Toolbox: http://www.mathworks.com/products/daq

4. Using the DAQ Adaptor for MATLAB: http://www.datatranslation.com/products_software/dt-matlab-interface-tools.htm

Trial of MATLAB and MATLAB Data Acquisition Toolbox:

www.mathworks.com/products/daq/tryit.html

MATLAB Script used in this example:

www.matlabcentral.com