LAB VIEW 7.1

A Report on How to Use the Hardware and Software

Presented to

Dr. Feresteh Fatehi
Electronics Computer Technology Professor
NC A&T State University

Prepared by

Bradley Strickland

To Begin Using LabVIEW:

Make sure that the power cables are connected.

Turn on the computer (press the power button).

Press Control + Alt + Delete to log on to the computer.

Log on to the computer with the following:

- Login name: **ectadmin**
- Login password: **fatehi**

On the desktop there is a **NI LabVIEW 7.1** icon, double click it to bring up the LabVIEW Welcome screen. Hit the **Continue** button to bring up the start menu.

At the LabVIEW start menu, you can create a **New** VI or **Open** an already created VI. You can also **Configure** LabVIEW or access the LabVIEW **Help** files.

When creating a new VI you single click on the **New** button to bring up a new VI from **Template** or create a **Blank** VI from scratch.

How to use LabVIEW:

LabVIEW programs are called virtual instruments (VIs).

Each VI contains three main parts:

- Front Panel – How the user interacts with the VI.
- Block Diagram – The code that controls the program.
- Icon/Connector – Means of connecting a VI to other VIs.

The Front Panel is used to interact with the user when the program is running. Users can control the program, change inputs, and see data updated in real time. The controls are used for inputs-adjusting a slide control to set an alarm value, turning a switch on or off, or stopping a program. Indicators are used as outputs. Thermometers, lights, and other indicators indicate values from the program. These may include data, program states, and other information.

Every front panel control or indicator has a corresponding terminal on the block diagram. When a VI is run, values from controls flow through the block diagram, where they are used in the functions on the diagram, and the results are passed into other functions or indicators.

The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

Controls and Functions Palettes

Use the **Controls** palette to place controls and indicators on the front panel. The **Controls** palette is available only on the front panel. Select **Window»Show Controls Palette** or right-click the front panel workspace to display the **Controls** palette. You also can display the **Controls** palette by right-clicking an open area on the front panel. Tack down the **Controls** palette by clicking the pushpin on the top left corner of the palette.

Use the **Functions** palette, to build the block diagram. The **Functions** palette is available only on the block diagram. Select **Window»Show Functions Palette** or right-click the block diagram workspace to display the **Functions** palette. You also can display the **Functions** palette by right-clicking an open area on the block diagram. Tack down the **Functions** palette by clicking the pushpin on the top left corner of the palette.

Tools Palette

If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the **Tools** palette. Toggle automatic tool selection by clicking the **Automatic Tool Selection** button in the **Tools** palette.

- Use the **Operating** tool to change the values of a control or select the text within a control.
- Use the **Positioning** tool to select, move, or resize objects. The Positioning tool changes shape when it moves over a corner of a resizable object.
- Use the **Labeling** tool to edit text and create free labels. The Labeling tool changes to a cursor when you create free labels.
- Use the **Wiring** tool to wire objects together on the block diagram.

Status Toolbar

- Click the **Run** button to run the VI. While the VI runs, the **Run** button appears with a black arrow if the VI is a top-level VI, meaning it has no callers and therefore is not a subVI.
- Click the **Continuous Run** button to run the VI until you abort or pause it. You also can click the button again to disable continuous running.
- While the VI runs, the **Abort Execution** button appears. Click this button to stop the VI immediately.

Note: Avoid using the **Abort Execution** button to stop a VI. Either let the VI complete its data flow or design a method to stop the VI programmatically. By doing so, the VI is at a known state. For example, place a button on the front panel that stops the VI when you click it.

- Click the **Pause** button to pause a running VI. When you click the **Pause** button, LabVIEW highlights on the block diagram the location where you paused execution. Click the **Pause** button again to continue running the VI.
- Select the **Text Settings** pull-down menu to change the font settings for the VI, including size, style, and color.
- Select the **Align Objects** pull-down menu to align objects along axes, including vertical, top edge, left, and so on.
- Select the **Distribute Objects** pull-down menu to space objects evenly, including gaps,

compression, and so on.

- Select the **Resize Objects** pull-down menu to change the width and height of front panel objects.

When you create an object on the Front Panel, a terminal will be created on the Block Diagram. These terminals give you access to the Front Panel objects from the Block Diagram code.

Each terminal contains useful information about the Front Panel object it corresponds to. For example, the color and symbols provide the data type. Double-precision, floating point numbers are represented with orange terminals and the letters DBL. Boolean terminals are green with TF lettering.

In general, orange terminals should wire to orange terminals, green to green, and so on. This is not a hard-and-fast rule; LabVIEW will allow a user to connect a blue terminal (integer value) to an orange terminal (fractional value), for example. But in most cases, look for a match in colors.

Controls have an arrow on the right side and have a thick border. Indicators have an arrow on the left and a thin border. Logic rules apply to wiring in LabVIEW: Each wire must have one (but only one) source (or control), and each wire may have multiple destinations (or indicators).

The program in this slide takes data from A and B and passes the values to both an Add function and a subtract function. The results are displayed on the appropriate indicators.

In addition to Front Panel terminals, the Block diagram contains functions. Each function may have multiple input and output terminals. Wiring to these terminals is an important part of LabVIEW programming.

Once you have some experience programming in LabVIEW, wiring will become easy. At first, you may need some assistance. Here are some tips to get you started:

- The wiring tool is used to wire to the nodes of the functions. When you "aim" with the wiring tool, aim with the end of the wire hanging from the spool. This is where the wire will be placed.

- As you move the wiring tool over functions, watch for the yellow tip strip. This will tell you the name of the terminal you are wiring to.
- As you move the wiring tool over a terminal, it will flash. This will help you identify where the wire will attach.
- For more help with the terminals, right-click on the function and select **Visible Items>>Terminals**. The function's picture will be pulled back to reveal the connection terminals. Notice the colors- these match the data types used by the front panel terminals.
- For additional help, select **Help>>Show Context Help**, or press **CTRL+H**. This will bring up the context help window. As you move your mouse over the function, this window will show you the function, terminals, and a brief help description. Use this with the other tools to help you as you wire.

Wiring is very flexible in LabVIEW. Experiment with keystroke and clicking combinations when wiring. Here are some of the most often used features:

- Single, double, and triple clicking a wire selects the wire for movement or deletion
- Clicking while wiring tacks down a bend in the wire
- Right-clicking or pressing Escape while wiring cancels the wiring operation

Do not worry about wire colors- LabVIEW will automatically select the right wire for each situation.

Automatically Wiring Objects

LabVIEW automatically wires objects as you place them on the block diagram. You also can automatically wire objects already on the block diagram. LabVIEW connects the terminals that best match and leaves terminals that do not match unconnected. As you move a selected object close to other objects on the block diagram, LabVIEW draws temporary wires to show you valid connections. When you release the mouse button to place the object on the block diagram, LabVIEW automatically connects the wires. Toggle automatic wiring by pressing the spacebar while you move an object using the Positioning tool. You can adjust the automatic wiring settings by selecting **Tools » Options** and selecting **Block Diagram** from the top pull-down menu.

Use the **Context Help** window and the **LabVIEW Help** to help you build and edit VIs. Refer to the **LabVIEW Help** and manuals for more information.

Context Help Window

To display the **Context Help** window, select **Help»Show Context Help** or press the <Ctrl-H> keys. When you move the cursor over front panel and block diagram objects, the **Context Help** window displays the icon for subVIs, functions, constants, controls, and indicators, with wires attached to each terminal. When you move the cursor over dialog box options, the **Context Help** window displays descriptions of those options. In the window, required connections are bold, recommended connections are plain text, and optional connections are dimmed or do not appear. Above is an example **Context Help** window.

Click the **Simple/Detailed Context Help** button located on the lower left corner of the **Context Help** window to change between simple and detailed context help. The simple mode emphasizes the important connections. Optional terminals are shown by wire stubs, informing you that other connections exist.

Click the **Lock Context Help** button to lock the current contents of the **Context Help** window. When the contents are locked, moving the cursor over another object does not change the contents of the window. To unlock the window, click the button again. You also can access this option from the **Help** menu.

Click the **More Help** button to display the corresponding topic in the **LabVIEW Help,** which describes the object in detail.

LabVIEW Help

You can access the **LabVIEW Help** either by clicking the **More Help** button in the **Context Help** window, selecting **Help»VI, Function, & How-To Help**, clicking the sentence **Click here for more help** in the **Context Help** window, or pressing <Ctrl-?>.

The **LabVIEW Help** contains detailed descriptions of most palettes, menus, tools, VIs, and functions. It also includes step-by-step instructions for using LabVIEW features and links to the *LabVIEW Tutorial*, PDF versions of all the LabVIEW manuals and Application Notes, and technical support resources on the National Instruments Web site.

How to Access and Run the LabVIEW Exercises:

Make sure that the power cables are connected.

Turn on the computer (press the power button).

Press Control + Alt + Delete to log on to the computer.

Log on to the computer with the following:

- Login name: **ectadmin**
- Login password: **fatehi**

There is a folder on the desktop called **LabVIEW Introduction Six Hours,** double click the folder to open it. Once the folder is opened you will see three other folders, a word document and a PDF file.

The exercises are located in the **Exercise** folder, double click the folder to open it. Once the folder is opened you can access all the exercises in one word file called **All Axercises – Six Hours**. If you want to access individual exercises, open the folder called **Individual Exercises**. In this folder there are several different exercises to choose from all in word format. When you are ready to create these exercises refer to the section on **To Begin Using LabVIEW** and the section **How to Use LabVIEW** located at the beginning of this report.

If you need help with the exercises there is a power point presentation located in the **Slides** folder found in the **LabVIEW Introduction Six Hours** folder. Also located in the **LabVIEW Introduction Six Hours** folder are the correct answers to the exercises which are located in the **VIs** folder.

How to Connect and Run Elvis:

Make sure that the power cables are connected.

(every one except the Bulk Power Supply for the DC Motor)

Make sure that the Elvis cable is connected to the back of the computer.

Turn on the computer (press the power button).

Press Control + Alt + Delete to log on to the computer.

Log on to the computer with the following:

- Login name: **ectadmin**
- Login password: **fatehi**

Turn on the power to the Elvis workstation, there is a switch in the back and on the front. Turn the one in the back on first before you turn on the one in the front and make sure that the switch in the front is in the off position before you turn on the switch in the back.

Once the Elvis workstation is connected and power is on then you can open the Elvis control panel.

On the desktop there is an **NI Elvis** icon, double click it to open it. If every thing is connected properly then it will configure it self. If it is not connected properly then there will be an error message. Find the problem and try again.

Once the Elvis control panel is configured and open you will have access to the Virtual Suite Components (Virtual Instruments), single click on any one of them to open them. Refer to Virtual Suite Components in this section of the report to find information about the instrument you want to use.

Hardware and Software Integrated Instruments

NI ELVIS's signal analyzers (Oscilloscope, Dynamic Signal Analyzer, Digital Multimeter, Bode analyzer, and Current-Voltage Analyzers) use the analog input lines of the DAQ board. Because a typical DAQ board has up to 8 differential analog input lines, it is possible to have an oscilloscope and a DSA running simultaneously. The speed of the NI ELVIS oscilloscope is a function of the maximum sampling speed of the DAQ board. In addition, all of the analyzers can leverage the digital triggering functionality of the DAQ board as well as the storage and data processing capabilities of the computer for a multitude of applications.

A waveform can be generated using two different methods. First, the built-in full-featured function generator can be controlled using knobs on the control panel or programmatically with a specially designed LabVIEW software suite included with NI ELVIS. A student therefore gets practice with both a traditional knob-type instrument and a modern virtual instrument with enhanced functionality. Because of this, NI ELVIS is capable of training students on multiple platforms. The analog output capabilities of the DAQ board form the basis of the NI ELVIS arbitrary waveform generator (ARB). Programmable analog output facilitates the fully functional ARB, which is also capable of reading a previously saved data file and outputting a corresponding waveform. Additionally, the workstation's built-in function generator contains advanced features such as frequency sweep and AM/FM modulation inputs. You can also use the function generator as a source for the impedance analyzer. With the impedance analyzer, measurements such as capacitance and inductance, which were not previously feasible with a DAQ-based system, are now possible.

NI ELVIS also has a very important capability that makes it a fully capable laboratory platform. The NI ELVIS workstation contains a programmable power supply along with its fixed voltage power supplies. The programmable supplies draw their power from an external source that is easily replaceable. Custom electronics within the workstation provide 12-bit resolution for the power supply.

NI ELVIS Software Features:

Open platform based on industry-standard LabVIEW software and NI DAQ devices

Combination of instrumentation, data acquisition, and prototype station

Complete suite of virtual instruments with context help

- Oscilloscope, Digital Multimeter, Function Generator, Variable Power Supplies, Bode Analyzer, Arbitrary Waveform Generator, Dynamic Signal Analyzer, Voltage/Current Analyzer, Digital Reader and Writer
- LabVIEW source code provided
- Ability to customize each instrument in the LabVIEW environment

Data storage in Excel, text, or HTML format

The NI ELVIS software suite provides the user two different ways to perform measurements using the workstation: the instrument launcher and the LabVIEW API. The instrument launcher provides easy access to all of NI ELVIS's 12 standard instruments as well as the suite's source code and NI ELVIS examples. By using the instrument selector the user is able to quickly open the instruments they would like and use the pre-built interface to control them. Additionally, the instrument selector provides direct access to the NI ELVIS source code and NI ELVIS instrument examples.

Alternately, the user can use the included LabVIEW API to build custom instruments or take the functionality of NI ELVIS instruments and incorporate it into another LabVIEW program. This is done simply by selecting the appropriate functions from the NI ELVIS Instrument palette, within LabVIEW, and placing them on to the block diagram.

Virtual Suite Components

Here is an overview of the virtual instruments that are available to use with the Elvis workstation.

Oscilloscope

The oscilloscope has all the functionality of a standard desktop instrument found in typical undergraduate laboratories. It provides sensitivity and position adjustment knobs along with a modifiable time base. It is also capable of providing RMS and peak-to-peak voltage measurements as well as the signal's frequency. Additionally, the NI ELVIS oscilloscope has the ability to choose a trigger source and mode. Depending on the NI DAQ device, the user can choose either a digital or analog hardware trigger. It is also possible to internally route signals from the function generator and impedance analyzer to the NI ELVIS oscilloscope. The NI ELVIS oscilloscope VI is also capable of storing measured data to the hard drive, thus providing the functionality of an advanced storage scope. This computer-based scope display has the ability to use cursors for accurate screen measurements. The sampling rate of the scope is determined by the maximum sampling speed of the DAQ device.

Features:
- 2 independent channels
- Digital and analog triggering
- Programmatic signal routing
- Frequency and amplitude measurements
- Cursors
- Data storage

Specifications:
- Maximum input bandwidth - 50 kHz*
- Maximum sampling rate per channel - 500 kHz*
- Vertical resolution - 12-bit or 16-bit *
- Range - 100 mV – 10 V full scale
- Input impedance - 1 GΩ*

*DAQ device dependent

Function Generator

The function generator is a powerful device providing functionality above and beyond a typical instrument found in undergraduate laboratories. Using the NI ELVIS function generator, you can select waveform shape (sine, square, triangle), amplitude, and frequency settings. In addition, the instrument offers DC offset setting, frequency sweep capabilities, and modulation. You can use an external modulation input, route it from the programmable power supply, or use the analog output of the DAQ device as the source. The instrument intelligently uses timing I/O on the DAQ board to lock the frequency of the function generator. Additionally, the main function generator controls (frequency, amplitude, and shape) are available for manual control on the hardware control panel.

Features:
- Manual and programmatic control
- Coarse and fine frequency control
- Ability to output an exact frequency
- Frequency sweep
- AM / FM modulation

Specifications:
- Frequency range - 5 Hz to 250 kHz in 5 ranges
- Waveform shapes - Sine, triangle, square
- Output amplitude - ±2.5 V
- DC offset - ±5.0 V
- Frequency set point accuracy - 3% of range, max

Software amplitude resolution - 8 bits

Digital Multimeter (DMM)

This commonly used instrument makes AC/DC voltage and current measurements. It can also check continuity of the circuit or perform diode testing. Access to the DMM instrument inputs is available on the workstation control panel through convenient banana-style probe inputs and through the prototype board.

Resistance
- Accuracy - 1%
- Range - 5 Ω to 3 MΩ in four ranges
- Test frequency - 120 Hz
- Test frequency voltage - 1 V p-p sine wave

DC and AC Voltage
- Accuracy - 0.3%
- DC Range - ±20 V in four ranges
- AC Range - ±14 Vrms in four ranges
- Input impedance - 1 MΩ

Current
- Range - 250 mA in two ranges
- DC accuracy - 0.25% ±3mA*
- AC accuracy - 0.25% ±3mA*
- Maximum common-mode voltage - ±42 V
- Common-mode rejection - 70 dB
- Voltage burden - 0.5 mV/mA
- Shunt resistance - 0.5 Ω

Capacitance
- Accuracy - 2%
- Range - 50 pF to 500 μF in three ranges
- Test frequency - 120 Hz or 950 Hz
- Test frequency voltage - 1 V p-p sine wave

Inductance
- Accuracy - 1%
- Range - 100 μH to 100 mH
- Test frequency - 950 Hz
- Test frequency voltage - 1 V p-p sine wave

Continuity
- Resistance threshold - 15 Ω max
- Test voltage - 3.89 V

* Proper null correction at the common-mode voltage can reduce the ±3mA offset error to 200μA of noise

Arbitrary Waveform Generator (ARB)

With this advanced level instrument, you can choose a variety of signal types, such as periodic, harmonic, modulated, or other waveforms. The ARB also provides the ability to load a data file to generate a stored waveform or use the built-in waveform editor to create your own custom waveform. Because a typical DAQ board has two analog output channels, you can generate two waveforms simultaneously. You also have the choice of a continuous output or a one-shot output. The output of the ARB can also be routed to the modulation input of the function generator. Note that the ARB only functions with NI DAQ boards that contain buffered analog outputs.

Features:
- One-shot or continuous generation
- Dual channel
- Waveform editor
- Programmatic signal routing to modulation inputs of function generator

Specifications:
- Amplitude - ±10 V, 12 or 16-bit
- DC Frequency range - one-tenth of the DAQ device update rate
- Output drive current - 25 mA max
- Output impedance - 1 Ω
- Slew rate - 1.5 V / μs

Variable Power Supplies

Two types of power supplies come with the NI ELVIS platform. First, three fixed power supplies of +5 V and ±15 V are included to power electronic circuits built on the prototyping board. Two variable power supplies are also built into the system, which can be controlled from a programmable LabVIEW interface or from the control panel of the workstation. These independently adjustable power supplies have opposite polarity. The power supplies provide sufficient current to power small electromechanical systems, such as DC motors.

Specifications:
±15 V supply:
- Accuracy - ±5%
- Output current fused at 500 mA**
- Ripple and noise - 1%
- Line regulation - 0.5%

Variable power supplies:
- Range - 0–12 V and -12–0 V**
- Ripple and noise - 0.25%
- Software controlled resolution - 7 bits
- Current limiting:
  - Positive supply - 0.5 V at 160 mA, 5 V at 275 mA, 12 V at 450 mA
  - Negative supply - 0.5 V at 130 mA, 5 V at 290 mA, 12 V at 450 mA

5 V supply:
- Accuracy - ±5%
- Output current fused at 2A
- Ripple and noise - 1%
- Line regulation - 0.5%

* Total current drawn from -15V supply and negative variable power supply cannot exceed 500mA

Dynamic Signal Analyzer (DSA)

The dynamic signal analyzer uses the analog inputs of the DAQ board to make measurements. This instrument uses extensive LabVIEW analysis functionality to build a fully functional DSA. The DSA offers various filtering and averaging options, while also displaying THD and SINAD measurements. In addition, data recorded with the DSA can be stored in tab-delimited format. Additionally, the DSA provides a variety of triggering options and cursors.

Specifications:

• Input range - ±10 V in four ranges
• Input resolution - 12-bits or 16-bits

Impedance Analyzer

The impedance analyzer provides phase and frequency polar plots to display measurements at various frequencies. A phasor diagram is visible to determine phase relationships between various signals. Using this virtual instrument, you can specify logarithmic and linear scales as well as specify the correct display quadrant and measurement frequency.

Specifications:
• Frequency range - 5 Hz – 35 kHz

Bode Analyzer

The power of virtual instrumentation is evident when you can use the same virtual set of hardware to build instruments that undergraduate labs can rarely afford. By simultaneously making use of the frequency sweep feature of the function generator and analog input of the DAQ board, you can build a fully functioning bode analyzer. You can also set the frequency range and choose between linear and logarithmic display scales. In addition, you have the option to set cursors and log data, which are usually only available on more expensive analyzers.

Specifications:
• Frequency and phase plots
• Frequency range and step control
• Logarithmic or linear frequency spacing
• Amplitude accuracy - 12 bits or 16 bits
• Frequency range - 5 Hz – 35 kHz
• Phase accuracy - 1°

Current-Voltage Analyzers

These instruments make use of the programmable power supplies available with NI ELVIS. These instruments also offers full flexibility in setting parameters such as voltage and current ranges, linear and logarithmic scales, cursors, and the ability to save data to file.

Two wire:

With the 2-wire curve analyzer, you can conduct diode testing and view I-V curves.

Specifications:
• Current range - ±10 mA
• Voltage sweep range - ±10 V

Three wire:

This instrument is designed to function with NPN BJT transistors and familiarize students with curve analyzers.

Specifications:
• Minimum base current increment - 15 uA
• Maximum collector current - 10 mA
• Maximum collector voltage - 10 V

Digital Writer

The digital writer is a useful tool in analyzing digital circuits and providing control signals to circuits built on the prototype board. It provides TTL logic write access to the eight digital input channels on the NI ELVIS prototype board. Additionally, you can select from testing patterns such as walking 1's, alternating 1's and 0's, and ramp. The digital writer also provides the user the ability to input the value to write to the digital channels in binary, hexadecimal, decimal, and octal.

Specifications:
• Digital output resolution 8-bits

Digital Reader

The NI ELVIS digital reader provides the ability to easily view the status of the digital input channels. The virtual instrument also provides the numeric representation of the digital channels in a variety of formats.

Specifications:
• Digital input resolution - 8-bits

How to Run Quanser (DC motor module) for Elvis:

1. Introduction

The Quanser – NI Engineering Trainer is a versatile and powerful training tool.
Amongst its many capabilities, the QNET series of trainers allow for PC based control using the LABVIEW programming language, a National Instruments E-Series data acquisition card and an ELVIS workstation. The QNET allows for a scalable laboratory setup utilizing the ELVIS workstation platform.

2. QNET – Plug & Play

The QNET modules are designed to simply plug right into the ELVIS prototype board slot and be operational instantaneously. The following steps will ensure proper operation of your QNET module. **Make sure that the ELVIS prototyping board power switch is OFF and the COMMUNICATIONS switch is in BYPASS mode!**

Step 1: Plug the QNET into the ELVIS

Step 2: Power On the prototype board
Once the QNET is plugged into the ELVIS workstation, power on the prototype board. At this point you should have **3 LEDs on**. Check to make sure the +15v, -15v and +5v LED are lit. This ensures the QNET module is properly connected to the ELVIS

Step 3: Choose and Start the appropriate Controller, Press RUN
***The controller should be running before you plug in the Bulk power (external power supply).**

Step 4: Plug in the Bulk Power Supply
The final step is to power the on board amplifier. The +B (Bulk) led should now be lit. This indicated there is power to amplify the control signals.
***Note: If there are any voltage being applied by the analog outputs (AO 0, AO 1), the signals will be amplified and delivered to the motor! Extra care should be taken when connecting the bulk power to QNET. If the motor begins to turn once bulk power is applied, disconnect the bulk power immediately!**

3. Labview QNET – Supplied Controllers

All of the LabVIEW 7.1 Controlers are located in the **Quanser** folder on the desktop, in the **QNET.mx Release 1.0** folder. The Controllers for LabVIEW 7 express are located in the **QNET for LabVIEW 7.0** folder on the desktop, in the **QNET Release 2.2** folder. The controllers for LabVIEW 7 express will run in LabVIEW 7.1.

Here are some of the controllers found in the **QNET.mx Release 1.0** folder and a brief description of how they manipulate the DC Motor.

QNET_Position.vi (QNET-DC Motor)
This controller will position the camera at a given setpoint (multiplied by a square wave of 0.5 Hz). The controller is a standard PD controller with tunable P and D gains. The setpoint is also adjustable from the control panel.

QNET_ Haptic _Knob.vi (QNET-DC Motor)
This is a haptic demonstration controller. The user can set the knob step size as well as the stiffness and detent of the controller. With this controller, the DC motor behaves like a mechanical knob that can be programmed to behave as desired.

4. Conclusion

The QNET DC Motor Control Trainer has been designed to illustrate the fundamentals of DC motor control. It can quickly and easily be configured for a range of hardware-in-the-loop experiments including system ID, speed and position control.