Lab1B: Discrete-Time Signals in the Time-Domain using LabVIEW

Kevin Cao* Dhruvit Naik**

** Electrical and Computer Engineering Department, The College of New Jersey, (e-mail: caok1@tcnj.edu)}*
*** Electrical and Computer Engineering Department, The College of New Jersey, (e-mail: naikd2@tcnj.edu)}*

ABSTRACT: In this lab we re-familiarized ourselves with the LabVIEW software tool. The objective was to develop hand on experience in building Virtual instruments and to generate discrete time signals to carry out simple and complex operations involving sequences. In the lab, three sinusoidal waves were created. Two of the waves' frequency and amplitude were independent while, the third wave was dependent on the other two. This system was built three times in LabVIEW, using Express VI, Regular VI, and a hybrid with LabVIEW MathScript. In the end we were able to simulate the same results in all three setups.

1. INTRODUCTION

There are many software tools that help engineers design and analyze systems. Laboratory Virtual Instrument Engineering Workbench or LabVIEW for short is one of many software tools that do so. LabVIEW was developed by National Instruments and is written in the graphical language G. The program is most commonly used for data acquisition, instrument control, and industrial automation. LabVIEW is able to create programs called Virtual Instruments (VI). The programs are called Virtual Instruments because their appearance and operation imitate physical instruments. The program allows the user to create flowchart-like block diagrams. There are two windows that user can interface with, the front panel and block diagram window. The user mainly interacts with the program through the front panel.

The front panel contains a controls palette. The palette contains controls and indicators that are used on the front panel. The controls are organized into several categories that the user can access. The controls are used to supply inputs to the virtual instruments while indicators are used to display the virtual instruments' outputs. The controls may be knobs, push buttons, dials, sliders, or strings. The controls simulate instrument input devices and supply data to the block diagram of the virtual instruments. The indicators are graphs, charts, and status strings that are simulated from the output of the instrument devices. The indicators display the data the block diagram acquires or generates.

The block diagram window includes objects such as terminals, sub virtual instruments, functions, constants, structures, and wires that transfer data to other objects within the block diagram. When the front panel is created code is added to the graphical representation of functions to control the front panel objects. The block diagram window contains the graphical source code.

We have previously used LabVIEW in a past course. However, to ensure that we are capable of completing complex labs, we are creating some basic systems to refresh our memory of the software and its capabilities. In this particular lab we are constructing a system that contains three sinusoidal waves. Where the first two waves have independent frequency and amplitude that will affect that third wave's output. We constructed this system three times with different methods. One experiment used LabVIEW Regular, the second used LabVIEW Express, and the last was a hybrid of LabVIEW Express and MathScript.

2. METHODS

We began the lab by first familiarizing ourselves with LabVIEW and the tools it provided to its users. This was done by going through the "Lab 1: Getting Familiar with LabVIEW: Part 1" and "Lab 2: Getting Familiar with LabVIEW: Part 2" sections within the textbook. After familiarizing ourselves with LabVIEW we then constructed a system with in three different ways. We created a system that contained 3 sinusoidal wave generators. All of the waveforms were based on a sampling frequency of 8000 Hz with 256 samples. Two of the waveform generators were configured to output between 100 to 400 Hz, and the amplitude was set between 20 and 200. The third waveform generator's frequency and amplitude was dependent on the other two waveforms. The third signal's frequency is described by (1) and its amplitude is described by (2). The signals were then displayed on a single graph with a legend indicating which lines pertained to the signals.

$$f_3 = \left(\gcd(f_1, f_2) + mean(f_1, f_2)\right) \tag{1}$$

$$A_3 = A_1 + A_2 \tag{2}$$

3. RESULTS

After we got familiar with generating signals using LabVIEW and experience with modelling with Express, Regular, and Hybrid Programming. We were tasked to design our own system with given parameters. Each different method of modelling the problem is outlined in the following subsections.

3.1 Express Virtual Instruments

Express VIs allow users to abstract the process of defining parameters behind an interactive user interfaces. This allows for less wires and blocks in the block diagram, allowing for a cleaner looking diagram. To generate a signal in an Express VI, we used a “simulate signal” block. Placing the block on the diagram, brought up a configuration window which allowed for us to edit the different types of parameters for any given signal. Figure (1) is the Block Diagram and Figure (2) is the sub block for the calculation for frequency 3. Finally, Figure (3) is the Front Panel.

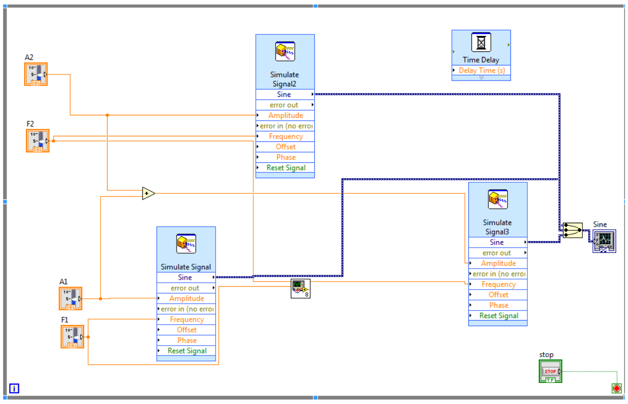

Fig. 1. Block Diagram for Express VI

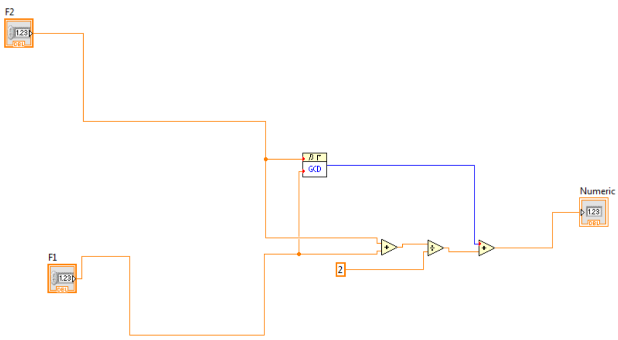

Fig. 2. Sub Block Diagram representing (1)

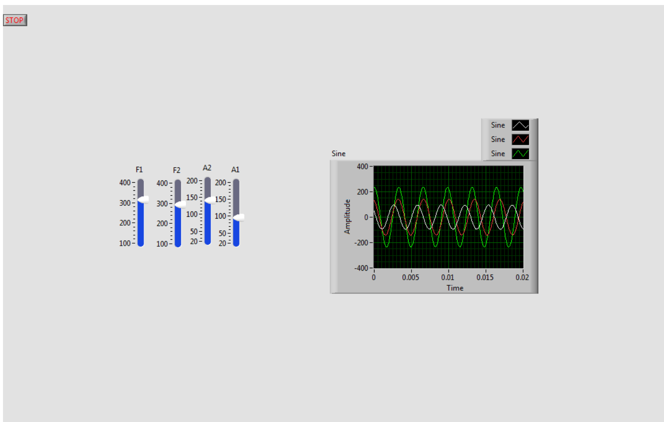

Fig. 3. Front Panel for Express VI.

3.2 Regular Virtual Instruments

Compared to Express VIs, Regular VIs require more wires. Also it requires parameter blocks for function generator blocks. This can be seen in Figure (4). We can still use sub blocks in Regular VIs. We used a sub block to encapsulate (1) and (2). Figure (5) shows this. Finally, Figure (6) is the Front Panel.

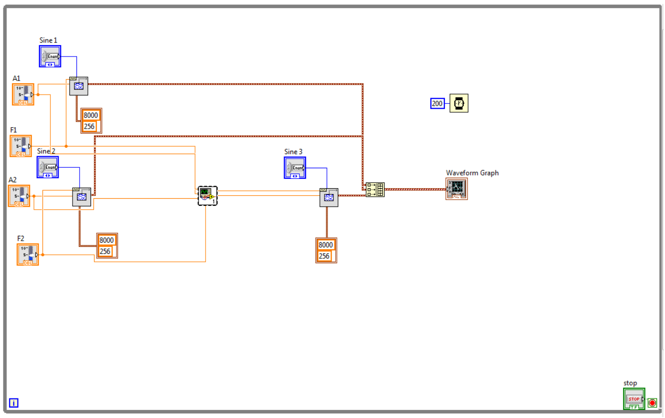

Fig. 4. Block Diagram for Regular VI

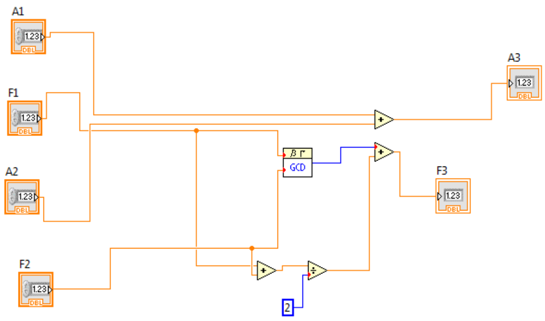

Fig. 5. Sub Block Diagram representing (1) & (2)

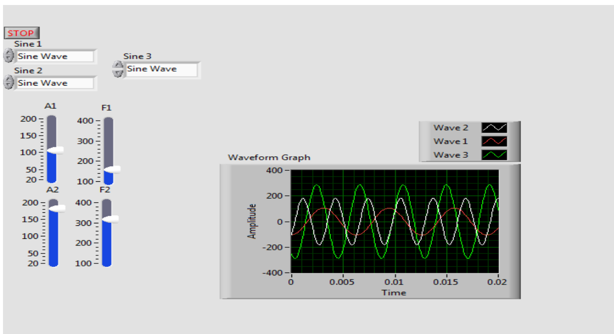

Fig. 6. Front Panel for Regular VI

3.3 Hybrid Programming

For Hybrid Programming we modified and redesigned the Express VI using a MathScript block. The version of LabVIEW included a MatLAB block rather than a MathScript block. Both blocks utilize a .m file. Numeric blocks are used to represent inputs and outputs. Below in Figure (7) shows the Front Panel of the Hybrid Programming VI. Below is the code snippet used in the MatLAB block. We had to cast F1 and F2 to an integer in order to use the 'gcd' function.

```
avg = (F1 + F2)/2;
g = gcd(round(F1), round(F2));
F3 = avg + g;
A3 = A1 + A2
```

In Figure (8) shows the Front Panel of the Hybrid Programming.

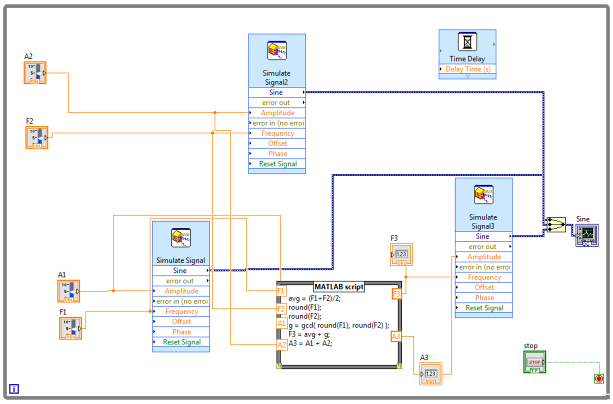

Fig. 7. Block Diagram for Hybrid Programming

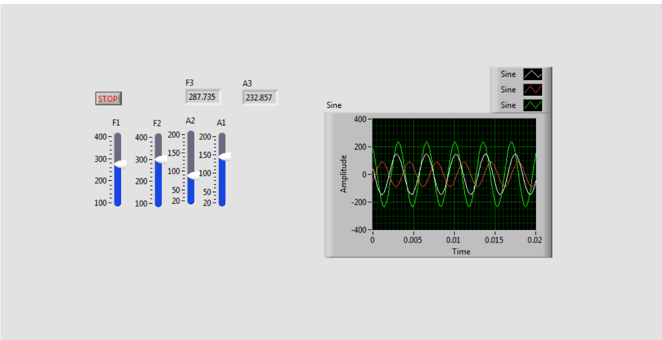

Fig. 8. Front Panel for Hybrid Programming

4. DISCUSSION

In all three experiments using LabVIEW Regular, Express, and a Hybrid it can be seen that the results are identical. All three systems contain three sinusoidal signals, where one of the signals is based off of the other two signals. Express VI's allows less wiring on the block diagram and allows users to access a GUI to adjust parameters quickly. Regular VI's do not offer this feature but requires physical blocks to edit inputs and other parameters. This allows users to fully capture the scope of the design without getting lost in the abstraction of Express VI's. The front panel for both Express and Regular are identically the same and function the same. We found using the Regular VI easier to understand and follow compared to the Express VI. Hybrid Programming can be a modification of either Regular and Express VI. The MathScript block is powerful tool that can reduce the number of blocks and perform the powerful functions available in the MatLAB library.

5. CONCLUSIONS

By the end of the lab we were able to successfully create the same system through the three different methods. We were able to create the system by using LabVIEW Regular, LabVIEW Express, and a hybrid of using LabVIEW Express and MathScript. By learning to use LabVIEW we have gained another tool to design and analyse systems. In addition to LabVIEW we also have Matlab to use to design and analyse systems. We gained sufficient experience in order to prepare ourselves to complete labs that require designing and analysing more complex systems. The lab has prepared us with various approaches to tackle a problem by using LabVIEW. We have three different options to model systems now either using LabVIEW Regular, Express, or a hybrid that uses MathScript.