August 27, 2013

Laboratory #2 Guide: Introduction to Laboratory Hardware and Software

Introduction

In the last lab we did a simple exercise to introduce students to the major components in our optical breadboard set-up, and used a simple LabVIEW (LV) program to acquire data relating to the time constant of a chopped thermopile detector. A wavelength near the peak output of the lamp was selected to maximize the signal. The data was analyzed using Igor Pro Software (IP). Thus, students were exposed to three key elements of the laboratory course; LV, IP, and several pieces of hardware and instrumentation.

In order to understand how light interacts with condensed matter/solid state materials and devices one needs to understand in detail the qualities and quantities of the light associated with the source. In the laboratory, we are concerned with the output of the lamp/monochromator combination and, to a lesser degree, the lenses and other optical components that may be employed in an experimental set-up. We want to understand the power and photon number distribution as a function of wavelength.

In this lab we will characterize the output of the monochromater/lamp combination as a function of wavelength. This activity will further build-up our core capabilities in LV so that more complicated experiments related to photovoltaic devices and materials can be done as the course progresses. Also, a detailed characterization of the light source will be required for a variety of future experiments and analyses.

Goals of Lab #1: Characterize the Output of the Lamp/Monochromator Combination

You will construct a LV program that will step the monochromator from an initial to a final wavelength at selected intervals (e.g. from 900 nm to 300 nm, at 10 nm steps). The thermopile detector will be used to measure the power at each wavelength. The data will be plotted and recorded.

Concepts

The lamp is Blackbody source, much like the Sun, that generates a broad distribution of wavelengths. The monochromator selects a fraction of this output depending on the position of the internal diffraction grating and the width of the slits. The output power will be a function of the slits used, the optical throughput of the monochromator, and the intensity of the lamp. The transmission characteristics of the lenses can also be important in some cases, but we will neglect these issues here. The output can be specified in terms of Watts/cm^2-nm, or # photons/cm^2-nm since the energy of a photon equals hc/λ.. The “cm^2” term refers to the area over which the photons are incident. We will not specify this area now, but consider it to be a constant for the experiments done in the lab, so long as no optical components are moved during measurements that will be compared. The "nm" refers to the optical bandwidth of the monochromator, and depend on the slit width – see the accompanying lecture notes.

Experimental Steps

1. Start at the end of Lab #1 (basically, steps 2 – 10 of Lab Guide #1), which boil down to:
    a. Establish communication with the monochromator (MC) (Note that COM Port 4, rather than 7, may be operational for communications with the MC).
    b. Establish data acquisition from thermopile through the USB 6009.
    c. Set-up and align optical components to maximize signal. Though you may wish to have the Chopper operating to do this, it will not be used for this experiment.

2. Open a new VI in LV. On the BD, pop-up on the Function Palette, go to Exec Control, and place a While Loop in the BD. Wire an indicator to the "i" terminal (aka, the counter). Go to the front panel and run the VI. Note the behavior of the Numeric indicator on the FP. Press Stop to end.

****Helpful Hint: Control-E switches back and forth between the FP and the BD*****

3. Return to the BD, and Exec Control within the Function Palette, and drop in a Time Delay. Set to 1 s, and run the VI again. Note and understand the new behavior of the Numeric indicator on the FP.

4. Return to the BD, replace the time delay with a DAQ Assist (DA). Set it up to acquire data from the Thermopile. Note that the settings you provide to the DAQ assist will act as a time delay. Remember these in case you need to change them. Even better, wire up front panel controls for the key Rate and # of Samples inputs.

5. Return to the BD, pop-up on Function Palette and expand it with the small arrow at the bottom. Then, go to Mathematics\Prob&Statistics and drop in the Mean VI. Wire the data output of the DA to the input of the Mean VI. The data stream should convert to the correct type automatically. Bring the output all the way to the right hand wall of the While loop, and enable indexing on the block connection that appears.

6. Run the "i" terminal to the While loop wall and enable indexing as well.

7. Outside of the While loop, on the right hand side, create a Bundle element (you can use the search function on the Function Palette to find this, or expand the Function Palette, go to Programming\Cluster, and select Bundle), and wire up

the “i” array and the mean array to the inputs to form a cluster of two one dimensional arrays. Now wire the output of the Bundle to an XY graph. Run the VI to make sure the output is properly updated.

Hint: With contextual help on, you can always place the cursor over an item of interest and learn more about it **

8. To store the data, take the two arrays and form a new two dimensional array (use Build Array in Programming\Array). Then couple the new two dimension array to Write to Spreadsheet File (located in Programming\File I_O). Run the VI and make a test data file. Open the file and look at the organization of the data. It’s llikely you’ll see two rows of data in N columns. At this point the data cannot be easily read into Igor Pro, so transpose the two dimensional array before it is written to the File using Transpose Array (in Programming/Array).

*******Hint: Whenever writing data to a file, inspect the file to make sure the data is written in the format that you will need later, and that the acquired data is the same as the written data.**************************

9. Now add a Goto11x.vi within the While loop. Create a control for the Initial Wavelength on the front panel. Use the “i” terminal to act as an indexer that can be multiplied to increment the wavelength according to an interval control that can be created on the front panel. Use the natural updating of the While Loop to control updating of the values written to Goto11x.vi. You can stop the While Loop manually with the provided button, or use logic to stop the scan when the final wavelength is reached. At this point, you should be able to collect the needed data to finish the report for this lab.

10. **Additional activity:** You have noticed that the data is not updated until the loop is exited. It would be preferable to observe the data while it is being collected. Besides being more satisfying, the data collection can be abandoned earlier of something is not quite correct. However, more complicated programming is required to update the data on the screen. One approach that may work is shown in the following screen shot. Note that in this example we are simply plotting the ‘i” array versus itself, for demonstration purposes. The same thingk could be implemented for the separate voltage and wavelength waves that our program is creating. The key difference from what was implemented previously is that the arrays are built-up within the While Loop using an Array variable and associated Local Variables that can be specified as Read Local or Write Local. (Local Variables can be created from the Programming\Structures palette for an already created variable). The 1-dimensional array variable is wired into a Build Array, as is the “i” terminal. You can pop-up on the Build Array and set it to concatenate data. The Array is initialized before the loop to be all zeros by writing in a an array full of zeros into a write local Array variable. During execution, the “i” (or wavelength, or voltage) values are added to the Array. The output with the added value is then used to replace the previsoulsy existing

Array via a second write local Array variable. This process continues until the loop is exited. The two arrays can then be bundled directly to the graph, which now can reside within the While Loop. The data is written to the file outside of the loop. Note that, in this case, indexing is not enabled fo the wire that goes through the While Loop wall.

Note that other approaches are possible as well. One which was discussed in class include the use of Shift Registers. No matter what approach you take it is important to do thorough testing to insure that the data you wish to save is the correct and in the right format.

Report

Report on experiment and present a plot of the power output of the Lamp/Monochromator in units of W/nm and photons/s-nm for two different slit widths.