# AC 2012-5316: LOW-COST EDUCATIONAL LASER BASED VIBRATION MEASUREMENT SYSTEM WITH IMPROVED SIGNAL CONDITIONING, PYTHON AND MATLAB

**Dr. Jonathan M. Hill, University of Hartford**

Jonathan Hill is an Associate Professor in electrical and computer engineering at the University of Hartford in Connecticut. He has a Ph.D. and M.S.E.E. from Worcester Polytechnic Inst. in Worcester, Mass., and he was previously a Project Engineer at Digital Equipment Corp. He instructs graduate and undergraduate computer engineering computer courses, directs graduate research, and performs research involving embedded microprocessor based systems. His current projects involve small system design, signal processing, and intelligent instrumentation.

**Dr. Devdas Shetty, University of Hartford**

Devdas Shetty is a professor of mechanical engineering and the Dean of Research at the University of Hartford, Conn. Previously, he held the position as Dean of Engineering at Lawrence Technological University, Michigan, and a faculty position at the Cooper Union for Advancement of Science and Art, N.Y. He is the author of more than 200 papers and three books on mechatronics and product design. Shetty's field of expertise involves mechatronics system design, innovative product design, laser instrumentation, laser material processing, unmanned aerial systems, guided projectiles, rehab system for gait and walking without fall, engineering education, and ABET accreditation.

# Low Cost Educational Laser Based Vibration Measurement System with Improved Signal Conditioning, Python and MATLAB


## Abstract

This project involves a laser based vibration measurement system that has educational value and can be used in a student laboratory. The system must be small, inexpensive, and convenient to use, without extensive programming. The LabJack U3 acquisition system was used with a laptop and a netbook computer. Our use of Python and MATLAB are suitable software choices for this system. The vibration measurement system provides cross-disciplinary educational opportunities with hands-on experimental activities. The acquisition module provides an opportunity to study sampling and the sampling delay between successive channels. The input signal conditioning amplifier provides an opportunity to study simple operational amplifier circuits as well as study instrumentation principles such as the notion of the common-mode and differential-mode with respect to signals.


## Introduction

This project involves a laser based vibration measurement system that has educational value and can be used in a student laboratory. This project continues research reported by Shetty, Kondo, and Noriega[1], which investigated the development of the experimental apparatus used here. The experimental apparatus is used in the ME472 capstone design project as well as the ME505 mechatronic system design course, taken by mechanical engineering students.

One major challenge of any concentration, such as mechatronics is the balance between topics. The focus of our prior research led to a system useful to the mechanical engineering degree program. However, today in industry multi-disciplinary teams are most often found, consisting of professionals involved with hardware, software, electronics, as well as mechanical hardware. As such, our vibration measurement system must be more useful to other degree programs as well as multi-disciplinary teams of students.

This research is based on a new set of design goals, to improve the usefulness of the vibration measurement system. While our new system is compact and is less inexpensive than the prior system, in particular, it is more multi-disciplinary, providing entirely new educational value. The system allows students to more closely study principles involved in sampling and signal conditioning, as well as the opportunity to study data acquisition software, but without involving intensive programming. These goals were achieved with an inexpensive acquisition system along with two software tools, namely Python and MATLAB.

The overall system comprises the experimental apparatus, signal conditioning electronics, a data acquisition module, and a host computer. The experimental apparatus includes a laser, reflective film, and a position sensitive device (PSD), arranged as in Figure 1. A change in position dy of the reflective film causes a change in position dx of the spot, of the incident laser beam. The PSD is like that illustrated in Figure 2. A PSD is a photo-electric device with two output terminals, V1 and V2.

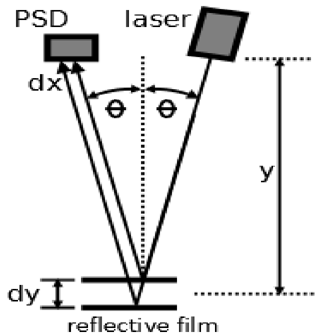

Figure 1: Experimental apparatus

When the laser is incident at the center of the PSD device, the terminals V1 and V2 have the same Voltage relative to ground. Equation (1) is used to determine the position of the spot of light, along the device axis.

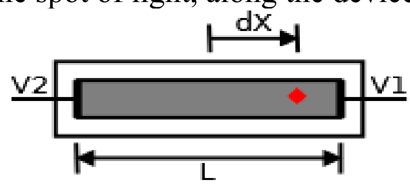

Figure 2: Illustration of PSD

$$dx = \frac{V1 - V2}{V1 + V2} \times \frac{L}{2} \tag{1}$$

The experimental apparatus is general purpose in nature, with suitable adjustments it can be adapted as a component in various projects. The sensitivity and calibration of the experimental apparatus is dependent on the distance y of the reflective film and the angle $\theta$ of the laser beam, as shown in Figure 1. Apart from the experimental apparatus, we implemented our own signal conditioning electronics, a LabJack[5] model U3 data acquisition module was used, and a laptop and a netbook computer each served as the host.

The topics presented in this paper can be used in the classroom as well as with interdisciplinary projects involving electrical, computer, and mechanical engineering students. The intent of a capstone project is for students to demonstrate their design abilities in each of their own corresponding disciplines. Such a design involve real life constraints.

In the following we present our choices with regard to the host computer system, data acquisition and software. Educational opportunities and hands-on activities are presented. Students can examine the notion of sampling and measure the delay between channels. Students can examine and construct signal conditioning electronics and study the nature of the common-mode and differential-mode with regards to signals. A differential and common mode signal conditioning circuit is presented in detail, along with a test scenario. A practical example using a mass and spring apparatus is presented.

**Host Computer System**

We chose PC type systems, as hosts they can run a modern operating system. While very compact hand-held PC type systems exist, they can be expensive. The laptop computer and netbook used for this research are each inexpensive and relatively small. More details regarding the operating system and selected software are given below.

In selecting the host we did not consider a small microprocessor based embedded system for use in the classroom, as most often such things involve cross-platform programming, which can be tedious and complicated. At the opposite extreme, we did not select a prebuilt turn-key type system as the project itself is meant to provide educational value to a range of students.

**Data Acquisition**

The LabJack model U3-LV is a small and affordable data acquisition module that connects to a computer using the universal serial bus (USB). The following are some features provided by the U3-LV.

- 16 multi-purpose ports for digital or analog input, or digital output
- Analog to digital conversions produce 12 bit samples over 0 to 2.4 Volt span
- Full resolution sampling is guaranteed for sampling rates up to 2.5kHz.
- Supports software or hardware timed acquisition
- 2 analog outputs, each with 10 bit resolution over 0 to 5 Volt span

While the U3 performance is modest in comparison to other devices, it provides teaching opportunities. In particular, in this application given the need for matching samples, we consider the sampling delay between successive channels. We also designed a signal conditioning circuit to better match the PSD range to the span of the ADC.

**Software Choices**

The choice to use a PC type system and the LabJack U3 provides us with many options regarding the operating system and application software. As of this writing, the LabJack U3 has two sets of drivers that provide support for three different operating systems, namely Windows, Linux, and Macintosh OS X. A search in the LabJack user forums reveals some interest in the community for developing applications with tablet PCs.

Besides the lower level drivers, LabJack[5] supports LabJackPython which is a cross-platform library which allows Python[6] programmers to write applications for LabJack hardware, using the Windows, Linux, or the Macintosh operating systems. LabJackPython works on Windows using the LabJack UD Driver and on Linux and Macintosh OS X it uses the LabJack exodriver.

For this research we performed data acquisition using scripts written in Python and we later analyzed the collected data using MATLAB[7]. Similar tool such as Octave[8] or FreeMat[9] can also be used. For those less interested in data acquisition and analysis software, there are existing applications such as LabVIEW[12] that supports LabJack as well.

Python is a powerful and convenient scripting language finding acceptance in the industry for use in laboratory automation. LabJack[5] as well as National Instruments[3,4] both provide support for the Python language. MATLAB is used for technical computing and is well known for its educational value. All undergraduate students enrolled in our college are required to learn MATLAB and most learn it as freshmen in ES115 Engineering Computer Applications.

Students less interested in data acquisition software can run Python software standalone, without being aware of the actual language used. Students with only a modest interest can study the code given in Appendix A which serves as a complete example. For others, Hughes[2] has a comprehensive text that supports the use of Python in an undergraduate setting. Python provides ample opportunities for further development. The SciPy[10] and NumPy[11] libraries provide Python with extensive data analysis capability for extremely large data sets. Likewise, there are several graphical libraries to choose from. We suggest the wxPython library which is a cross-platform graphical user interface (GUI) toolkit built upon the wxWindows library.

Finally, for this research we first used the TinyCore[13] Linux operating system running on a laptop computer and later used a Windows 7 netbook. Given that our software runs on Linux, Windows, or Macintosh OS X, picking and an operating system is a modest choice. TinyCore is a very small Linux distribution, approximately 12MB in size, and is suitable for embedded PC type systems. Our laptop boots TinyCore Linux off a compact disk and runs entirely from RAM so that the system runs applications quickly and has short latency times. A USB memory stick provides persistent storage, allowing us to easily move data to other computer systems.

**Educational Opportunities and Hands-On Activities**

The vibration measurement system provides educational opportunities with hands-on experimental activities. To start, the acquisition module provides an opportunity to study sampling and given the need in equation (1) for matching samples, we examine the sampling delay between successive channels. The input signal conditioning amplifier provides an opportunity to study simple operational amplifier circuits as well as study instrumentation principles such as the notion of the common-mode and differential-mode with respect to signals. Finally our use of Python and MATLAB to perform the acquisition and analysis of data involves scientific programming.

**Sampling Delay Between Successive Channels**

While the LabJack U3 acquires samples from up to 16 analog inputs or *channels*, the device has only one analog to digital converter (ADC). The U3 actually acquires samples from channels in rapid succession, so that the U3 is not capable of sampling multiple channels at a single instance or in a single event. It is educationally valuable for students to investigate this phenomenon and verify that the delay between successive channels is small compared to the the sampling period.

To examine the sampling delay between a pair of channels, apply a triangle wave extending over the span of the device and examine the sampled waveform. The ideal waveforms are shown in Figure 3 and the difference between the waveforms is in Figure 4. The delay Td between channels is found using the triangle wave slope and the difference waveform.

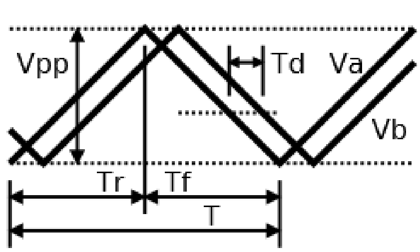


Figure 3: Triangle waveform

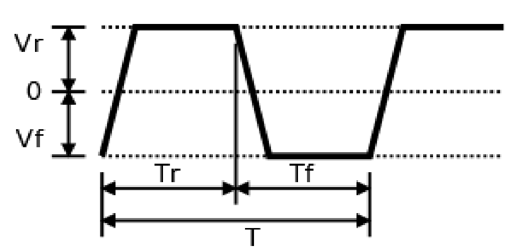


Figure 4: Difference waveform

In equation (2), *mr* and *mf* are the rising and falling slopes of the triangle wave and in equation (3), *Td* is the delay between channels.

$$mr = \frac{Vpp}{Tr} \qquad mf = \frac{Vpp}{Tf} \tag{2}$$

$$Td \approx \frac{Vr}{mr} \approx \frac{Vf}{mf} \tag{3}$$

To demonstrate the circuit, a 100Hz triangle wave was measured with the LabJack. The waveforms sampled by the first two channels is in Figure 5, both waveforms appear to overlap. Having the waveforms close in value causes the difference waveform to be reduced in resolution. Figure 6 is the difference waveform produced by MATLAB, which is comparable to the step-size $Vs$. The code is in Appendix A. The approximate value of the delay between successive samples $Td$ is 3.2% of the sample period $Ts$ or 0.1% of the period $T$ of the sampled waveform.

- $Vpp = 1.96\text{V}$
- $T = 10\text{ms}$, $Tr = 5\text{ms}$
- $Vr \approx 5\text{mV}$
- $Vs = 0.58959\text{mV}$
- $mr \approx 392\text{V}/Sec$
- $Td \approx 12.8\text{uSec}$
- $Ts = 400\text{uSec}$

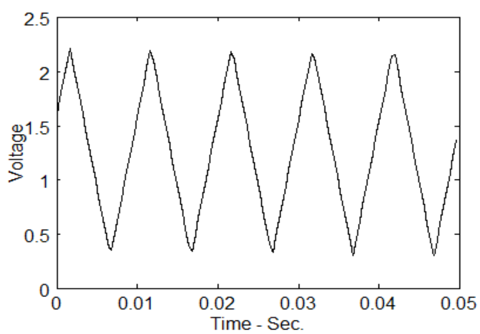


Figure 5: Sampled triangle waveforms

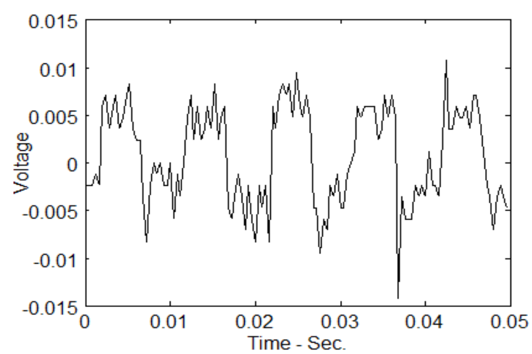


Figure 6: Difference waveform

If the delay between successive channels is ever significant, the data can be interpolated with MATLAB to produce an estimate of matching samples. To get a feel for the significance of Td, consider the measurement of a sinusoidal waveform. It is well known that the largest slope occurs at the zero crossing so that for a 100Hz, 2.4 Volt pk-pk sinusoid, the slope is no larger than 754.0 V/Sec, which corresponds to a worst-case error of 10mV between channels.

**Input Signal Conditioning**

The quality of samples produced by an ADC is related to how well the corresponding signal is matched to the ADC span. If the input spans half that of the ADC, then one sample bit is effectively lost. Likewise, with the input spanning one-quarter that of the ADC then two bits are effectively lost. This observation is important as we need to make the best possible use of the 12 bit sampling provided by the U3. A signal conditioning circuit helps to make better use of the span provided by the ADC, to keep the largest possible number of effective bits.

In considering equation (1), the denominator is the sum of V1 and V2, which is related to the common-mode Voltage and the numerator is the corresponding differential mode Voltage. Given the need for the common-mode Voltage, it is undesirable to use a conventional instrumentation amplifier, which produces only the differential mode. A first approach is to handle V1 and V2 separately as single-ended signals, using amplifiers like that in Figure 7. With V1 and V2 ranging from 0 to 0.6 Volts, and the ADC spanning 0 to 2.4 Volts, a gain of 4 is appropriate.

In cases where the AC component of a signal is significantly smaller than the DC component, consider the circuit in Figure 8 which provides AC gain $Av_{ac}$ that is larger than the DC gain $Av_{dc}$. Capacitor C1 is chosen for the desired corner frequency *Fc*. In comparing these two amplifiers, a pair of amplifiers like that in Figure 8 requires four additional components.

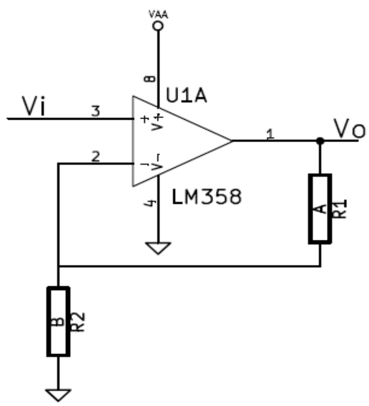


Figure 7: Non-inverting amplifier

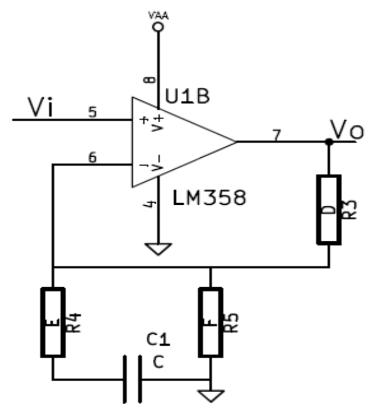


Figure 8: Amplifier with increased AC gain

$$Av = 1 + \frac{R1}{R2} \tag{4}$$

$$Av_{dc} = 1 + \frac{R3}{R5} \quad Av_{ac} = 1 + \frac{R3}{Rx} \tag{5}$$

where $Rx = \frac{R4\,R5}{R4 + R5}$ and

$$Fc = \frac{1}{2\pi\,R4\,C1}$$

In cases involving a differential signal with a DC component and the differential-mode signal is smaller than that of the common-mode, consider the circuit in Figure 9. This circuit does not use a capacitor in the signal path, allowing the circuit to work with DC signals as well. The circuit is constructed by replicating the amplifier in Figure 7 and inserting resistor R10. A roll-off can be applied to reduce the gain for interfering out of band signals by inserting capacitors parallel to R6 and R7, respectively.

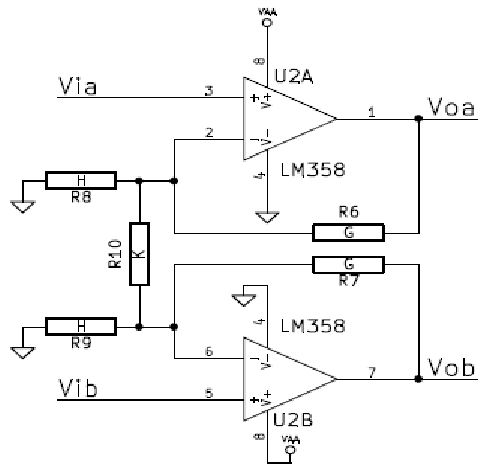


Figure 9: Differential and common mode gain circuit

The circuit in Figure 9 is essentially the first stage of a differential amplifier, but with R8 and R9 it is modified to provide additional common-mode gain. The common and differential mode Voltages defined in equations (6) are indicated with the subscripts *c* and *d*, respectively. Likewise, equations (7) define the common-mode and differential-mode gains. Equations (8) reverse equations (6), producing the input and output Voltages.

$$Vic = 0.5(Via + Vib) \qquad Voc = 0.5(Voa + Vob) \tag{6a}$$

$$Vid = Via - Vib \qquad Vod = Voa - Vob \tag{6b}$$

$$Avc = \frac{Voc}{Vic} \qquad Avd = \frac{Vod}{Vid} \tag{7}$$

$$Via = Vic + 0.5\,Vid \qquad Voa = Voc + 0.5\,Vod \tag{8a}$$

$$Vib = Vic - 0.5\,Vid \qquad Vob = Voc - 0.5\,Vod \tag{8b}$$

Interpreting the sampled values is straight-forward as equations (9) provides the relationships needed to evaluate equation (1) for the PSD. Equation (10) gives the gain values, based on given resistor values.

$$Via - Vib = \frac{Voa - Vob}{Avd} \quad \text{and} \quad Via + Vib = \frac{Voa + Vob}{Avc} \tag{9}$$

$$Avc = 1 + \frac{R6}{R8} \quad \text{and} \quad Avd = 1 + \frac{R6}{Ry} \quad \text{; where} \tag{10}$$

$$Ry = \frac{R8\,R10}{2\,R8 + R10}$$

**Constructing the Signal Conditioning Circuit**

The circuit can be constructed on a breadboard as we have done, or on a PC board. Our test instruments project webpage[15] will have artwork and a list of materials for constructing a PC board. The LM358 operational amplifier is one possible choice for this project. The LM358 is designed for low-power applications and is powered by the LabJack, which provides 5 Volt power. The LM358 device can pull it's output close to the more negative rail Voltage, which in this case is ground. The device can also swing it's output positively, to approximately 1.5 Volts below the positive rail or 3.5 Volts, which covers the ADC span of 2.4 Volts. For the U3, to produce correct conversions all the analog inputs must be in the range from -0.3 Volts to +3.6 Volts, but otherwise the ports can withstand slightly higher Voltages. The first 8 ports can withstand up to +/– 10 Volts and the rest can withstand up to +/– 6 Volts continuously. In a nutshell, the LM358 is safe and compatible for use with the LabJack U3 device.

Figure 10 is the configuration used to test the amplifier in Figure 9. The Voltage sources Vid and Vix together model a signal generator with a DC offset. The signal frequency was 100Hz. The following are the selected component values and the predicted gain values. The resistor Rx and capacitor Cx cause Vib to appear to be approximately the DC Voltage Vix.

- R6 = R7 = 3K
- R8 = R9 = 1K
- R10 = 1K
- R11 = R12 = 100
- *Avc* = 4
- *Avd* = 10
- Rx = 100
- Cx = 1mF

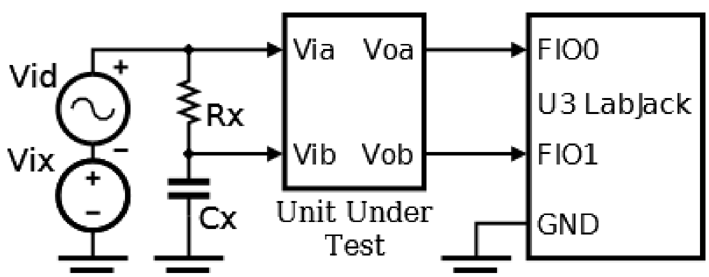


Figure 10: Test Circuit for signal conditioning amplifier

The applied Voltages in (11) can be produced with commonly used lab instruments. The common-mode and differential mode Voltages are in (12), and the outputs are in (13).

$$Via = 0.15\cos(\omega t) + 0.3 \qquad Vib = 0.3 \tag{11}$$

$$Vic = 0.075\cos(\omega t) + 0.3 \qquad Vid = 0.15\cos(\omega t) \qquad Voc = 0.30\cos(\omega t) + 1.2 \qquad Vod = 1.5\cos(\omega t) \tag{12}$$

$$Voa = 1.05\cos(\omega t) + 1.2 \qquad Vob = -0.45\cos(\omega t) + 1.2 \tag{13}$$

Figure 11 is the corresponding output. Note that as expected, while the amplitudes differ, the output appears to have the correct differential-mode and common-mode Voltages.

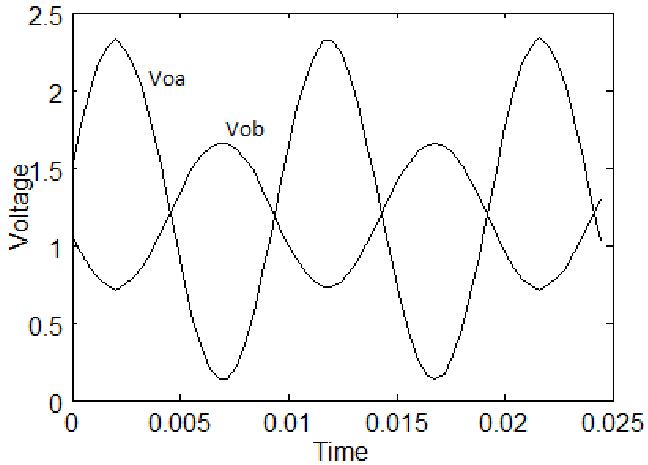


Figure 11: Measured waveforms from signal conditioning amplifier

**Mechanical Test**

The apparatus used in the vibration measurement system is general in concept. Figure 12 is an example system using a mass and spring apparatus. Three of the four vertical brackets suspending the mass by wire are visible in the figure. The suspended mass moves freely up and down in reaction to a force applied by a voice coil. The laser and PSD are above the mass. The LabJackU3 is centered in the foreground, the breadboard just to the left has the signal conditioning circuit. The amplifier used to drive the voice coil is in a case just above the breadboard. The suspended mass is above the amplifier case.

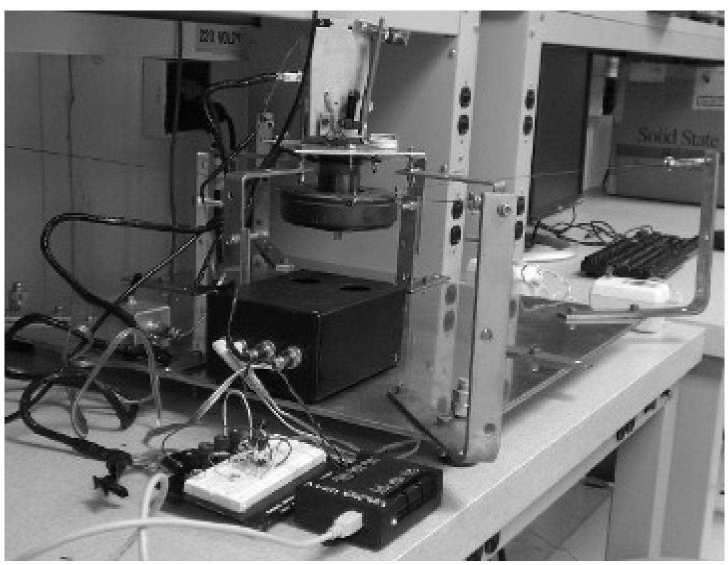


Figure 12: Mass and spring apparatus

A sinusoidal signal was applied to the voice coil and the frequency was adjusted for the resonant frequency, which was approximately 3.3 Hz, by watching for the maximum excursion. Figure 12 shows the calculated relative position where the laser beam is incident on the PSD and Figure 13 is the corresponding spectrum magnitude waveform.

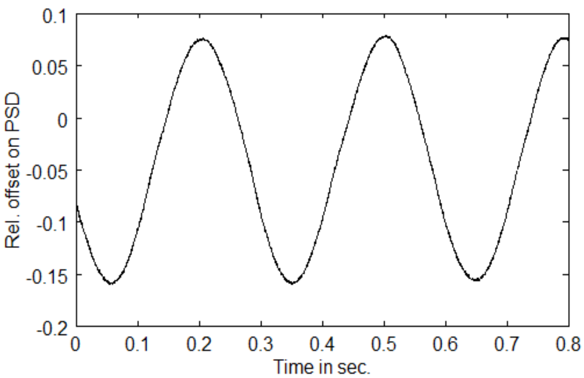

Figure 12: Relative position on PSD

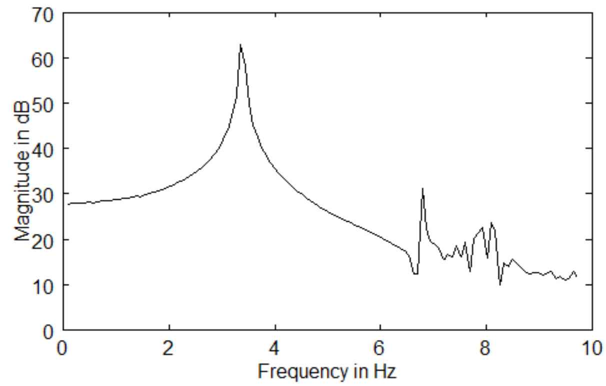

Figure 13: Magnitude Spectrum

Signal quality can be an issue, especially with a breadboard. A good ground is required and students must watch for the tell-tale sign of a peak in the spectrum near the frequency of the AC main. It was also found that having the laptop operating off its battery improves the quality of the measured signals. Finally, to reduce nuisance out band signals, we applied a roll-off to the gain near 100Hz, by inserting 0.47uF capacitors in parallel with R6 and R7, respectively

**Conclusion**

This project involved a laser based vibration measurement system with educational value. The system we developed is meant for cross disciplinary use, is small, inexpensive, and convenient, without extensive programming. The LabJack U3 acquisition system was first used with a laptop computer running the TinyCore Linux operating system and later with a netbook running Windows 7. We used Python and MATLAB for our application software. The system provides educational opportunities with hands-on experimental activities, in particular, the acquisition module provides an opportunity to study sampling and the sampling delay between successive channels. The input signal conditioning amplifier provides an opportunity to study simple operational amplifier circuits as well as study instrumentation principles such as the notion of the common-mode and differential-mode with respect to signals. Our actual use of the system to measure vibration provided ample opportunities for student projects.

**Appendix A, Example with Python and MATLAB**

Here we present the code we used to examine the sampling delay between successive channels. Listing 1 contains the stream06.py Python script. The sample period for each channel is 2.5kHz and the sample values produced are written to a simple text file. Listing 2 is the corresponding MATLAB script.

In stream06.py, samples are acquired from the first two channels, FIO0 and FIO1. The LabJack U3 is capable of hardware or software timing, here we use a hardware timer to acquire the samples. The U3 itself stores samples in a buffer, which the host reads. Hence, to avoid losing samples the host must read the buffered samples in a timely fashion.

An important feature of the Python language is that line indentation is an element of the syntax. In the listing, based on the indentation, lines 20 and 40 mark the top and bottom of a loop.

Likewise, lines 21 and 38 mark the if and else parts of a test structure. The following is an outline of the code:

- lines 03, 04 – user defined constants
- lines 05 to 07 – open LabJack library and device, report device name
- lines 10, 13 – configure the device
- lines 16 to 18 – initialize variables, start sampling, open output file
- line 20 – top of loop, read a packet
- line 21 – verify reception of a valid packet, else go to line 38
- lines 22, 23 – check for exit condition
- lines 24 to 30 – check for errors, overflow, and missed samples
- lines 31 to 35 – write samples to file and report averages
- line 36 – increment counter value, go to line 20 to repeat loop
- lines 38, 39 – with an invalid packet report a timeout, go to line 20 to repeat loop
- lines 40 to 47 – close file, stop sampling, close device, produce final report

The code was executed on the following two hosts, a laptop computer running TinyCore Linux as well as a netbook running Windows 7. Both computers execute the same script using LabJackPython, which is aware of the underlying operating system and calls the appropriate LabJack driver code.

- Toshiba Satellite Laptop model A135 with Intel T2080 dual core CPU @ 1.73GHz, 2 GByte RAM, TinyCore Linux version 4.02, Linux kernel 3.03, Python 2.7.2, exodriver labjackusb version 2.0.4, LabJackPython-8-26-2011

- Toshiba netbook model NB305 with Intel Atom N455 CPU @ 1.66GHz, 2 GByte RAM, Windows 7 starter edition service pack 1, Active State Python 2.6.6 (32 bit version), UD driver version 3.25

In running the stream06.py script on the laptop, there were no missed samples. In running the script on the netbook, missed samples were observed but only rarely. It was also found that running the script on the netbook in the Idle development system for Python slows execution such that samples are regularly missed; this aspect was not tested for on the laptop.

The MATLAB code is straightforward. The code in Listing 2 is used to produce Figures 5 and 6. Following the comments, the code in lines 03 to 06 read in the sampled data, compute the difference waveform, as well as the sample period Ts, and the corresponding sample times. Lines 08 to 11 plot the sampled data and lines 14 to 16 plot the difference waveform. The given code in Listing 2 was executed with MATLAB 7.12.0, Octave 3.2.4 and GnuPlot for Windows, as well as Freemat 4.1 for Windows.

```
# stream06.py - Derived from streamTest.py example in LabJackPython
# This acquire two streaming channels from a LabJack U3
MAX_REQUESTS = 50           # the number of packets to read.
FILE_NAME = "SysAcq06.dat"  # name of file produced
import u3                   # open u3 library
d = u3.U3()                 # open and report the device
print d.configU3()['DeviceName']

# bitmap representing channels, with FIO0 being the lsb
d.configIO(FIOAnalog = 0x03)

# For single ended channels, match each with NChannel 31
d.streamConfig( NumChannels = 2, PChannels = [ 0,1],\
                NChannels = [ 31,31], Resolution = 3,\
                SampleFrequency = 2500 )
missed = 0; dataCount = 0
d.streamStart()
myfile = open(FILE_NAME,"w")

for r in d.streamData():
    if r is not None:
        if dataCount >= MAX_REQUESTS: # The stop condition
            break
        if r['errors'] != 0:
            print "--- Error: %s ; " % r['errors']
        if r['numPackets'] != d.packetsPerRequest:
            print "--- Underflow : %s : " % r['numPackets']
        if r['missed'] != 0:
            missed += r['missed']
            print "--- Missed ", r['missed']
        nn = len( r['AIN0'] )
        for ii in range(nn):
            myfile.write( "%f %f\n"  %(r['AIN0'][ii],r['AIN1'][ii]))
        print "Avg1",nn,"reads: %.4f" % (sum(r['AIN0'])/nn),\
              "Avg2",nn,"reads: %.4f" % (sum(r['AIN1'])/nn)
        dataCount += 1

    else:  # USB timeout exceeded, approx. 1 sec.
        print "No data - timeout"
myfile.close()
d.streamStop()
d.close()

total  = dataCount*d.packetsPerRequest*d.streamSamplesPerPacket-missed
print "%s requests, %s packets per request, %s samples per packet" %\
      (dataCount, d.packetsPerRequest, d.streamSamplesPerPacket)
print "%s samples missed, %s samples acquired" % (missed,total)
# end of stream06.py
```

Listing 1: Example Python code

```
% timediff06.m - Jonathan Hill
% read samples from waveforms and plot difference
sd = load('sd_100hz_triangle.dat');
tdiff = sd(:,1) - sd(:,2);
Ts = 1/2500
x = (0:124)*Ts;

figure(1)
plot([x' x'],sd(1:125,:),'k')
xlabel('Time - Sec.')
ylabel('Voltage')

figure(2)
plot(x',tdiff(1:125),'k')
xlabel('Time - Sec.')
ylabel('Voltage')

% end of timediff06.m
```

Listing 2: Example MATLAB code

## References


1. D. Shetty, J. Kondo, and S. Noriega, "Non Contact Vibration Analysis Using Innovative Laser Based Methodology," ASEE National Convention, 2006 - 2619.
2. John M. Hughes, Real World Instrumentation with Python: Automated Data Acquisition and Control Systems, copyright 2011 by John M. Hughes, published by O'Reilly
3. National Instruments, "Calling Python Code from LabVIEW," modified Jan. 21, 2010, https://decibel.ni.com/content/docs/DOC-8995
4. National Instruments, "Using Python for Rapid Prototyping," modified Sep. 15, 2011, https://decibel.ni.com/content/docs/DOC-16251
5. LabJack Corp., http://www.labjack.com/
6. Python official website, http://www.python.org/
7. MathWorks MATLAB, http://www.mathworks.com/
8. Octave, http://www.gnu.org/software/octave/
9. FreeMat, http://freemat.sourceforge.net/
10. SciPy Scientific Tools for Python, http://www.scipy.org/
11. NumPy Numeric Tools for Python, http://numpy.scipy.org/
12. National Instruments, http://www.ni.com/
13. TinyCore Linux, www.tinycorelinux.com/
14. wxPython library, http://www.wxpython.org
15. Test Instruments Project, http://uhaweb.hartford.edu/jmhill/projects/testinst/