NUST School of Electrical Engineering and Computer Science

Faculty Member: Dr. Neelma Naz Date: February 27, 2023

Semester: 6th Section: C

Department of Electrical

Engineering Control Systems

EE-379

LAB 4: Data Acquisition in LabVIEW

Student name	Reg. No.	Lab Report Marks / 10	Viva Marks / 5	Total /15
M Ahamd Mohsin	**333060**			
Imran Haider	**332569**			
Zafar Azhar	**340908**			

NUST School of Electrical Engineering and Computer Science

Lab 04: Data Acquisition and Control in LabVIEW and Simulink

1. Objectives

- Introduction to data acquisition
- Learn data acquisition in LabVIEW using NI ELVIS

2. Data Acquisition

Data acquisition is the process of sampling real world physical signals and converting them into digital numeric values that can be manipulated by a computer.

For example, a physical signal is the humidity of a room as a function of time. Most of the humidity sensors generate an analog voltage/current based on the current value of the humidity. However, computers or microcontrollers cannot works with analog values. A data acquisition system will sample this analog signal and convert each sample into a digital numeric value. This would allow digital machines like a computer to analyze the physical signal.

Some of the other physical signals are temperature, pressure, flow rate of a fluid in a pipe, motor speed, pendulum angle, wind speed, etc.

Data acquisition devices for modern computers

Most data acquisition hardware that interface with computers are based on a PCI, USB, or a similar interface. They are usually compatible with the popular engineering software like MATLAB, LabVIEW, Simulink, etc. This allows us to easily interface real physical signals in our simulations.

By definition, data acquisition is only converting analog signals to digital. However, if we want to interface our simulation software (e.g. MATLAB, Simulink, etc.) with a physical system (e.g. DC motor), we would also like to be able to generate inputs for that system. The inputs for physical systems are also often analog in nature. Moreover, some

3.1. Instructions to connect and power up the NI ELVIS

i. Without turning on the power adapters, connect the power adapter cables at point 5 and point 9. See the figure and the table for details.
ii. Connect a USB cable at point 6 of the NI ELVIS board. Connect the

other end to your PC.

iii. Turn on the supply for power adapters.
iv. Turn on the switch near point 5 of the NI ELVIS. The switch is on the rear panel of the NI ELVIS.
v. Turn on the switch at point 2 of the NI ELVIS. If the DC motor starts running, turn the board off and inform the lab staff.
vi. The power (point 3) and ready (point 4) LEDs of the NI ELVIS should be on now.
vii. The LEDs on the DC motor board at point 8 will also turn on.
viii. The NI ELVIS board and the DC motor are now ready to use.

3.2. Schematic of DC motor signals and NI ELVIS channels

The figure below shows which channels of the NI ELVIS board are used by the DC motor and for what signal.

i. Analog output channel AO#0 is used to generate the voltage to be applied to the motor.
ii. Analog input channel AI#1 is used to acquire the armature current of the motor.
iii. Digital input channel DI#0 is connected to the encoder and used to determine the position of the motor.
iv. Analog input channel AI#4 is used to acquire the speed of the motor.

3. Data Acquisition using LabVIEW

We have given a brief introduction to the NI ELVIS and QNET DC motor. Now we will see how we can use LabVIEW for data acquisition. In this section we will only show the data acquisition for the DC motor. However, the data acquisition for any device is similar in LabVIEW especially if the DAQ is manufactured by National instruments (NI).

DAQmx is the LabVIEW driver for the programming of data acquisition hardware in LabVIEW. It is a single programming interface for programming analog input, analog output, digital I/O, & counters on hundreds of multifunction DAQ hardware devices. The following are the steps for data Acquisition using DAQmx:

1. Create a virtual channel and task using the **NI-DAQmx Create Virtual Channel VI**
2. Write or Read data using **DAQmx Write** or **DAQmx Read VI** respectively.

4.1. Generating voltage for DC motor using analog output channel AO#0

i. Open LabVIEW and create a new VI by going to **File » New VI**
ii. Right-click on the block diagram and select **Measurement I/O**
iii. Drop the "**NI-DAQmx Create Channel** VI" to the block diagram and choose "AO voltage".

i. Right click on the physical channel input of "NI-DAQmx Create Virtual Channel VI" and create constant. Here you will select the channel you want to use. According to the schematic in QNET User Manual, the DC motor voltage is generated at analog output channel 0. Select this channel from the drop down menu of the Physical channel constant.

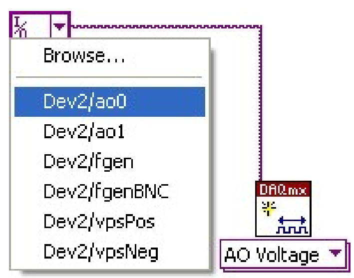

Figure 1

ii. According to the QNET User Manual, the output of the AO#0 is connected to an amplifier with a gain of 2.3 and the maximum and minimum voltage for the DC motor are +24 and -24, respectively. Therefore, we should set the output signal range from (-24/2.3) to (24/2.3). However, to be on the safe side, we will keep the output signal range from -8 to 8 volts.
iii. Create constants for maximum and minimum value and enter 8 and -8 respectively.

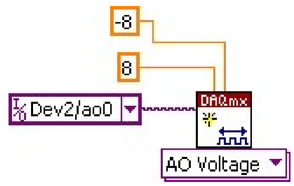

Figure 2

iv. Right-click on the block diagram and select **Measurement I/O**

v. Drop the **"NI-DAQmx Write"** to the block diagram and select Analog-DBL- 1Channel-1Sample

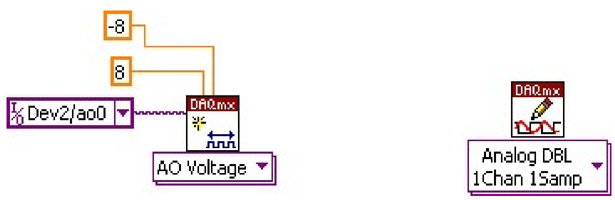

Figure 3

i. Now go to the front panel and insert a pointer slide from Numeric Control palette. Name this input block as "voltage input". Right click on the pointer slide and open its properties. Go to scale tab and enter a range of -10 to 10.

ii. Now go to block diagram and connect the voltage input block to the data input of the DAQmx Write.

i. The VI that we have created will only generate a single sample for the output voltage. To continuously generate the sample, we have to insert a simulation loop. Insert a Control and Simulation loop outside the DAQ to execute the simulation diagram until the Control & Simulation Loop reaches the simulation final time by right-clicking on Block Diagram and using path shown in figure:

ii. Insert the Control and Simulation loop outside the DAQmx Write

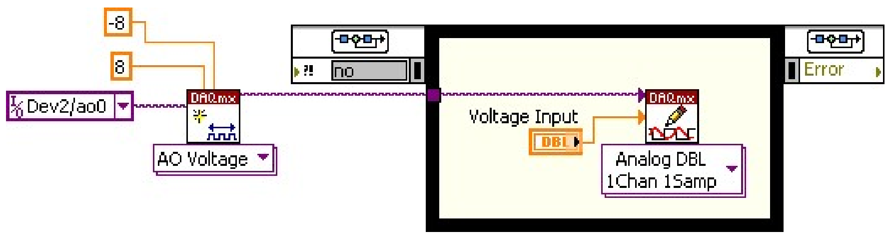

Figure 4

i. When setting the voltage with the slider input, we are not considering the gain of the amplifier between AO#0 and the DC motor. To accommodate for this gain we insert a reciprocal of this gain in our VI. In the block diagram, insert a gain block by going to Functions Palette >> Control Design and Simulation >> Simulation >> Signal Arithmetic >> Gain. Set the gain to 1/2.3 = 0.43

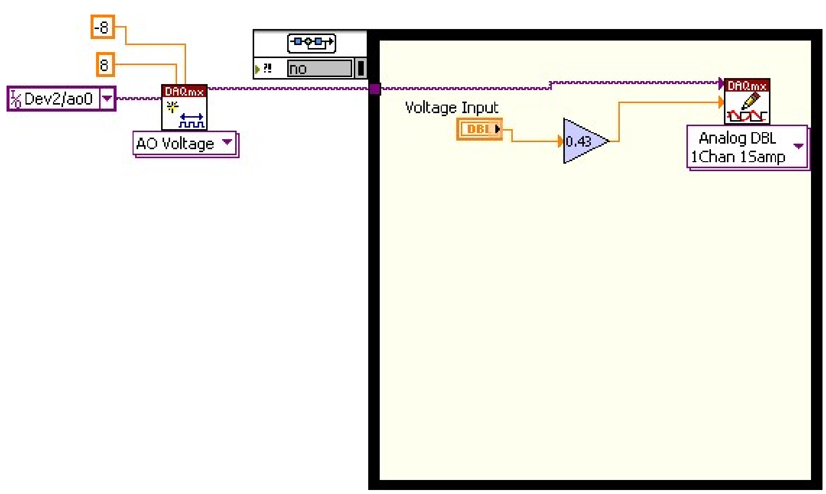

Figure 5

ii. This completes the voltage generation part of the VI. Run the VI and change the slider to apply different voltages to the motor. The speed of the motor should change.

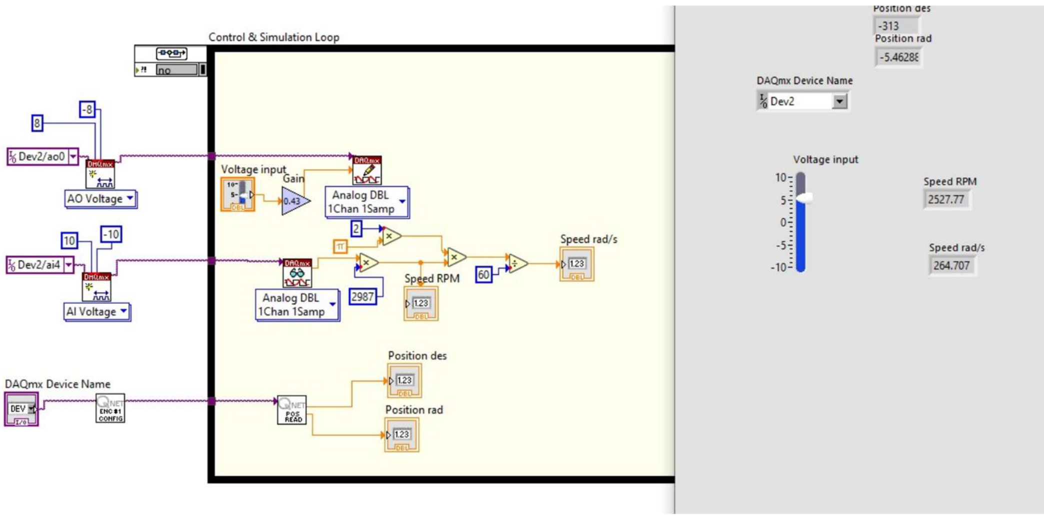

Figure 6

i. In the block diagram, connect the data output of the "DAQmx Write" to the numeric indicator block

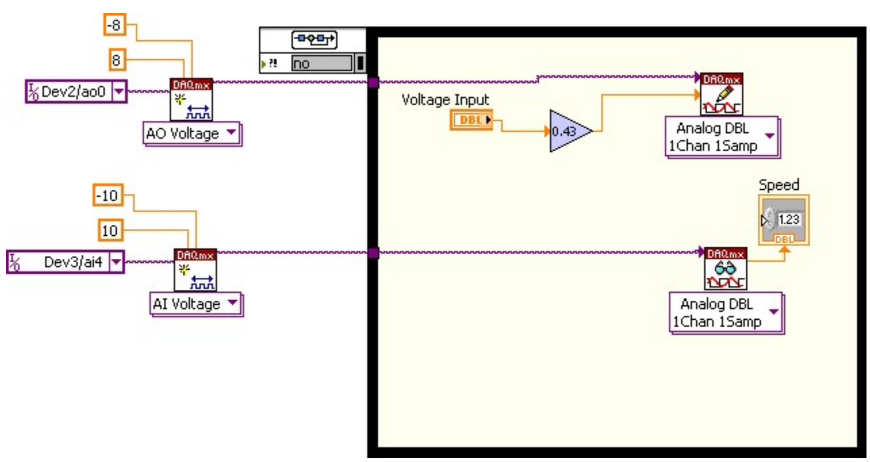

Figure 7

ii. Run the VI and apply some voltage to the motor. If the motor is moving you should see some non-zero value in the numeric indicator for the speed.

iii. According to the QNET User Manual, the tachometer calibration is 2987 RPM/volts. There 2987 rotations in a minute, the tachometer generates 1 volt. Since we are acquiring the voltage from the tachometer, we are not seeing the speed of the motor in any standard units on our numeric indicator. To accommodate for the calibration of the tachometer, we will multiply the acquired voltage by a factor of 2987. This will give us the speed of the motor in RPM.

iv. In the block diagram, insert a multiplier block to scale the acquired voltage. Rename the numeric indicator as “Speed (RPM)”.

v. Test your VI again. Now you should be seeing the speed in RPM.

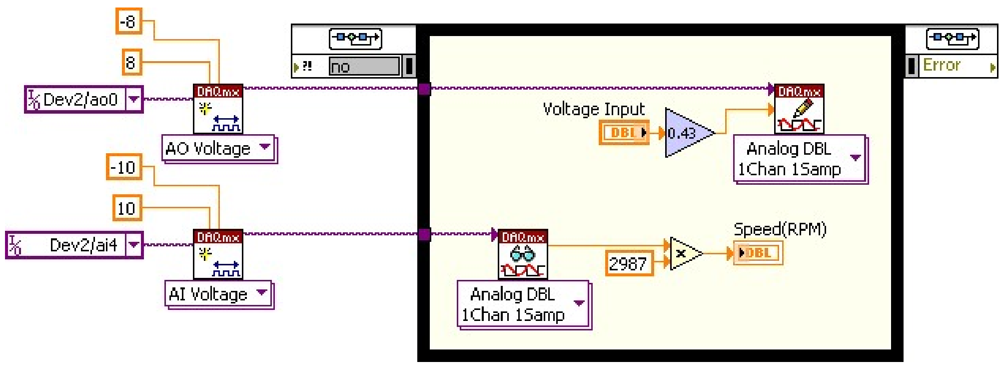

Figure 8

i. Since in the earlier labs we derived the transfer function for motor speed in rads/sec, we would also like to have the speed in these units. Insert another numeric indicator in your VI and name it “Speed rads/sec”. Scale the speed in RPM to find the speed in rads/sec. The formula is given below:

1 RPM = 2*pi/60 rads/sec

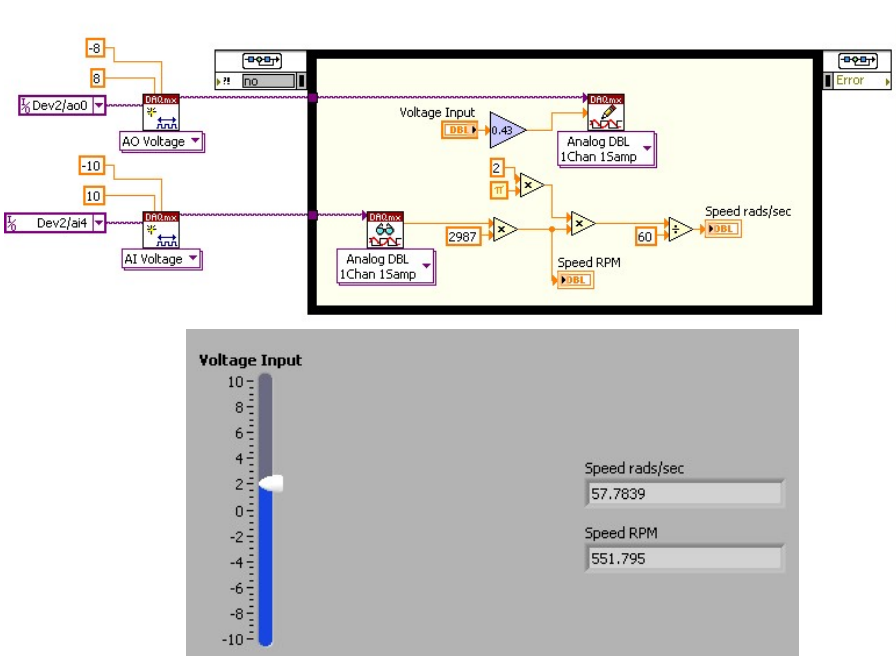

Figure 9

ii. Test your VI again. Now you should be able to see the speed in both units.

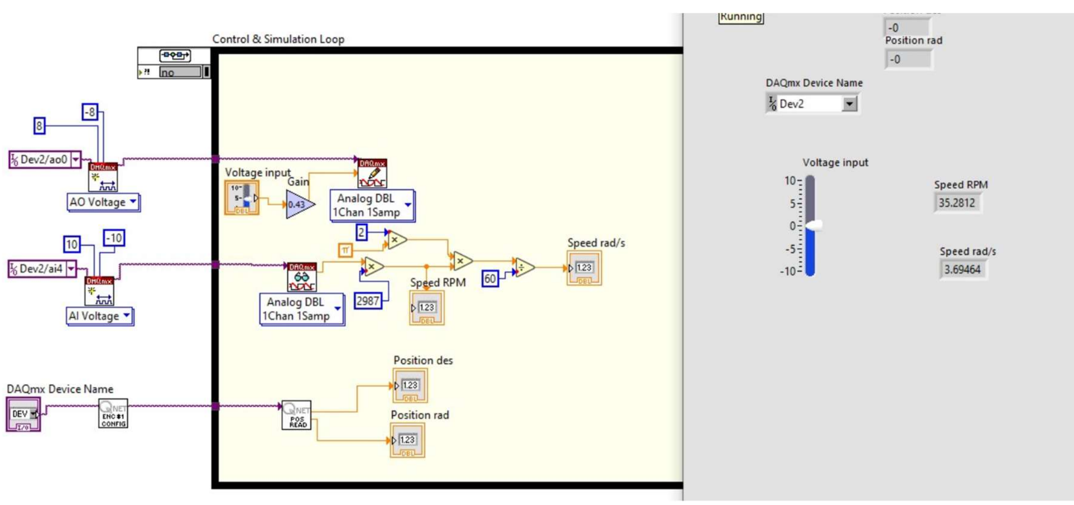

Figure 10

Exercise 1: A simple test of our data acquisition VI

We have created a VI for data acquisition. However, we haven't tested if the signals that are being generated are of the correct value or not. Similarly, we haven't tested if the acquired signals are of the correct value or not.

Stop the VI. Adjust the voltage input to zero volts. Run the VI. Rotate the motor by approximately 90 degrees. See whether the VI shows the same value. Repeat this for various other angles like 180, 360, -90 and -180 degrees etc.

Attach a small marker to one side of the rotating disc attached with the motor. Use a small piece of white tape or something similar. Run the VI and apply a small voltage so that the motor is rotating at around 30 to 60 RPM according to your front panel indicators. Now using a stop watch or time and by looking at the motor itself, try to count how many rotations the motor is making. It should match with the one shown in your front panel.

In our hardware setup, it is difficult to measure the applied voltage. So we will be unable to verify the voltage generation part of the data acquisition directly. However, we can tell you that the motor speed is approximately 158 rads/sec when a voltage of 5 volts is applied. Apply 5 volts to the motor and verify if this really is the case.

Exercise 2: Data Acquisition for Inverted Pendulum

The figure below shows which channels of the NI ELVIS board are used by the Inverted Pendulum and for what signal.

i. Analog output channel AO#0 is used to generate the voltage to be applied to the motor.
ii. Analog input channel AI#0 is used to acquire the armature current of the motor.
iii. Digital input channel DI#0 is connected to the Motor encoder 0 and used to determine the position of the Motor Arm.
iv. Digital input channel DI#1 is connected to the encoder 1 and used to determine the position of the Pendulum Link.
v. Analog input channel AI#4 is used to acquire the speed of the motor.

Note: You can use Pendulum_Position.vi present on the desktop of your computer

Connect Rotary Inverted Pendulum system to your PC using same procedure you used to connect the DC Motor System and Create a VI which:

- Selects device number
- Applies Voltage to the Pendulum's Motor
- Measures Arm Angle in Degree and rad/sec
- Measures Link Angle in Degree and rad/sec
- Measures Motor speed in RPM and rad/sec

Your VI should look similar to the VI shown in the figure below:

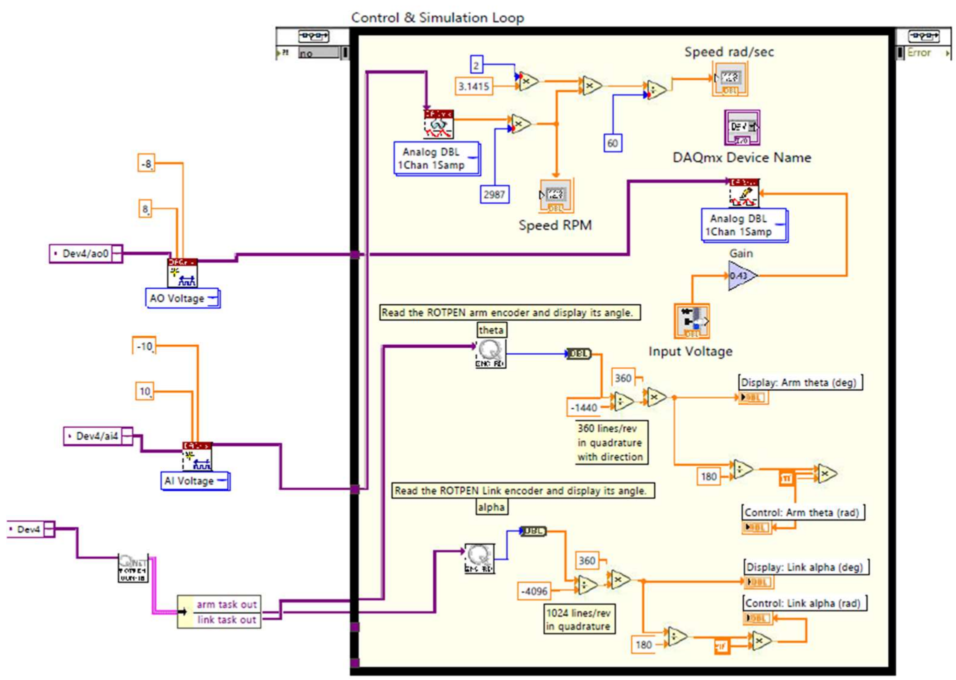

Figure 11

Conclusion:

In this lab we gained a thorough understanding of data acquisition LabVIEW using NI ELVIS. The experiments were performed in the lab and results were duly observed. The speed of the DC motor was controlled using LabView. Afterwards another experiment was performed using the Rotary Inverted Pendulum.