INTRODUCTION TO LABVIEW

In this lab we are going to understand the fundamentals of labVIEW. We further go on to perform certain experiments using labVIEW simulations. The Agilent scopes are also used to prove that the simulations are just as accurate as performing the experiment using actual hardware.

LabVIEW is short for Laboratory Virtual Instrument Engineering Workbench, is a programming environment in which you create programs with graphics; in this regard it differs from traditional programming languages like C, Pascal in which you program with text. However LabVIEW is much more than a language. It is a program development and execution system designed for people, like you engineers, who need to program as parts their jobs.

Using very powerful G language programming, LabVIEW can speed up programming considerably as it is specifically designed to take measurements, analyze data, and present results to the user. It is more versatile than bench top instruments as you, not the instrument manufacturer define instrument functionality. Because of LabVIEW’s graphical nature, it is inherently a data presentation package. Output appears in any form you desire. Charts, graphs, and user-defined graphics comprise just a fraction of available output option.

LabVIEW programs are called virtual instruments (VI’s) because their appearance and operation imitate actual instruments. Behind the scenes they are analogous to main programs, functions and subroutines from popular programming languages like C or BASIC.

The VI has three main parts:

- The front panel: this is the interactive user interface of the VI, so named because it simulates the front panel of a physical instrument. It can contain knobs, push buttons, graphs and other controls that can be inputted by the user.
- The block diagram is the VI’s source code constructed in the programming language G. This is the actual executable program. The components are lower level VI’s. built in functions, constants and program execution control structures. Front panel objects have corresponding terminals on the block diagram so data can pass from the user to the program and back to the user.
- In order to use the VI as a subroutine in a block diagram of another VI, it must have an icon and a connector. A VI used within another VI is called a subVI and is analogous to a sub-routine.

This was some of the introduction to LabVIEW. Now it is time we perform some experiments to actually get acquainted with the software.

- Open the National Instruments LabVIEW from the desktop.
• Close the VI windows that opens along with the wizard.
• On the main LabVIEW window click on DAQ solutions.
• Click on view current channel configuration. Ideally it should not have any channels defined.
• Close that window and click on DAQ channel wizard.
• Right click on data neighborhood and select create new.
• Create a new virtual channel.

Now select analog input and follow the self-explanatory instructions to create a channel that has an input range of plus/minus 5V. Select channel 0 first to be the designated input channel. Thus you have to create a channel that measures voltage. Give comments wherever there are spaces given and write in brief about the type of channel and the range it has. Select the single ended referenced input when configuring the channel.

Close the window and then click on view current channel configuration again. This time you should see the channel that you just created.

• Now click on next and select solutions gallery before clicking next.
• Select bench-top instruments and select the 1 channel oscilloscope.

Select the channel you have described and then click open to view the 1 channel oscilloscope.

A new VI should open up where we see a graph and some other buttons. This is the virtual oscilloscope. To see the block diagram, go to window and click on show diagram. This is the connections diagram and the corresponding G code for the oscilloscope we just opened.

We need to give an input to this oscilloscope. This is done through an external hardware function generator. We use the Agilent 33250A for this experiment. Set up the function generator so that it produces a 1kHz sine wave with a peak-to-peak amplitude of 1V_{p-p}. Make sure that the function generator is in the high Z mode.

Now connect the “T” BNC connector so that the main connector is connected to the function generator and there are two more BNC connectors available to connect two other wires.

The external signal is fed to the computer using the National Instruments board. One wires from the function generator is connected to the input channel that was designated for input from the oscilloscope while the other wire is connected to the “ref” connection, which is the bottom most single connection on the same board. This is the ground connection for the board. If we do not connect this we have a floating ground problem. This is overcome using the ground reference. Now we can see the signal on the scope.

We have successfully been able to configure LabVIEW to accept signals from an external source.
USING THE A/D CONVERTER IN THE OSCILLOSCOPE.

For the further experiments, take plots for what ever you feel will be better explained with the use of the graph.

Using the same frequency we used for the simulations using LabVIEW, we put that signal in the *Agilent 54621D* oscilloscope.

- Select the math function and the FFT to obtain the spectrum of the sine wave.
- Change the time axes to change the number of samples/sec, making it 20kSa/sec.
- Choosing the more FFT function, change the scale to 10dB.
- Now select the acquire function and select the averaging function. Let the number of averages be 16384.

We now have the frequency to 1kHz. Observe the spectrum and explain the peaks obtained at the 1,3,5 harmonic.

Now increase the frequency in steps of 0.01kHz to 1.1kHz. Observe the change in the spectrum and explain the new spectrum. Turn the scope to normal instead of averaging signal and then see the spectrum. We see that the peak we saw that was created with averaging was actually just a shift in the peak. But since the scope is in averaging, we see two peaks.

The above two parts of the experiment will enable us to perform the other experiments described below.

ALIASING

For this we need to open a virtual spectrum analyzer. Using the same steps described above we open a spectrum analyzer instead of a 1-channel oscilloscope. The steps are described below for convenience.

- From the LabVIEW wizard select solutions gallery.
- There select bench top instruments and spectrum analyzer.
- Select the channel that you have configured and then click open.
- This opens the virtual spectrum analyzer.

On the virtual spectrum analyzer we see that the instrument front panel is divided into two sections. The upper half that represents the input signal in time scale and the bottom that represents the signal on a frequency axes as a spectrum analyzer.

The input is a 1kHz sine wave. From the controls in the front panel of the spectrum analyzer we select the sampling rate to be 22050 and the frame size to be 1024. These are important parameters that help us view aliasing correctly.
On the spectrum analyzer we see a peak at 1kHz. This corresponds to the sine wave. The frequency of the display can be defined as $f_s - f_{in}$, where $f_s$ is the sampling frequency and $f_{in}$ is the input frequency. Sampling frequency is approximately 22kHz. Thus as we change the value of input frequency, the display frequency also changes. Change the input frequency in steps of 1kHz. We see that the peak shifts by 1 kHz every time we increase the frequency. After 11kHz, the peak shifts in the opposite direction. Every time we increase the frequency by 1 kHz, the peak shifts to the left now. Explain this phenomenon.

**CLIPPING DISTORTION**

For this part of the experiment we keep the frequency of the sine wave at a constant 100Hz instead change the amplitude of the signal. This frequency was chosen so that we can actually see the sine wave clearly on the spectrum analyzer.

Now increase the amplitude of the sine wave and observe the time scale graph on the virtual spectrum analyzer till we see clipping occur. At the exact point where clipping occurs observe the spectrum in the frequency spectrum. We see that a lot more peaks appear where clipping begins to occur. This means that the quantization noise is greater than $Q^2/12$, which is the theoretical noise between $0 - f_s/2$. Explain the reason for this.

Now switch back to the Agilent 54621D oscilloscope and view the FFT of the signal. Make the necessary changes so that we can view the spectrum correctly. Now we reduce the amplitude. Reduce the amplitude from 1V$_{p-p}$, to 100mV. Now see the averaged spectrum. Recall that we can obtain averaging by selecting the acquire option on the oscilloscope and then selecting averaging from the on screen menu. How many dB has the peak gone down by? Now reduce the amplitude further till we loose the signal in the noise (without averaging). What is this amplitude?

**TONES**

For the first part of this experiment we use the virtual spectrum analyzer from labVIEW. Now in the spectrum analyzer, change the sampling rate to 10012 and the window size to 1027. Sine wave has a frequency of 1kHz and the amplitude is 1V$_{p-p}$. Explain the spectrum that you obtain.

Now switch to the Agilent 54621D oscilloscope. View the same signal as a FFT on the oscilloscope first without the averaging function and then with the averaging function on. Detune the circuit by 100Hz and again observe the circuit. Observe the spectrum with and without averaging. Explain the spectrum.

*Don’t forget to clean up before you leave your table!!*