

## LAB 7

In this lab, we will perform an experiment that demonstrates the concept of propagation of errors while learning how to use LabView to control the experiment. The experiment consists of measuring 160 pairs of resistors. The total resistance for each pair of resistors is

$$R_0 = R_1 + R_2 .$$

In practice, we will supply a constant current through the resistors and hence

$$V_0 = V_1 + V_2 .$$

Here you measure  $V_1$ ,  $V_2$ , and  $V_1 + V_2$ . By propagation of errors, the total uncertainty is just the radical of the quadratic sum of the errors of individual resistance,

$$\sigma_0 = \sqrt{\sigma_1^2 + \sigma_2^2} .$$

In this lab, you write the LabView program to read the three voltages. There is a large collection of plug-in boards, each containing sixteen pairs of resistors on each side. All the resistors are supposed to have similar values:

$$R_1 = 750 \Omega$$

$$R_2 = 1500 \Omega$$

Write a program to automate the measurement of 48 voltages on one side of a board. You then repeat the measurement with the other side of the board. This is followed by the measurements of next four new boards, yielding a total measurement of 160 resistor pairs. You should keep track of the serial number of each board so that you do not reuse the same board. Program the LabView to write the voltages to a text file that can be read by KaleidaGraph. Histogram the three voltage distributions and then re-plot each distribution using “Linear/Scatter” option so that you can superimposed the error bar of each data point. Fit each distribution to a Gaussian to extract the standard deviation. Is Gaussian a good description of the data, i.e. how is the  $\chi^2$  of the fit compare with the number of degree of freedom? Is the total uncertainty consistent with the expectation by propagating the two individual errors?

## Instructions to Create a LabVIEW Program for Acquiring the 48 Voltages on One Side of a Resistor Card

1. Open LabVIEW, open a new vi (file, new vi)
2. Click on the block diagram to select it
3. Add a for loop to the block diagram
  - 3.1. Right click in space and the functions palette will appear
    - 3.1.1. Select the programming menu
      - 3.1.1.1. Select the structures menu
      - 3.1.1.2. Choose the for loop function and then draw the loop in the block diagram
      - 3.1.1.3. Make sure the loop somewhat large, to house other code, by stretching it out using the position/size/select tool
4. Set the number of loops to run (since there are 16 resistor pairs per side of the resistor card, we will perform 16 measurement operations)
  - 4.1. Right click on the for loop N and choose create constant
  - 4.2. Using the edit text tool, change the constant to 16
5. The measurements need to be performed in a sequence, so add a stacked sequence inside the for loop
  - 5.1. Right click in space and the functions palette will appear
    - 5.1.1. Select the programming menu
      - 5.1.1.1. Select the structures menu
      - 5.1.1.2. Choose the stacked sequence structure function and then draw the stacked sequence inside the for loop
      - 5.1.1.3. Stretch out the sequence structure to nearly fill the for loop by using the position/size/select tool
6. Now we need to control the FET on the resistor measurement board using the DAQ card
  - 6.1. Right click in space and the functions palette will appear
    - 6.1.1. Select the measurement I/O menu
      - 6.1.1.1. Select the DAQmx – data acquisition menu
      - 6.1.1.2. Choose the DAQ Assistant function and drop it into the stacked sequence
    - 6.2. Double click on the DAQ assistant to configure it
      - 6.2.1.1. Select generate signals, digital output, line output

- 6.2.1.2. Select port 0 line 4 (this connects directly to the NTA4001N FET gate on the resistor test card)
      - 6.2.1.3. Click finish
      - 6.2.1.4. The configuration window will appear, nothing needs to be changed so click ok to finish configuring the DAQ assistant
  - 6.3. Rename the DAQ assistant using the edit text tool to “FET Control”
  - 6.4. Set the input of the FET Control DAQ assistant appropriately to turn on the FET
    - 6.4.1. Right click on the data input and choose create constant
    - 6.4.2. Using the operate value tool, change the value for the zero element to be bold T (true), this will turn on the FET when the program is executed
7. Next we need to select which resistor pair to connect to the NI-6220 by setting the MUX
  - 7.1. Add a frame to the stacked sequence structure by right clicking on the filmstrip and choosing add frame after
  - 7.2. In the newly created frame (1) drop another DAQ assistant as in 6.1
    - 7.2.1. Choose generate signals, digital output, line output
    - 7.2.2. Select the 4 MUX control lines simultaneously, port 0 line 3, port 0 line 2, port 0 line 1, port 0 line 0, by holding down shift while clicking them
    - 7.2.3. Choose finish
    - 7.2.4. Again, the configuration window will appear and nothing needs to be changed so click ok to finish configuring the DAQ assistant
    - 7.2.5. Rename the new DAQ assistant “MUX”
    - 7.2.6. We want LabVIEW to cycle through all 16 resistor pairs. This is simple to perform by using the for loop’s “i” signal. However, we need to convert the integer loop number into a Boolean array for configuring the MUX DAQ assistant. Instructions to do this are as follows:
      - 7.2.6.1. Right click in space and the functions palette will appear
        - 7.2.6.1.1. Select the programming menu
        - 7.2.6.1.2. Select the Boolean menu
        - 7.2.6.1.3. Choose the number to Boolean array function and drop it in the frame with the MUX control DAQ assistant
      - 7.2.6.2. Right click in space and the functions palette will appear

- 7.2.6.2.1. Select the programming menu
  - 7.2.6.2.2. Select the array menu
  - 7.2.6.2.3. Choose the array subset function and drop it in the frame with the MUX control DAQ assistant
  - 7.2.6.2.4. Configure the array subset function by adding constant controls
    - 7.2.6.2.4.1. Right click on the index input and choose create constant
      - 7.2.6.2.4.1.1. Set the value to 0
    - 7.2.6.2.4.2. Right click on the length input and choose create constant
      - 7.2.6.2.4.2.1. Set the value to 4
  - 7.2.6.3. Connect the for loop “i” to the input of the number to Boolean array function
  - 7.2.6.4. Connect the output of the number to Boolean array function to the input of the array subset function
  - 7.2.6.5. Connect the output of the array subset function to the data input of the MUX DAQ assistant
8. Now it is time to measure VA, the voltage across the first resistor in the selected pair
- 8.1. Add a new frame to the stacked sequence as in 7.1
  - 8.2. In the new frame (2), drop a DAQ assistant as in 6.1
    - 8.2.1. Select acquire signals, analog input, voltage
    - 8.2.2. Choose AI0 and click finish
      - 8.2.2.1. This time, we need to modify the configuration slightly. Change the terminal configuration selector to RSE and click OK
    - 8.2.3. Rename the DAQ assistant to VA MEAS
  - 8.3. We need to configure the sampling of the signal
    - 8.3.1. On the rate input of the DAQ assistant, right click and select create constant
      - 8.3.1.1. Set the constant to 1000
    - 8.3.2. On the number of samples input of the DAQ assistant, right click and select create constant
      - 8.3.2.1. Set the constant to 100
  - 8.4. Since our signal is DC, it is reasonable to take an average of the 100 samples and use the result as our measurement

- 8.5. Right click in space and the functions palette will appear
  - 8.5.1. Select the mathematics menu
    - 8.5.1.1. Select the probability and statistics menu
    - 8.5.1.2. Choose the mean function and drop it to the right of the DAQ assistant
- 8.6. Connect the data output of the VA MEAS DAQ assistant to the X input of the mean function
- 8.7. Right click on the mean output of the mean function and select create indicator
  - 8.7.1. Rename the indicator “VA”
9. Now we need to turn off the FET to measure the second resistor in the selected pair
  - 9.1. Click on the frame number selector and go to the frame with the DAQ assistant named “FET control”. This should be frame 0.
  - 9.2. Right click on the film of the stacked sequence structure and choose duplicate frame
    - 9.2.1. This will copy frame 0 and insert the copy as frame 1. However, we want this to be the last frame (3). To move the copied frame, with the copied frame showing, simply right click on the stacked sequence structure film and choose make this frame and select 3
  - 9.3. Now navigate to the last frame of the sequence (3) and this should be your copy
    - 9.3.1. As mentioned above, we need the FET to be off so simply change the zero element of the input to the FET Control2 DAQ assistant from T to F
10. Now it is time to measure VB, the voltage across the first resistor in the selected pair
  - 10.1. Go to the frame with the measurement of VA and duplicate the frame as you did in 9.2
    - 10.1.1. Again the copy is in the wrong spot of the sequence so move the copied frame to the end of the sequence like you did in 9.2.1
    - 10.1.2. Navigate to the last frame of the sequence (4) and this should be your copy
  - 10.2. Rename the DAQ assistant to VB MEAS and rename the indicator to VB
  - 10.3. Lastly we need to modify the DAQ assistant so that it is reading the proper voltage
    - 10.3.1. Double click on the VB MEAS DAQ assistant
      - 10.3.1.1. Once the configurator opens, under the configuration tab, right click on the “Voltage” channel and select change physical channel. In the pop up

window, select ai1 instead of ai0 and click ok. Close the configurator by clicking ok.

11. Finally, we need to measure VA + VB
  - 11.1. Go to the frame with the measurement of VA and duplicate the frame as you did in 9.2
    - 11.1.1. Again the copy is in the wrong spot of the sequence so move the copied frame to the end of the sequence like you did in 9.2.1
  - 11.2. Navigate to the last frame of the sequence (5) and this should be your copy
  - 11.3. Rename the DAQ assistant to VA + VB MEAS and rename the indicator to VA + VB
12. If you followed the steps properly, the program should be runnable. Check this by looking at the run button to see if it is solid white. If it is not, debug your errors. If it is, switch to the front panel and arrange VA, VB, and VA + VB neatly.
13. Run the program and you should see VA, VB, and VA + VB changing on the front panel as the program executes
14. Now we need to write the data to a file that can be read into KaleidaGraph
  - 14.1. Click on the block diagram
  - 14.2. Right click in space and the functions palette will appear
    - 14.2.1. Select the programming menu
    - 14.2.2. Select the array menu
    - 14.2.3. Choose the build array function and drop it outside of the for loop
      - 14.2.3.1. The original instantiation of the build array function only has one input. We want to record the MUX channel, VA, VB, and VA + VB so we need 4 inputs to the build array function. To add these inputs, simply select the build array function and drag the bottom edge until 4 inputs are present.
  - 14.3. Connect the acquired data to the build array block
    - 14.3.1. Connect the for loop "i" to the top input of the build array block
    - 14.3.2. In frame 2, connect VA to the second input of the build array block
    - 14.3.3. In frame 4, connect VB to the third input of the build array block
    - 14.3.4. In frame 5, connect VA + VB to the bottom input of the build array block
  - 14.4. Right click in space and the functions palette will appear

- 14.4.1. Select the programming menu
- 14.4.2. Select the File I/O menu
- 14.4.3. Choose the write to spreadsheet file function and drop outside of the for loop to the right of the build array function
  - 14.4.3.1. Connect the appended array output of the build array function to the 2D data input of the write to spreadsheet file function
  - 14.4.3.2. Right click on the append to file input and choose create constant
    - 14.4.3.2.1. Change this constant to true
  - 14.4.3.3. Right click on the transpose input and choose create constant
    - 14.4.3.3.1. Change this constant to true
- 15. Now the program is complete. When a run finishes, a dialog window will pop up to ask for the file name and location to write. This is up to you to decide and keep track of.