

EE 448: Sensors and Actuators

Laboratory Module #1

Input and Output

Assigned: January 12, 2009

Due: 12:30 PM, Tuesday January 20, 2009 (In CSE 236)

Note: Labs will usually be due in-class on Mondays

Objectives

The objective of this Module is to use the I/O card to acquire data, generate a signal, and to implement simple discrete transfer functions in Labview. The first transfer function is zero order hold with delay (due to computation), and the second one transforms an input duty-factor to a pulse-width modulated signal with that duty factor. Once you complete this module, you should be familiar enough with the card and the techniques used to program with it to start using it in control applications.

You Will Need...

To complete this module, you will need

1. A USB-1208FS I/O card.
2. A USB cable
3. A Windows PC
4. Software for communicating with the I/O board: *InstaCal*, *Labview*, and the *Universal Library*.
5. An oscilloscope.

6. A function generator.
7. Wire.

All of this should be available in Sieg 233.

1 Connect and Test the I/O Card

First, you connect the I/O card to a computer and check that it is working properly. You will use the USB-1208FS for almost every laboratory module in this course, and the following procedure will be the first thing you do each time.

1. Connect the USB-1208FS card to computer using a USB cable.
2. Open *InstaCal*. A prompt to *add* the board will be displayed. If the prompt is not displayed, click the "add board" icon in the toolbar at the top of the screen and select the appropriate device from the list. The USB-1208FS card, labeled "Board 0" should appear on the PC board list.
3. Double click on card's icon to display the board configuration options. Select the "Flash LED" option. Verify that the LED on the board blinks (it is underneath the sticker). This indicates that the board is working properly. If you detach and re-attach the USB-1208FS from the computer, you may need to do this again.

2 Acquire an Analog Input Signal

The USB-1208FS card can be used to measure analog voltages and capture signals for data analysis. Using the "analog in" channel of the card, you will eventually measure the output of sensors attached to a control apparatus. For now, you will simply record some signals using the board's high sample rate input.

1. Configure the I/O card, the function generator and oscilloscope as shown in Figure 1. The signal will be input to CH0 IN. Note that CH0 and CH1 are a differential input, thus you also need to ground CH1, as shown in the figure. Otherwise the CH0 value will be relative to an unknown, floating signal.
2. Capture a 20 Hz sine wave from the function generator. To do this, set the function generator to generate the appropriate sine wave. For this and all sine waves in this module, be sure that you have an offset so that the voltages generated are all positive. Check with the oscilloscope that the appropriate signal is being generated.
3. To capture the signal, open the *Tracer DAQ* program and figure out how to acquire data from CH0 IN. Note that the default view is a strip chart. Change the view to "Scope View" for an oscilloscope-like view.

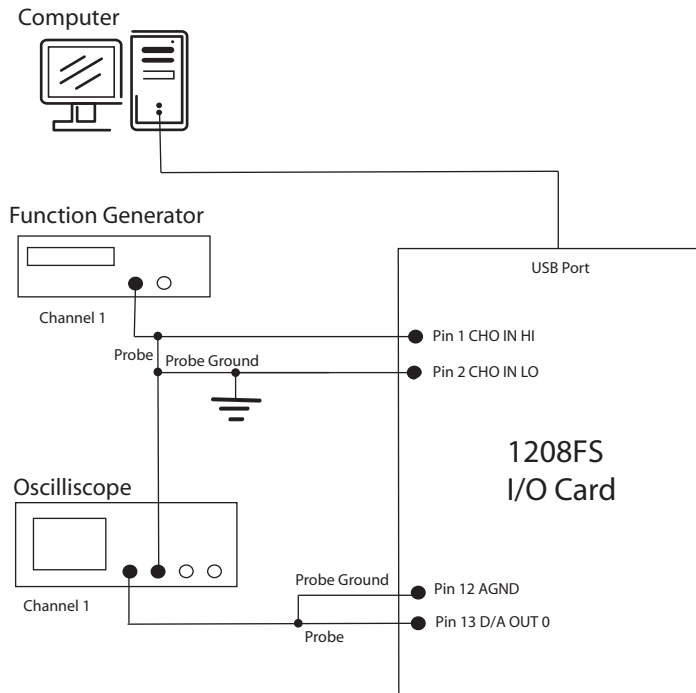


Figure 1: Hardware configuration for this Module. You will also need to use a second oscilloscope probe connected to the output of the function generator.

4. Change the frequency of the signal using the function generator to 40 Hz, 80 Hz, 160 Hz and so on. What happens to the measured version of the signal?

For Your Writeup: Describe the differences between the captured signal and the signal shown on the oscilloscope. Describe how the quality of the captured data changes as the frequency is increased.

Note: The function generators have a 50-ohm output, but sending them into a high-impedance input (not loaded to 50 ohms) means that you get double the voltage out than the function generator indicates. Believe the oscilloscope, not the function generator.

3 Use the I/O Card with Labview

In the previous section, we used the I/O card in buffered data mode. For example, when you acquire data at 5000 samples per second, it is not being sent to the computer at that rate. It is being buffered by the card and then sent to the computer at whatever rate the USB bus can handle. This is not particularly useful for control applications.

You will now investigate the use of the I/O card with which you use the arrival of data to trigger an action (in this case the sending of data). The amount of time it takes

between input samples is the fundamental limitation of this (or any) I/O card for control applications.

For most of the course, we will use Labview to communicate with the I/O card. The installation of Labview in Sieg 233 has libraries for communicating with the USB-1208FS.

Use the same hardware configuration as you did for the previous task. Set the function generator so that it produces a 1 Hz sine wave. Close *Tracer DAQ* and start Labview.

1. Download the `EE448mod1.vi` from the course wiki and open it. The code is essentially a discrete approximation of the transfer function $G(s) = 1$.

Look up all of the blocks in the Labview help. You will need to gain some facility with these and the other I/O functions, so make sure you know what these do.

2. Set the function generator to generate a 1 Hz sine wave. Connect channel one of the scope to the output of the card and channel two to the output of the generator. As before, the output of the generator should also be sent to the input of the card. Execute the above code and set the oscilloscope so that you can see one period of the sine wave. Try several frequencies to see how well the computer can track the sine wave.

For Your Writeup: Using the oscilloscope, zoom in on the I/O card output. Estimate the time between updates to the signal. What is the frequency of the above for loop?

For Your Writeup: Determine the delay of the system. That is, what is the phase lag of the generated signal from the original signal?

For Your Writeup: Make sure you get visual data from the Labview front panel that is useful for your report. You may need to export the data on the strip-chart into a format that works for your report.

4 Generate a PWM Signal

This week you will create a pulse-width modulated (PWM) signal that you will use next week to control a peltier module in the temperature plant. A PWM signal with period KT (where T is the minimum sample period of the DAQ interface, and K is some multiple of that) and duty factor p (where $0 < p < 1$) is a signal that is high for pKT seconds, then low for $(1-p)KT$ seconds, and then repeats (hint: look up PWM on the internet somewhere).

For this exercise, save the `.vi` with a new name and then modify it to take the input from CH0 IN, convert it to a duty factor, and then output a PWM signal with that duty factor. The DAQ interface has both analog and digital outputs, and here we are using the analog output.

To accomplish this, first notice the box `[ i ]` in the lower left of the while loop. This is the loop iteration, and increments by one each time through the loop. If you mod this by some appropriate number (the PWM period), you will get a sawtooth signal. One tooth corresponds to one PWM cycle. Dividing that signal by the PWM period will scale it onto a range 0 to 1. You can then compare the sawtooth signal with the input duty factor, and if the sawtooth is less than the duty factor then output low, and if greater or equal then output high (or vice versa).

In LabVIEW, the blocks you'll need are "Quotient & Remainder", "Divide", "Greater or Equal", and "Select". By the way, there is more than one way to do this exercise – these are just suggestions.

Assume that the duty factor (with range 0 to 1) is determined from the input voltage V_{in} by the formula $p = V_{in}/V_{max}$, where you specify the constant V_{max} . You will also specify the PWM period, KT , and should be able to explain the effect that different values of K have on your ability to create PWM signals with different duty factors. For example, what is the difference between using $K = 3$ versus $K = 100$?

To test your system, apply a constant voltage to the duty-factor input, e.g. the voltage corresponding to 25%, run your `.vi`, and set the oscilloscope to visualize one or two periods of the PWM signal. Then change the duty factor and see what happens (you can use a low freq square wave to alternate between two duty factors).

For Your Writeup: Include a figure showing your `.vi` diagram, and explain how it works.

For Your Writeup: Include a figure showing the PWM signal(s) you created for different values of p .

For Your Writeup: Explain the relationship between the PWM period, the resolution of the duty factor and the frequency of the system

5 Reminder: Report Guidelines

1. Title, group number, authors and lead author, percentage of work done by each.
2. Brief abstract stating objectives and main results.

3. Intro providing background, and goals of the module.
4. Discussion of relevant theory, equations.
5. All significant calculations, and not the insignificant ones.
6. Results, figures, tables, etc.
7. Discussion and conclusion.
8. List of references.