

LabVIEW 2: Mathscript and Data Acquisition

Reading :

	Reading	pages
Hands-On Introduction to LabVIEW by J. Essick	Chapter 3	All
	Chapter 4	All

Main focus :

- Check and be aware of aliasing.
- Learn how to use the *Mathscript* node, which is based on the very powerful Matlab language and uses the same syntax.
- Know how to read analog (and digital) voltages from your NI-Elvis board through the use of the DAQ assistant and the PCI-card inside your computers.
- Know how to write voltages to the NI-Elvis board.

Most relevant VI's :

- Waveform Simulator.vi
- Digital Oscilloscope (Express).vi
- Waveform Generator (Express).vi

Due Date: Friday, 11AM, Nov. 14

Assignments: Hand in a printout of the *Block Diagram* and *Front Panel*, where the *Front Panel* graphs should display relevant data. Make sure to include your name and date and most importantly sufficiently detailed comments. The comments on the *Block Diagram* should concern the details of how the program works, and why you chose to use certain LabVIEW language programming structures. The comments on the *Front Panel* can be used to talk about the meaning of the results.

1. Print (two) *Front Panels* of the **Waveform Simulator** in order to demonstrate the aliasing effect. (Don't forget comments explaining what one should see and how to interpret what the plots show!)

Intro to Problem 2 - Damped Harmonic Oscillator: A damped harmonic oscillator that starts from rest at a distance x_0 from equilibrium can be described by

$$x(t) = e^{-\gamma t} [A \cos(\omega t) + B \sin(\omega t)] \quad (1)$$

with

$$\omega = 2\pi f \quad A = x_0 \quad B = \frac{\gamma x_0}{\omega} \quad (2)$$

The phase space is spanned by the two independent quantities in this problem, e.g. x and $v = \dot{x}$. Thus, to obtain the phase space trajectory, plot $x(t)$ vs. $v(t)$.

Note that in mathscript, scalar arrays are generated using “:”, for example “t=start:step:stop”. Also note that to multiply or divide two equal-length scalar arrays element-by-element you need to use “.*” and “./”, respectively. For example, “tsquare=t.*t” or, equivalently, “tsquare=t.^2”.

2. Write a program called **Damped Oscillator** that produces and plots the time evolution of a damped harmonic oscillator and in addition shows the phase space trajectory. The user should be able to control the decay rate, frequency, and amplitude and also the number of samples and sampling frequency.

3. Use the DAQ assistant to write a **Dual Channel Oscilloscope**. Read in simultaneously a sine-wave signal generated by the NI-Elvis board (Func_Out) and the TTL-synchronization signal (Sync_Out) and display the result. You may trigger on either the digital (TTL) signal or, alternatively, on the analog voltage signal.

4. Write a **Random LED** program that turns on one of five LEDs (0-1-2-3-4) on the NI-Elvis breadboard with equal probability for a time interval of 100 milliseconds. Which LED is lit at a given time interval is random. The program continues to turn on and off LEDs until the user presses a stop button on the front panel. In addition, place five LEDs on the front panel that mirror the behavior of the physical LEDs.

Suggestion: Indicator Cluster and LED subVI: For later use it is advantageous to split off the LED part as a subVI from the random number generation and decision part. I would suggest to put the front-panel 5 output LEDs into an indicator cluster and save the cluster (similar to what you did in chapter 3, pg 25). Then create a **LED.vi** that takes 5 boolean inputs for the 5 LEDs and one boolean stop input. If the stop input is false, the 5 LED inputs determine the state of both the physical and on-screen LED’s. If the stop input is true, all LED’s are turned off. The VI output should be the LED-cluster that you created. Creating this SubVI also implies that you have to edit the LED.vi icon.

Mathscript: One way, but not the only way, to implement the decision which LED should be lit, is to use the *Random Number (0-1)* icon to generate an input to a *Mathscript Node*. In mathscript you can generate a boolean output variable simply by assigning the outcome of a true/false decision to a variable and putting this variable as a *Scalar⇒Boolean* output. For example, “ $LED0 = ((x > 0) \& (x \leq 1))$ ” would make *LED0* true, if $x \in (0, 1]$.

DAQ: [From Chapter 4] A Digital Input/Output (DIO) pin on an NI DAQ device is called a line. A typical DAQ device has 2 or 3 ports, where each port is composed of 8 lines. A particular line is specified as P port:line, so for example P0.3 is line 3 of port 0.

On the block diagram, place a DAQ Assistant and configure it as *Generate Signals⇒Digital Output⇒Line Output⇒port0/line0* and *Generation Mode⇒1 Sample (On Demand)*. With this configuration, when a Boolean TRUE (FALSE) is passed to DAQ Assistants data input, P0.0 will go HIGH (LOW). Then, add 4 additional channels in the same way using the *Add* button in the DAQ assistant. Since a DAQ device has N DIO lines (in our case N=5), data is formatted to accept an array of Boolean values with up to N elements. Hence, the **Build Array icon** (found in *Functions⇒Programming⇒Array*) is necessary to convert the five separate Boolean inputs to an array.

NI ELVIS: In its normal functioning, some versions of NI ELVIS block access to the port 0 DIO lines, and because these lines are used by the NI ELVIS to communicate with its associated DAQ device. In order to obtain control of these DIO lines, you must disable this communication by setting NI ELVISs front panel Communications switch to Bypass (as opposed to Normal), then run a VI called NI ELVIS-Enable Communications Bypass.vi, which found in *Functions*⇒*Instrument I/O*⇒*Instrument Drivers*⇒*NI ELVIS*⇒*Low Level NI ELVIS VIs*⇒*Digital I/O*. Construct the following block diagram and run it to obtain control of the port 0 DIO lines.



The port 0 lines will then appear on the NI ELVIS breadboard at the pins labeled DI 0 through DI 7 in the DIGITAL I/O section.

To turn on LEDs simply connect the output that is present at the DI pins to the LED pins that are pre-installed on the NI-Elvis board.

Have fun watching the LEDs blink. Can you discern any patterns?