I. Getting to know Labview / Matlab

A. Objective

This lab will introduce you to LabView and the setup available to you in the lab. By the end you will be able to write simple LabView programs, as well as collect basic data using the hardware setup.

B. Needed Hardware

- 1kΩ Resistor
- Wire
- Wire cutter/stripper
- 1 solderless breadboard

C. Pre-Lab

- Read through the lab.
- Go through the LabView Introduction in 3 hours material (note: you don't have to do the exercises, I just want you to familiarize yourself with the LabView environment)
D. Background

1. LabView
LabView is a graphical programming package from National Instruments. Like C/C++ it allows you to control the computer and make it work for you. However, unlike C/C++ all of the functions are pictures!

LabView programs are called Virtual Instruments or VI's. All of the programs that you write will have icons instead of lines of text. Each VI has 3 parts:

- Front Panel
- Block Diagram
- Icon/Connector

The Front Panel should be thought of as the user interface. Look at the oscilloscope that is on your desk. There are many buttons, as well as a large screen where you can see waveforms. The Labview “Front Panel” is where you will have buttons, and graphs and controls that allow the user (you) to interact with the underlying code. It is the GUI. This interface can be as basic as a simple plot to a very complicated interface similar to the most sophisticated oscilloscopes.

The Block Diagram is the underlying code that controls (or is controlled by) the Front Panel. Think about everything inside of the oscilloscope. When you push a button on the front panel, something happens inside the box. The code that executes is equivalent to the Labview “Block Diagram.” Anything that you place on the Front Panel (buttons, graphs, indicators, etc.) will have a corresponding block (icon) on the block diagram.

The Icon/Connector is what allows you to take your code and reduce it down icon, thus making a sub-VI which could be used in other more complicated programs. This third option won't be used much early on, but know that it exists for the future.

2. Matlab
Hopefully, you are somewhat familiar with Matlab. It is a software tool that allows for complex mathematical analysis. We will make heavy use of its plotting and curve fitting capabilities.

You can get help on Matlab functions by typing “help <function>” or “doc <function>”. They both will return to you how to use particular functions. The difference is that help puts to the text in the Matlab command window. Doc opens a pretty box which also has searching capabilities (useful if you don’t know the name of the Matlab function that you are looking for) and the ability to follow links to help on other function names.
Some functions that you may want to familiarize yourself with are:

- `plot`
- `csvread`
- `polyfit`
- `polyval`
- `semilogy`
- `log vs. log10`

### 3. Data Acquisition

In this course we are going to be doing a lot of data acquisition. There are a few key things about data acquisition that you need to understand. First, in the real world... nothing is fixed! Assume that you are placing 3V on a line. At any given point in time, it might be 3.0V, 2.99V, 3.01V, etc. Really cheap power supplies might fluctuate even more. (In fact one of the differences between a $10 power supply and a $10,000 power supply is how much ripple you are going to see).

Let’s assume that you have a power supply that you have set to 5V. Next, assume that you read what voltage that power supply is putting out 1,000 times over the next second. Some of those readings will be precisely 5.0V, many will not. Look at Figure I-1. Assume this figure shows 1,000 readings from 3 different power supplies (Note that these are exaggerated for effect). The x-axis shows the value that we read from the power supply. The y-axis shows the number of times we got that particular value.

The average of those 1,000 readings determines what we call the **Accuracy** of the data. Note that the red curve and the blue curve both have the same accuracy. If you take the average of the red curve and the blue curve you will get the same number: 5.0V. However, the average of the yellow curve is around 5.3V. So in this example, the red and blue power supplies are both accurate, while the yellow is not. (Note that this is the technical use of the word accurate, not the common use). However, the blue and the yellow curves look different from the red curve. The width of the curves determines the **Precision** of the data. The blue power supply is as accurate as the red curve, but it is much more precise. The blue and yellow power supplies are equally precise, but the blue one is more accurate. (So the yellow power supply is precisely wrong, while the red curve is imprecisely right... weird right?)
The funny thing about data acquisition is, not only might the power supply be moving, but the piece of equipment taking the reading might be moving around as well! In reality there is nothing we can do to completely get rid of this movement, but we buy expensive equipment to take our readings because the manufacturers of that equipment have taken care to minimize the effects of their equipment on our readings. Generally, the more expensive the piece of equipment, the less you have to worry about it... but you can never totally get rid of it.

Sitting on the lab benches are many different types of meters. However, all of these meters will read whatever data they are set up to read, and will convert that information to a representation that the computer can understand (or you can on the displays). These pieces of equipment communicate with the computer via 1 of 2 different cables (standards). The first is GPIB. The Keithley 6485 and the Agilent 34401 both use this method. GPIB is an old standard, but widely supported. USB is the other. This is newer and faster, but is not as widely supported... yet. The function generator and the oscilloscope both use USB. (Note that we have installed a GPIB to USB converter to talk to the GPIB equipment).

Labview can talk to any piece of equipment on your bench except the oscilloscope. Generally though, the oscilloscope is for your benefit, and not for taking data. It is like a window into what is going on at that moment, but you don't usually pull data directly from it. Agilent, however, has provided us with software that allows us to grab data from the scopes and save it to files on the computer. More later...

You also have the National Instruments USB-6009 in front of you. This piece of equipment has 2 D/A units and 8 A/D units on it (as well as some digital... but who cares about digital?). D/A's are digital to analog converters. This means, you tell the computer to send a digital code to this converter... and the converter will take that code and convert it into and analog value. A/D's do the opposite. In other
words, you have the ability to read 8 separate voltages in your circuits, and you have the ability to control 2 with these cards. This will be important!

In addition to the USB-6009, we have a National Instruments PCI-7831. This is located inside the computer, and you interface to it through the SCB-68 box on the table in front of you. The PCI-7831 is a really cool card which has 8 D/As, and 8 A/Ds. It also has a 1 million gate FPGA on it. This makes it both cool... but a bit more complex to use than the USB-6009. The real benefit to using this card comes when you need a negative voltage. The PCI-7831 is capable of generating voltages from -10V to +10V. The USB-6009 can only do 0V to +5V. When you get to diodes... you’ll see why negative is good...

As this is just an introduction, we will stop here for now. However, there is still much to talk about... but don't worry. You'll catch on.

E. Assignment 1

1. Intro
First you need to watch the video tutorial which discusses some of the basics of Labview. In this I take you through the basic parts of a Labview VI, and I will take you through the code I have written for today’s laboratory. You can find this located at:

http://www.ece.utk.edu/~farquhar/labwiki/doku.php?id=tutorials

I have provided a simple LabView VI (Read_1E6) whose sole purpose in life is to read 1,000,000 data points from A/D channel 0 (the first one) and write it to a file. (You can find it on the laboratory portion of the website.)

http://www.ece.utk.edu/~farquhar/Demos/Labview/Read_1E6.htm

This year, you will primarily be using the FPGA based PCI-7831 acquisition card, which is located in the back of your computer. It connects to the real world via the SCB-68 breakout box. For this particular lab, you do not need to know much about what is going on in this card. However, there are a few key things that you need to know.

For each piece of code that you will write in this class, you will need to write 2 actual pieces of code. 1 is to run on the FPGA itself (which controls all of the low level hardware issues), and the second runs on the PC. There is a video tutorial of this located at:

http://www.ece.utk.edu/~farquhar/Demos/Labview%20FPGA/FPGA_DEMO.htm

I have written both pieces of code though... so for now, you just need to run the code.
2. **Experiment 1**
   1. Set your bench top power supply to 5.00V.
   2. Connect Analog Input channel 0 of the PCI-7831 to the power supply (make sure to connect the GNDs) (pins 68 and 34). (See the picture of the connector at the end of the lab)
   3. Run the VI by clicking the Run button (See Figure 2) and clicking on the Acquire button.
   4. Open the data in Matlab using csvread.
   5. Plot the magnitude of the data vs. the data point (use the plot command). Save this figure to turn in.
   6. Plot the histogram of the data using 50 bins (use the hist command). Save this figure to turn in.
   7. Calculate the average of the data.
   8. Calculate the standard deviation.
   9. Calculate the variance.
   10. Calculate the \% accuracy of this power supply.
   11. Now connect Analog Input channel 0 to the +5V supply on the PCI-7831 (pin 1 and the chassis of the SCB-68 which is GND) and repeat from step 3.

   Compare and contrast the results of the 2 power supplies. Which is more accurate? Which is more precise. Why do you think this is?

![Figure 2 Read_1E6 interface. The Labview "Run" button has been highlighted. Push the Acquire button to read 1 million data points after pushing the run button.](image)
F. Assignment 2

1. Intro
You are going to do a voltage sweep on your resistor and collect the current data. Using Figure 3 as a guide, wire up your resistor to Analog Output 0 of the PCI-7831. Connect the other end of the resistor to the 34401 which is set up in DC current measure mode. Both the PCI-7831 and the 34401 are already connected to the PC.

![Figure 3 Block diagram of Setup for Experiment 2](image)

Again, I have created a VI that does most of what you need. The VI is called `HOST_V_I_Sweep_HP34401`. Don’t confuse it with `HOST_V_I_Sweep_K6485`, which does the same thing but uses the Keithley 6485 meter. This meter is specially designed for reading extremely small currents. The HP34401 reads moderately sized currents.

Study the block diagram, and figure out what it is doing. In future labs, you will need to be able to build (or expand this code) something which does similar operations.

2. Experiment 2
   1. Use the VI to get the Current vs. Voltage data from the resistor.
   2. Plot the data in Matlab.
   3. Using the Matlab functions polyfit and polyval, plot a curve fit to this resistor on the same plot as above. Save this plot and hand it in.
   4. What is the slope? (Here I don’t mean simply the number, although I want to know what the number is... but what does the slope of this curve tell you? See Step 5)
   5. Using the curve fit information, determine the actual resistance of your resistor.
   6. Verify the above using the 34401 (this time configured as an Ohmmeter).
Figure 4 Connector Pinout. The numbers here correspond to the numbers on the SCB-68. So, for instance, to read a voltage in to AI0, connect the positive to pin 68 and the negative (or GND) to 34.
| LabView in 3 hours                        | ________________________________ |
| What were your Power Supply Averages?    | ________________________________ |
| What was your standard deviations?      | ________________________________ |
| What was your variances?                | ________________________________ |
| What was the % accuracy of these power  | ________________________________ |
|   supplies?                             |                                  |
| Appropriate curve fit to resistor data? | ________________________________ |
| What was your slope?                    | ________________________________ |
| What was your actual resistance?        | ________________________________ |

TA Signature: ________________________________