QNET-010 DCMCT

Quanser Engineering Trainer for NI-ELVIS

QNET DC Motor Control Trainer

Student Manual

Table of Contents

1. Introduction

This manual contains experimental procedures and lab exercises for the QNET DC Motor Control Trainer (DCMCT). The DCMCT is depicted in Figure 1 and the hardware of the device is explained in Reference QNET User Manual.

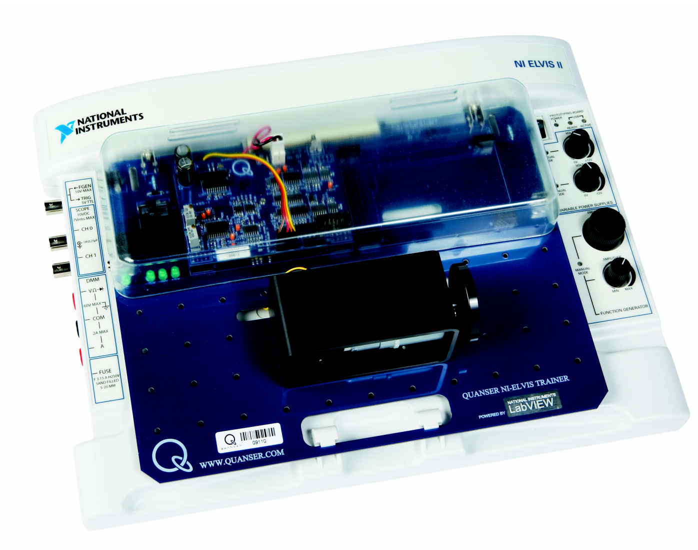

Figure 1: QNET DC motor control trainer on ELVIS II.

The prerequisites to run the LabVIEW Virtual Instruments (VIs) for the DCMCT are listed in Section 2 and described in Section 3. The in-lab procedures are given in Section 4 and split into three sections: modeling, speed control, and position control. In Section 4.1, the bumptest method is used to find the model parameters of the DC motor. This model is compared with the measured response by running the simulation and actual system in parallel. The model parameters are then tuned for a better fit. In Section 4.2, a PI compensator is used to control speed of the motor. This section includes exercises that demonstrates the effect of proportional and integral control, designing PI gains to meet specifications, set-point weight, and tracking a triangular wave. In Section 4.3, a PID compensator is used to control the position of motor. The effects of using only a PD controller is investigated and a PD controller is designed for certain time-domain requirements. How the system handles disturbances when using PD and PID compensators is then investigated. The exercises are given within the lab procedures and labeled “**Exercise”**. In that case, enter your answer in the exercises number in the corresponding section.

2. Prerequisites

The following system is required to run the QNET DCMCT virtual instruments:

- ✔ PC equipped with either:
  - ✔ NI-ELVIS I and an NI E-Series or M-Series DAQ card.
  - ✔ NI ELVIS II
- ✔ Quanser Engineering Trainer (QNET) module.
- ✔ LabVIEW 8.6.1 with the following add-ons:
  - ✔ DAQmx
  - ✔ Control Design and Simulation Module
  - ✔ *When using ELVIS II*: ELVISmx installed for required drivers.
  - ✔ *When using ELVIS I*: ELVIS CD 3.0.1 or later installed.

If these are not all installed then the VI will not be able to run! Please make sure all the software and hardware components are installed. If an issue arises, then see the troubleshooting section in Reference QNET User Manual.

3. DCMCT Virtual Instruments

3.1. Summary

Table 1 below lists and describes the DCMCT LabVIEW VIs supplied with the **QNET CD**.

VI	*Description*
QNET_DCMCT_Modeling.vi	Run DC motor in open-loop.
QNET_DCMCT_Speed_Control.vi	Control speed of DC motor load using a proportional-integral (PI) compensator.
QNET_DCMCT_Position_Control.vi	Control position of DC motor load using a proportional-integral-derivative (PID) compensator.

Table 1: DCMCT VIs supplied with the QNET CD.

3.2. Description

3.2.1. Modeling

The DCMCT Modeling VI, shown in Figure 2 and Figure 3, runs the DC motor in open-loop and plots the corresponding speed and input voltage responses. This VI can be used to take speed and voltage measurements of the responses, as illustrated in Figure 3, and runs a simulation of the DC motor in parallel. Table 2 lists and describes the main elements of the QNET-DCMCT Modeling virtual instrument front panel. Every element is uniquely identified through an ID number and located in Figure 2.

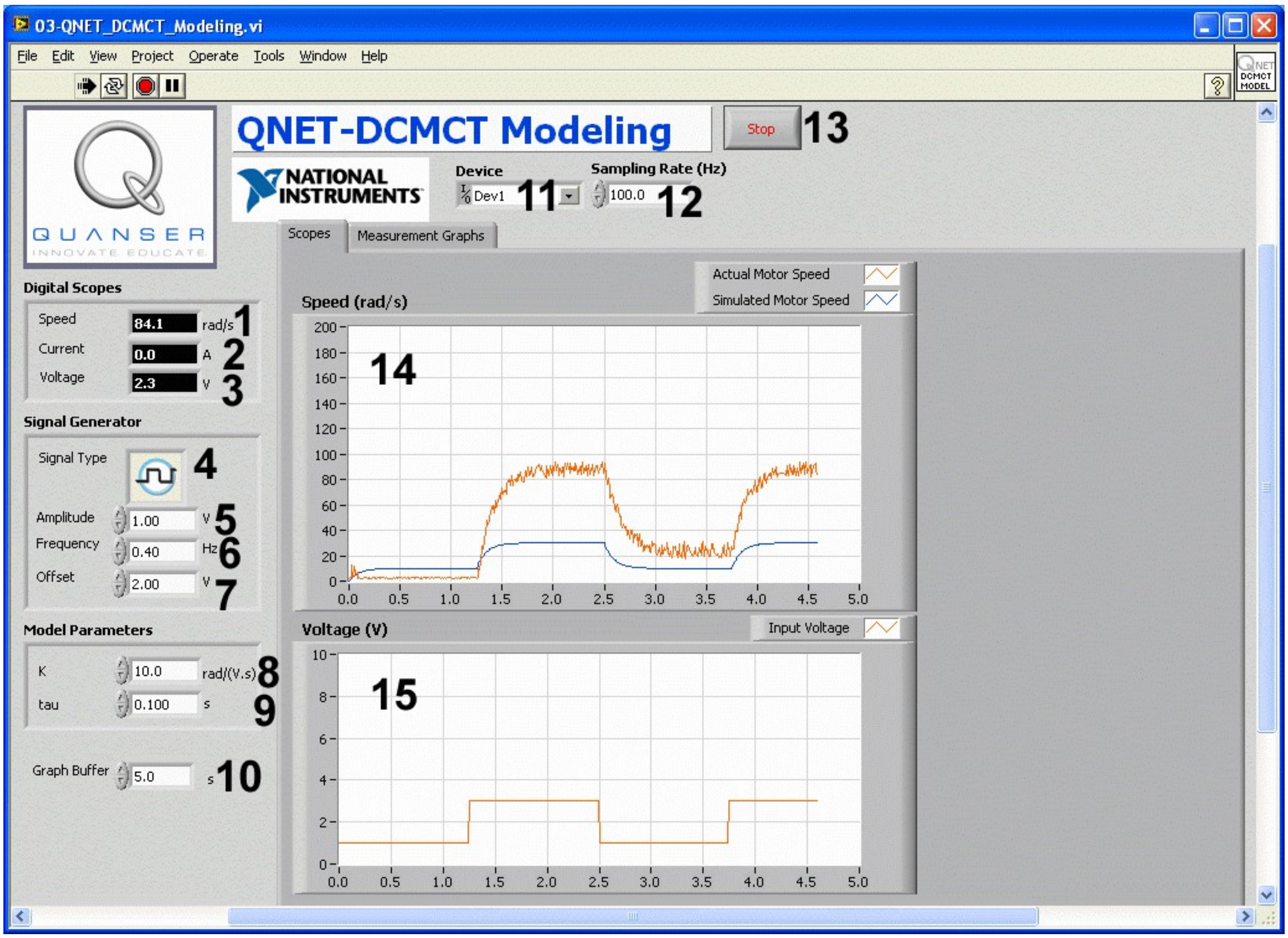

Figure 2: QNET-DCMCT Modeling virtual instrument.

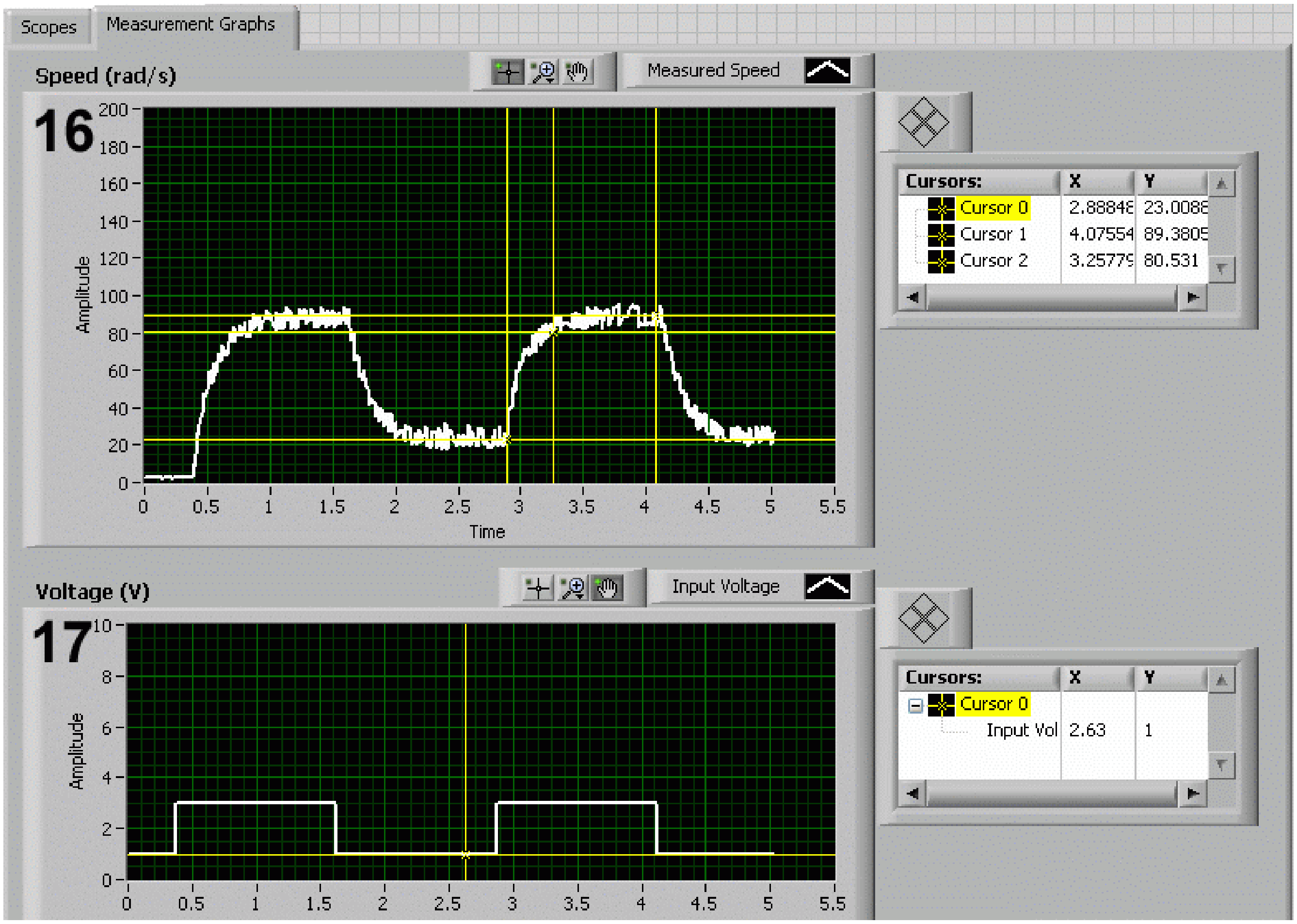

Figure 3: QNET DCMCT Modeling VI: "Measurement Graphs" tab selected.

ID #	Label	Parameter	Description	Unit
1	Speed	ω_m	Motor output speed numeric display.	rad/s
2	Current	I_m	Motor armature current numeric display.	A
3	Voltage	V_m	Motor input voltage numeric display.	V
4	Signal Type		Type of signal generated for the input voltage signal.	
5	Amplitude		Generated signal amplitude input box.	V
6	Frequency		Generated signal frequency input box.	Hz
7	Offset		Generated signal offset input box.	V
8	K	K	Motor model steady-state gain input box.	rad/(V.s)
9	tau	τ	Motor model time constant input box.	s
10	Graph Buffer		Buffer length of graph data.	s

11	Device		Selects the NI DAQ device.	
12	Sampling Rate		Sets the sampling rate of the VI.	Hz
13	Stop		Stops the LabVIEW VI from running.	
14	Scopes: Speed	ω_m	Scope with measured (in red) and simulated (in blue) motor speeds.	rad/s
15	Scopes: Voltage	V_m	Scope with applied motor voltage (in red).	V
16	Measurement Graphs: Speed	ω_m	Graph displays buffered measured motor speed after VI is stopped.	rad/s
17	Measurement Graphs: Voltage	V_m	Graph displays buffered input voltage used after VI is stopped.	V

Table 2: Nomenclature of QNET-DCMCT Modeling VI

3.2.2. Speed Control

In the QNET DCMCT Speed Control VI, a proportional-integral compensator is used to control the speed of the motor. The PI control also includes set-point weight. Table 3 lists and describes the main elements of the QNET-DCMCT Speed Control virtual instrument user interface. Every element is uniquely identified through an ID number and located in Figure 4.

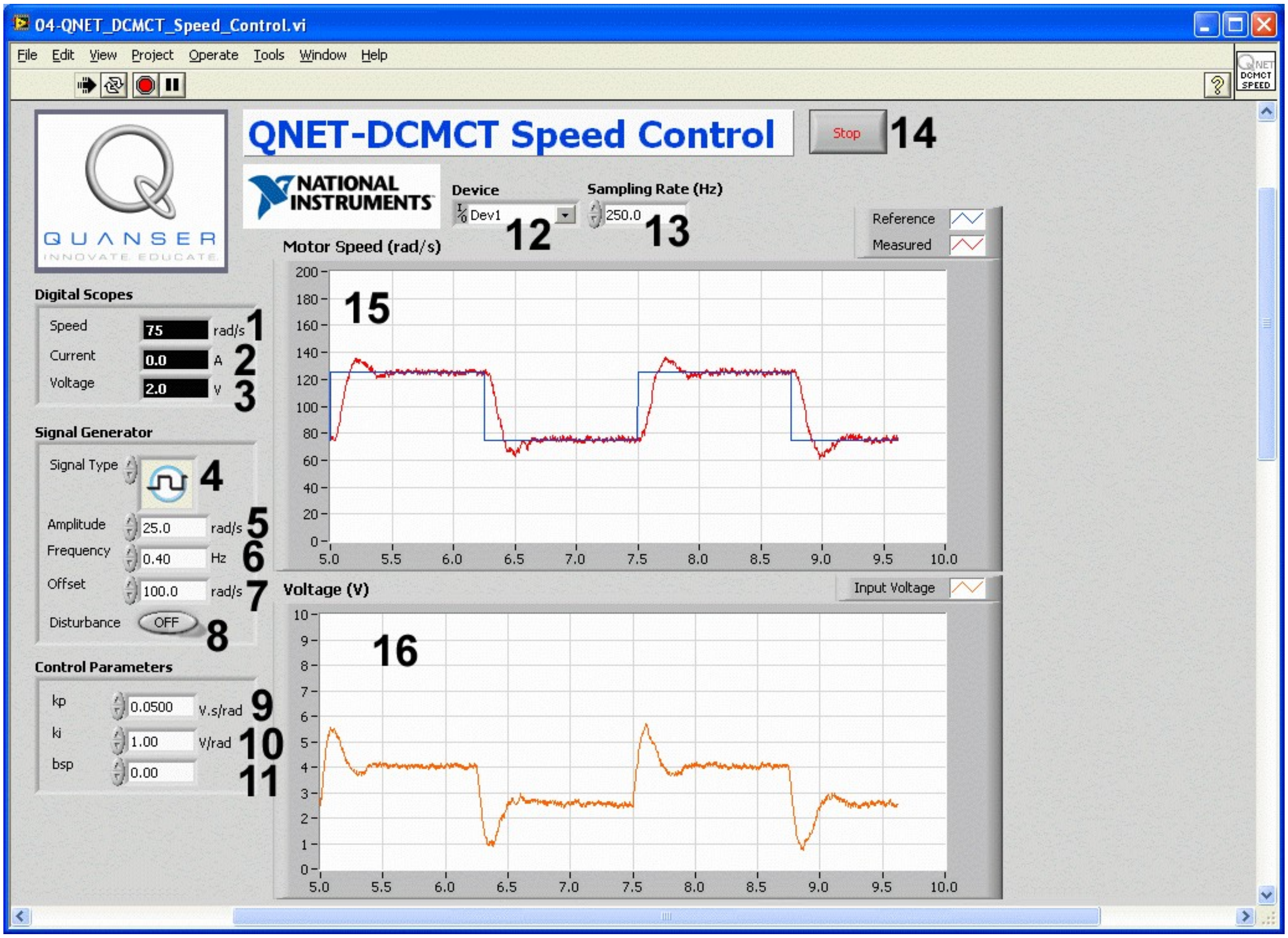

Figure 4: QNET DCMCT Speed Control VI.

ID #	***Label***	***Parameter***	***Description***	***Unit***
1	Speed	ω_m	Motor output speed numeric display.	rad/s
2	Current	I_m	Motor armature current numeric display.	A
3	Voltage	V_m	Motor input voltage numeric display.	V
4	Signal Type		Type of signal generated for the motor speed reference.	
5	Amplitude		Generated signal amplitude input box.	V
6	Frequency		Generated signal frequency input box.	Hz
7	Offset		Generated signal offset input box.	V
8	Disturbance	V_{sd}	Apply simulated disturbance voltage.	V
9	kp	k_p	Controller proportional gain input box.	V.s/rad

10	ki	k_i	Controller integral gain input box.	V/rad
11	bsp	b_{sp}	Controller set-point weight input box.	
12	Device		Selects the NI DAQ device.	
13	Sampling Rate		Sets the sampling rate of the VI.	Hz
14	Stop		Stops the LabVIEW VI from running.	
15	Speed	ω_m	Scope with reference (in blue) and measured (in red) motor speeds.	rad/s
16	Voltage	V_m	Scope with applied motor voltage (in red).	V

Table 3: Nomenclature of QNET-DCMCT Speed Control VI.

3.2.3. Position Control

The QNET DCMCT Position Control VI controls the position of the motor using a proportional-integral-derivative controller. The main elements of the VI front panel are summarized in Table 4 and identified in Figure 5 through the corresponding ID number.

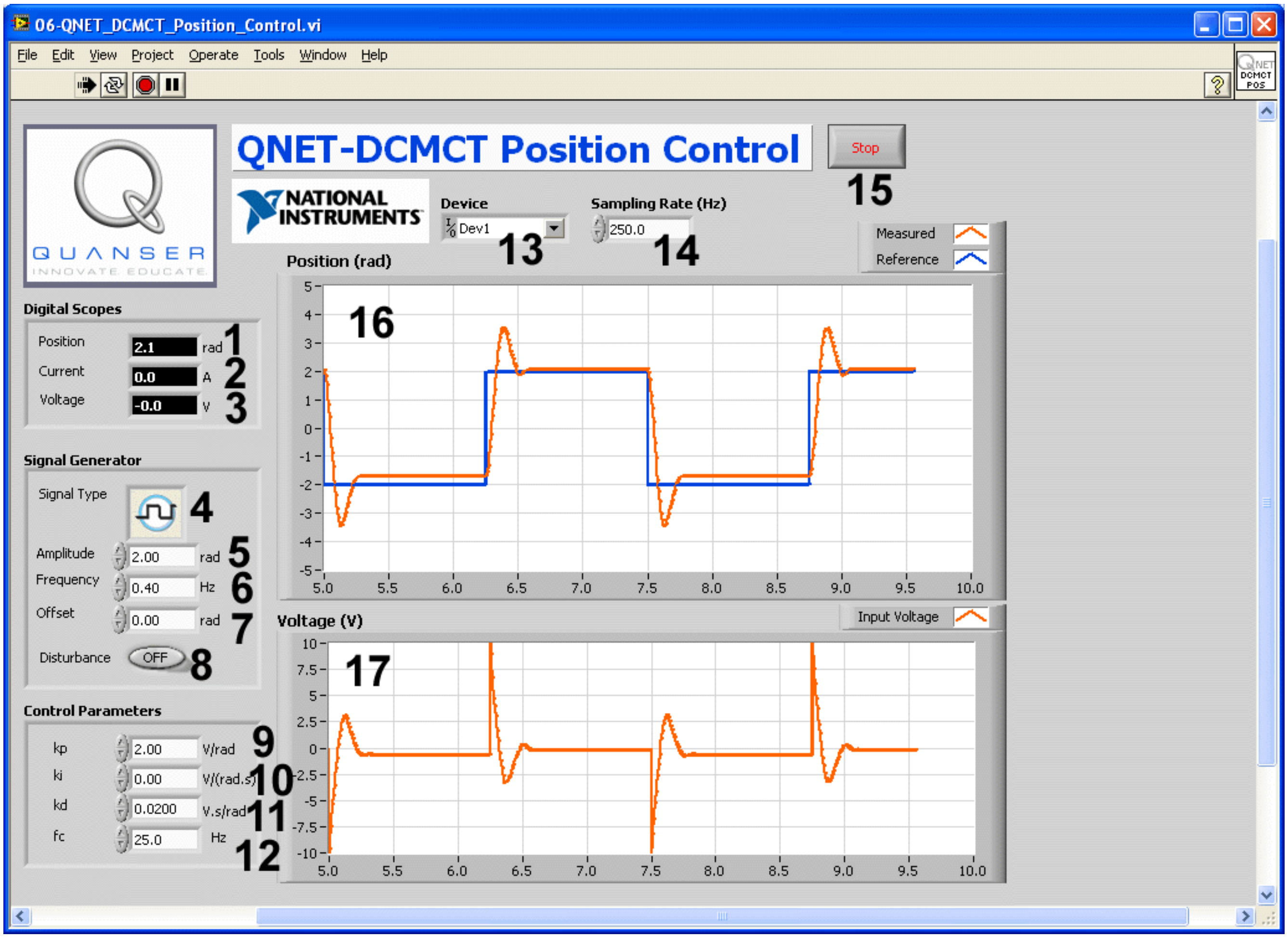

Figure 5: QNET DCMCT Position Control VI.

ID #	***Label***	***Parameter***	***Description***	***Unit***
1	Position	θ_m	Motor output speed numeric display.	rad/s
2	Current	I_m	Motor armature current numeric display.	A
3	Voltage	V_m	Motor input voltage numeric display.	V
4	Signal Type		Type of signal generated for the position reference.	
5	Amplitude		Generated signal amplitude input box.	V
6	Frequency		Generated signal frequency input box.	Hz
7	Offset		Generated signal offset input box.	V
8	Disturbance	V_{sd}	Apply simulated disturbance voltage.	V
9	kp	k_p	Controller proportional gain input box.	V.s/rad

10	ki	k_i	Controller integral gain input box.	V/rad
11	kd	k_d	Controller derivative gain input box.	V.s/rad
12	fc	f_c	Controller high-pass filter cutoff frequency.	Hz
13	Device		Selects the NI DAQ device.	
14	Sampling Rate		Sets the sampling rate of the VI.	Hz
15	Stop		Stops the LabVIEW VI from running.	
16	Position	θ_m	Scope with reference (in blue) and measured (in red) motor positions.	rad
17	Voltage	V_m	Scope with applied motor voltage (in red).	V

Table 4: Nomenclature of QNET-DCMCT Position Control VI.

4. In-Lab Experiments

4.1. Modeling

4.1.1. Bumptest

1. Open the QNET_DCMCT_Modeling.vi.
2. Ensure the correct *Device* is chosen, as shown in Figure 6

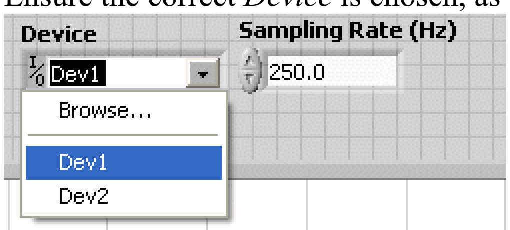

Figure 6: Selecting correct device.

3. Run the QNET_DCMCT_Modeling.vi. The DC motor should begin spinning and the scopes on the VI should appear similarity as shown in Figure 7.
4. In the *Signal Generator* section set:
   Amplitude = 2.0 V
   Frequency = 0.40 Hz
   Offset = 3.0 V
5. Once you have collected a step response, click on the *Stop* button to stop running the VI.
6. **Exercise 1**: Attach the responses in the *Speed (rad/s)* and *Voltage (V)* graphs. See the QNET User Manual for information on how to export a chart or graph to the clipboard.
7. Select the *Measurement Graphs* tab to view the measured response, similarly as depicted in Figure 8.

8. **Exercise 2**: Use the responses in the *Speed (rad/s)* and *Voltage (V)* graphs to compute the steady-state gain of the DC motor. Make sure you fill out Table 5. See the *Bumptest Method* section in the QNET Practical Control Guide for details on how to find the steady-state gain from a step response. Finally, you can use the *Graph Palette* for zooming functions and the *Cursor Palette* to measure data. See the LabVIEW help for more information on these tools.
9. **Exercise 3**: Based on the bumptest method, find the time constant. Make sure you complete Table 6 and see the *Bumptest Method* section in the QNET Practical Control Guide for information on how to find the time constant of the step response.
10. Enter the steady-state gain and time constant values found in this section in Table 7. These are called the *bumptest model parameters*.

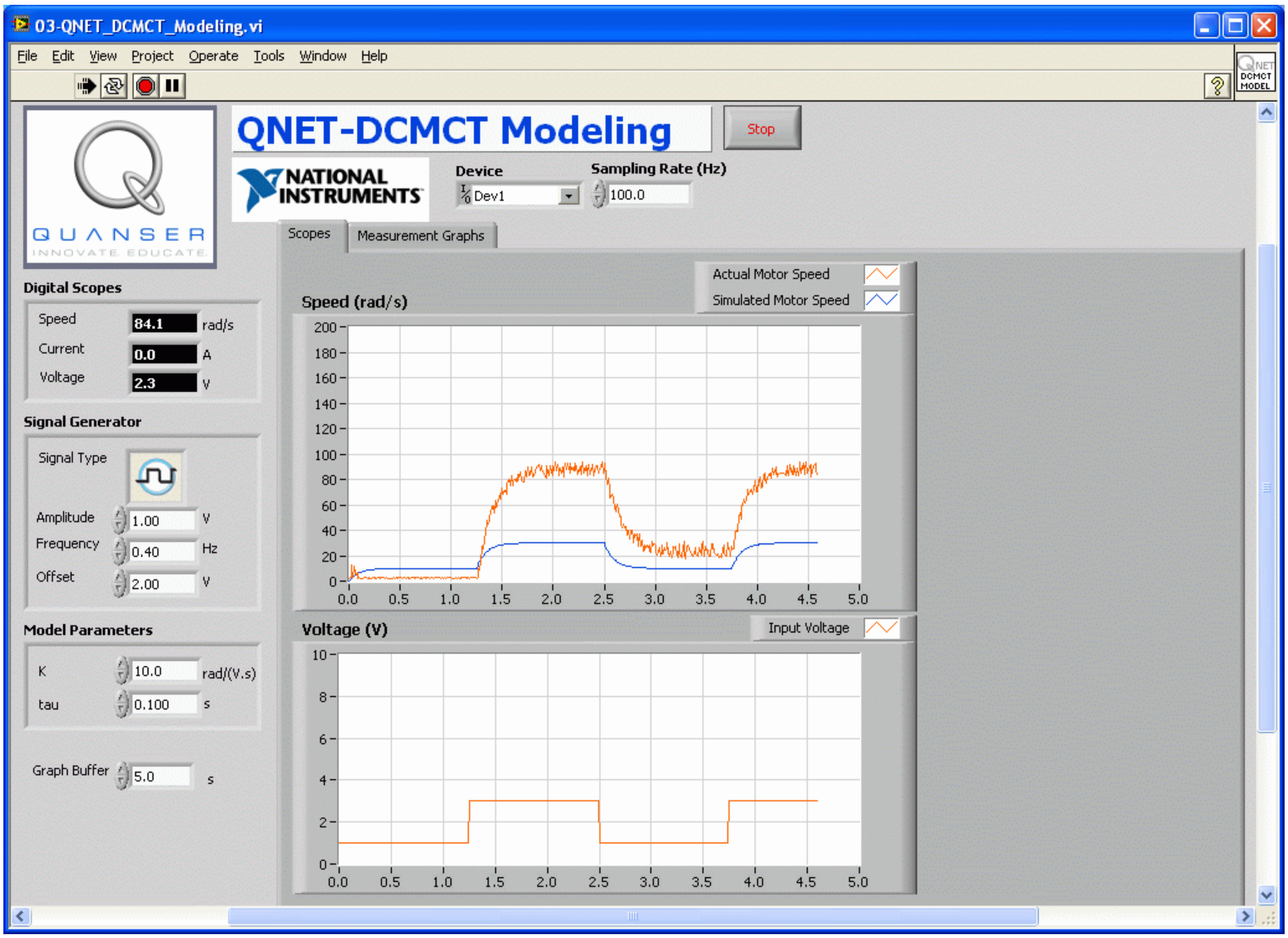

Figure 7: QNET DCMCT Modeling VI running.

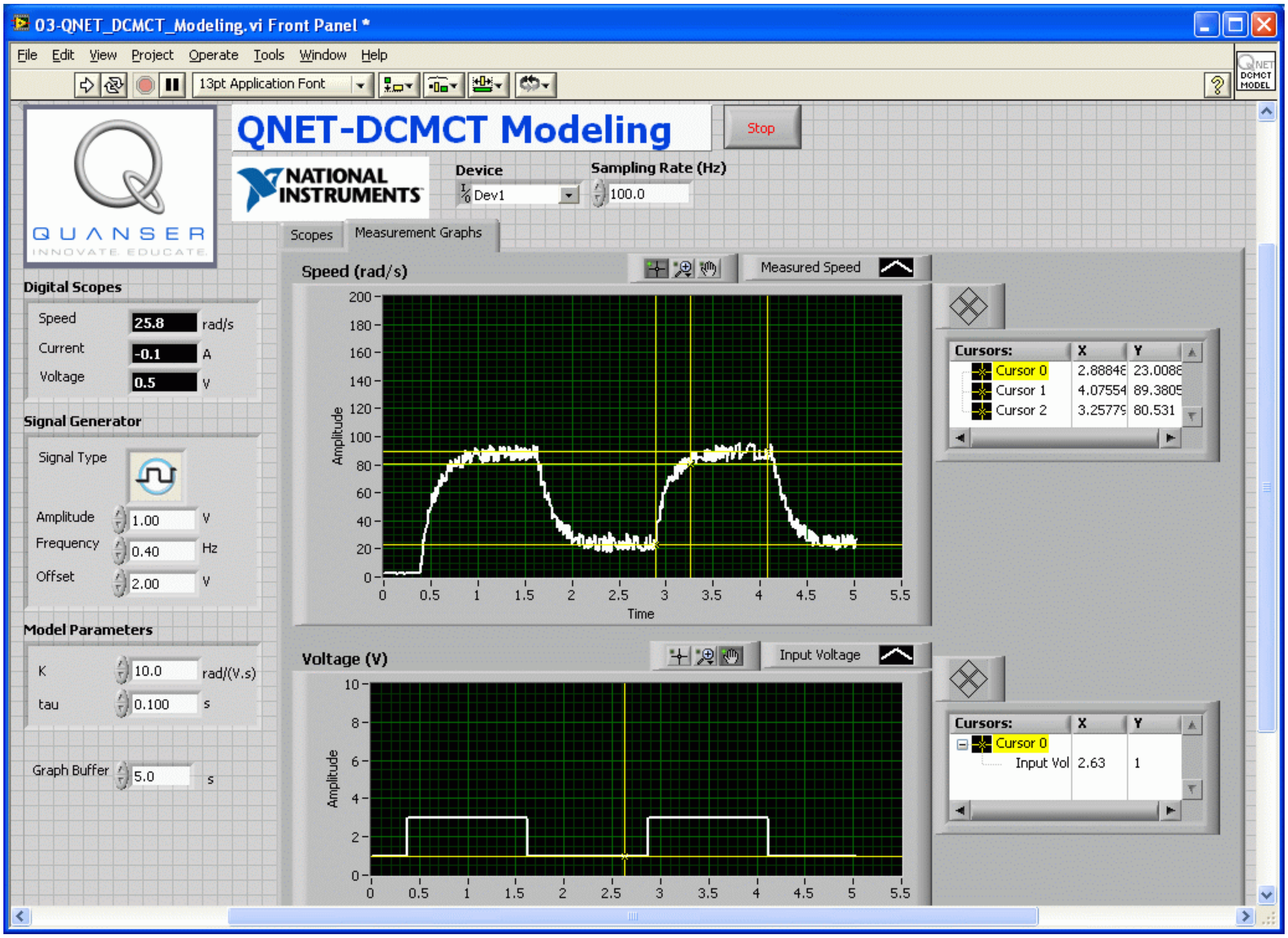

Figure 8: QNET DCMCT Modeling VI: sample response in Measurement Graphs.

4.1.2. Model Validation

1. Open the QNET_DCMCT_Modeling.vi.
2. Ensure the correct *Device* is chosen.
3. Run the QNET_DCMCT_Modeling.vi. You should hear the DC motor begin running and the scopes on the VI should appear similarity as shown in Figure 7.
4. In the *Signal Generator* section set:

   Amplitude = 2.0 V

   Frequency = 0.40 Hz

   Offset = 3.0 V
5. In the *Model Parameters* section of the VI, enter the bumptest model parameters, K and τ, that were found in Section 4.1.1. The blue simulation should match the red measured motor speed more closely.
6. **Exercise 4**: Attach the *Speed (rad/s)* and *Voltage (V)* chart responses from the *Scopes* tab. How well does your model represent the actual system? If they do not match, name one possible source for this discrepancy.

7. **Exercise 5**: Tune the steady-state gain, K, and time constant, *tau*, in the *Model Parameters* section so the simulation matches the actual system better. Enter both the *bumptest* and *tuned* model parameters in Table 7.

4.1.3. Exercises

Exercise 1: Bumptest Response

0	1	2

Exercise 2: Measure Steady-State Gain

Description	*Symbol*	*Value*	*Unit*
Steady-state motor speed	$\omega_{m,ss}$		rad/s
Initial step motor speed	ω_0		rad/s
Input step amplitude	A_v		V
Measured steady-state gain using bumptest	$K_{e,b}$		rad/(V.s)

0	1	2

Table 5: Finding steady-state gain using bumptest.

Exercise 3: Measure Time Constant

Description	*Symbol*	*Value*	*Unit*
Decay speed	$\omega_m(t_1)$		rad/s
Initial step time	t_0		s
Decay step time	t_1		s
Measured time constant using bumptest	$\tau_{e,b}$		s

0	1	2

Table 6: Finding time constant using bumptest.

Exercise 4: Bumptest Model Validation

0	1	2

Exercise 5: Tuned Model Parameters

0	1	2

Exercise 6: Results Summary

Description	Symbol	Value	Unit
In-Lab: Bumptest Modeling			
Open-Loop Steady-State Gain	$K_{e,b}$		rad/(V.s)
Open-Loop Time Constant	$\tau_{e,b}$		s
In-Lab: Model Validation			
Open-Loop Steady-State Gain	$K_{e,v}$		rad/(V.s)
Open-Loop Time Constant	$\tau_{e,v}$		s

0	1	2

Table 7: QNET DCMCT Modeling results summary

4.2. Speed Control

4.2.1. Qualitative PI Control

1. Open the QNET_DCMCT_Speed_Control.vi.
2. Ensure the correct *Device* is chosen.
3. Run the QNET_DCMCT_Speed_Control.vi. The motor should begin rotating and the scopes should look similar as shown in Figure 9.
4. In the *Signal Generator* section set:
   - *Signal Type* = 'square wave'
   - *Amplitude* = 25.0 rad/s
   - *Frequency* = 0.40 Hz
   - *Offset* =100.0 rad/s
5. In the *Control Parameters* section set:
   - *kp* = 0.0500 V.s/rad
   - *ki* = 1.00 V/rad
   - *bsp* = 0.00
6. **Exercise 1**: Examine the behaviour of the measured speed, shown in red, with respect to the reference speed, shown in blue, in the *Speed (rad/s)* scope. Explain what is happening.
7. Increment and decrement *kp* by steps of 0.005 V.s/rad.
8. **Exercise 2**: Look at the changes in the measured signal with respect to the reference signal. Explain the performance difference of changing *kp*.
9. Set *kp* to 0 V.s/rad and *ki* to 0 V/rad. The motor should stop spinning.
10. Increment the integral gain, *ki*, by steps of 0.05 V/rad. Vary the integral gain between 0.05 V/rad and 1.00 V/rad.
11. **Exercise 3**: Examine the response of the measured speed in the *Speed (rad/s)* scope and compare the result when *ki* is set low to when it is set high.
12. Stop the VI by clicking on the *Stop* button

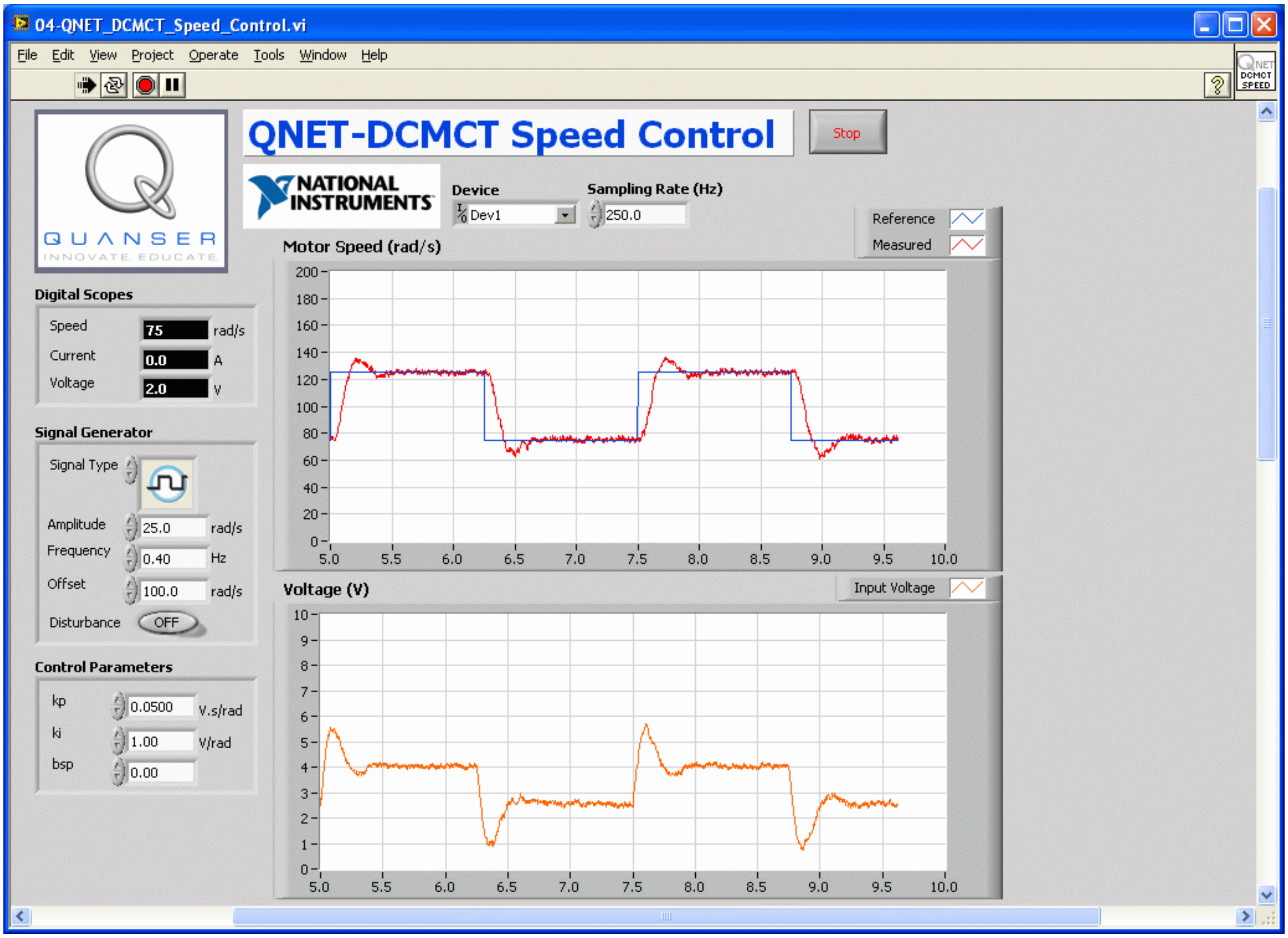

Figure 9: Running the QNET Speed Control VI.

4.2.2. PI Control according to Specifications

1. **Exercise 4**: Using the equations outlined in the *Peak Time and Overshoot* section of the QNET Practical Control Guide, calculate the expected peak time, t_p, and percentage overshoot, *PO*, given the following *Speed Lab Design (SLD)* specifications:
   - *zeta* = 0.75
   - *w0*= 16.0 rad/s

   Optional: You can also design a VI that simulates the DC motor first-order model with a PI control and have it calculate the peak time and overshoot.
2. **Exercise 5**: Calculate the proportional, k_p, and integral, k_i, control gains according to the model parameters found in Section 4.1.2 and the SLD specifications.
3. Run the QNET_DCMCT_Speed_Control.vi. The motor should begin spinning and the scopes plotting traces similarly as illustrated in Figure 9, above.
4. In *Signal Generator* set:
   - *Signal Type* ='square wave'
   - *Amplitude* = 25.0 rad/s

- *Frequency* = 0.40 Hz
- *Offset* = 100.0 rad/s

5. In the *Control Parameters* section, enter the SLD PI control gains found in Exercise 5 and make sure *bsp* = 0.00.
6. Stop the VI when you collected two sample cycles by clicking on the *Stop* button.
7. **Exercise 6**: Capture the measured SLD speed response. Make sure you include both the *Speed (rad/s)* and the control signal *Voltage (V)* scopes.
8. **Exercise 7**: Measure the peak time and percentage overshoot of the measured SLD response. Are the specifications satisfied?
9. **Exercise 8**: What effect does increasing the specification *zeta* have on the measured speed response? How about on the control gains? Use the damping ratio equation given in the *Peak Time and Overshoot* section of the QNET Practical Control Guide for more help if needed.
10. **Exercise 9**: What effect does increasing the specification *w0* have on the measured speed response and the generated control gains? Use the natural frequency equation found in the *Peak Time and Overshoot* section of the QNET Practical Control Guide for more help if needed.

4.2.3. Effect of Set-Point Weight

1. Run the QNET_DCMCT_Speed_Control.vi. The motor should begin rotating.
2. In the *Signal Generator* section set:
    - *Signal Type* = 'square wave'
    - *Amplitude* = 25.0 rad/s
    - *Frequency* = 0.40 Hz
    - *Offset* = 100.0 rad/s
3. In the *Control Parameters* section set:
    - *kp* = 0.050 V.s/rad
    - *ki* = 1.50 V/rad
    - *bsp*= 0.00
4. Increment the set-point weight parameter *bsp* in steps of 0.05. Vary the parameter between 0 and 1.
5. **Exercise 10**: Examine the effect that raising *bsp* has on the shape of the measured speed signal in the *Speed (rad/s)* scope. Explain what the set-point weight parameter is doing.
6. Stop the VI by clicking on the *Stop* button.

4.2.4. Tracking Triangular Signals

1. Run the QNET_DCMCT_Speed_Control.vi. The motor should begin rotating.
2. In Signal Generator set:
    - *Signal Type* = 'triangular wave'
    - *Amplitude* = 50.0 rad/s
    - *Frequency* = 0.40 Hz
    - *Offset* = 100.0 rad/s
3. In the *Control Parameters* section set:
    - *kp* = 0.20 V.s/rad
    - *ki* = 0.00 V/rad

- *bsp* = 1.00

4. **Exercise 11**: Compare the measured speed and the reference speed. Explain why there is a tracking error.
5. Increase *ki* to 0.1 V/rad and examine the response. Vary *ki* between 0.1 V/rad and 1.0 V/rad.
6. **Exercise 12**: What effect does increasing *ki* have on the tracking ability of the measured signal? Explain using the observed behaviour in the scope.
7. Stop the VI by clicking on the *Stop* button

4.2.5. Exercises

Exercise 1: Describe the Speed Response

0	**1**	**2**

Exercise 2: Effect of Proportional Gain on Speed Control

0	**1**	**2**

Exercise 3: Pure Integral Control Response

0	1	2

Exercise 4: Peak Time and Overshoot

Description	*Symbol*	*Value*	*Unit*
Natural frequency specification	ω_0	16.0	rad/s
Damping ratio specification	ζ	0.75	
Peak time	t_p		s
Percentage overshoot	PO		%

Table 8: Expected peak time and overshoot.

0	1	2

Exercise 5: Design PI Gains to Specifications

Description	***Symbol***	***Value***	***Unit***
Natural frequency specification	ω_0	16.0	rad/s
Damping ratio specification	ζ	0.75	
Steady-state model gain	K		rad/(V.s)
Model time constant	τ		s
Proportional gain	k_p		V.s/rad
Integral gain	k_i		V/rad

Table 9: PI speed control design.

0	**1**	**2**

Exercise 6: Designed Speed Control Response

0	1	2

Exercise 7: Peak Time and Overshoot of Response

0	1	2

Exercise 8: Effect of Increasing Damping Ratio

Description	*Symbol*	*Behaviour*	*Unit*
Peak time	t_p		s
Percentage overshoot	PO		%
Proportional gain	k_p		V.s/rad
Integral gain	k_i		V/rad

0	1	2

Exercise 9: Effect of Increasing Natural Frequency

Description	*Symbol*	*Behaviour*	*Unit*
Peak time	t_p		s
Percentage overshoot	PO		%
Proportional gain	k_p		V.s/rad
Integral gain	k_i		V/rad

0	1	2

Exercise 10: Set-Point Weight

0	1	2

Exercise 11: Tracking Error

0	1	2

Exercise 12: Effect of Integral Gain on Tracking Error

0	1	2

4.3. Position Control

4.3.1. Qualitative PD Control

1. Open the QNET_DCMCT_Position_Control.vi.
2. Ensure the correct *Device* is chosen.
3. Run the QNET_DCMCT_Position_Control.vi. The DC motor should be rotating back and forth and the scopes on the VI should appear similarity as shown in Figure 10.
4. In the *Signal Generator* section set:
   - *Amplitude* = 2.00 rad
   - *Frequency* = 0.40 Hz
   - *Offset* = 0.00 rad
5. In the *Control Parameters* section set:
   - *kp* = 2.00 V/rad
   - *ki* = 0.00 V/rad
   - *kd* = 0.00 V.s/rad
6. Change the proportional gain, *kp*, by steps of 0.25 V/rad. Try the following gains: kp = 0.5, 1, 2, and 4 V/rad.
7. **Exercise 1**: Examine the behaviour of the measured position (red line) with respect to the reference position (blue line) in the *Position (rad)* scope. Explain what is happening.
8. **Exercise 2**: Describe the steady-state error to a step input.
9. Increment the derivative gain, *kd*, by steps of 0.01 V.s/rad.
10. **Exercise 3**: Looks at the changes in the measured position with respect to the desired position. Explain what is happening.
11. Using the equations outlined in the *Peak Time and Overshoot* section of the QNET Practical Control Guide, calculate the expected peak time, t_p, and percentage overshoot, *PO*, given the following specifications:
    - *zeta* = 0.75

- *w0*= 16.0 rad/s

Optional: You can also design a VI that simulates the DC motor first-order model with a PI control and have it calculate the peak time and overshoot.

12. **Exercise 5**: Calculate the proportional, k_p, and integral, k_i, control gains according to the model parameters found in Section 4.1.2 and the SLD specifications.
13. Stop the VI by clicking on the *Stop* button.

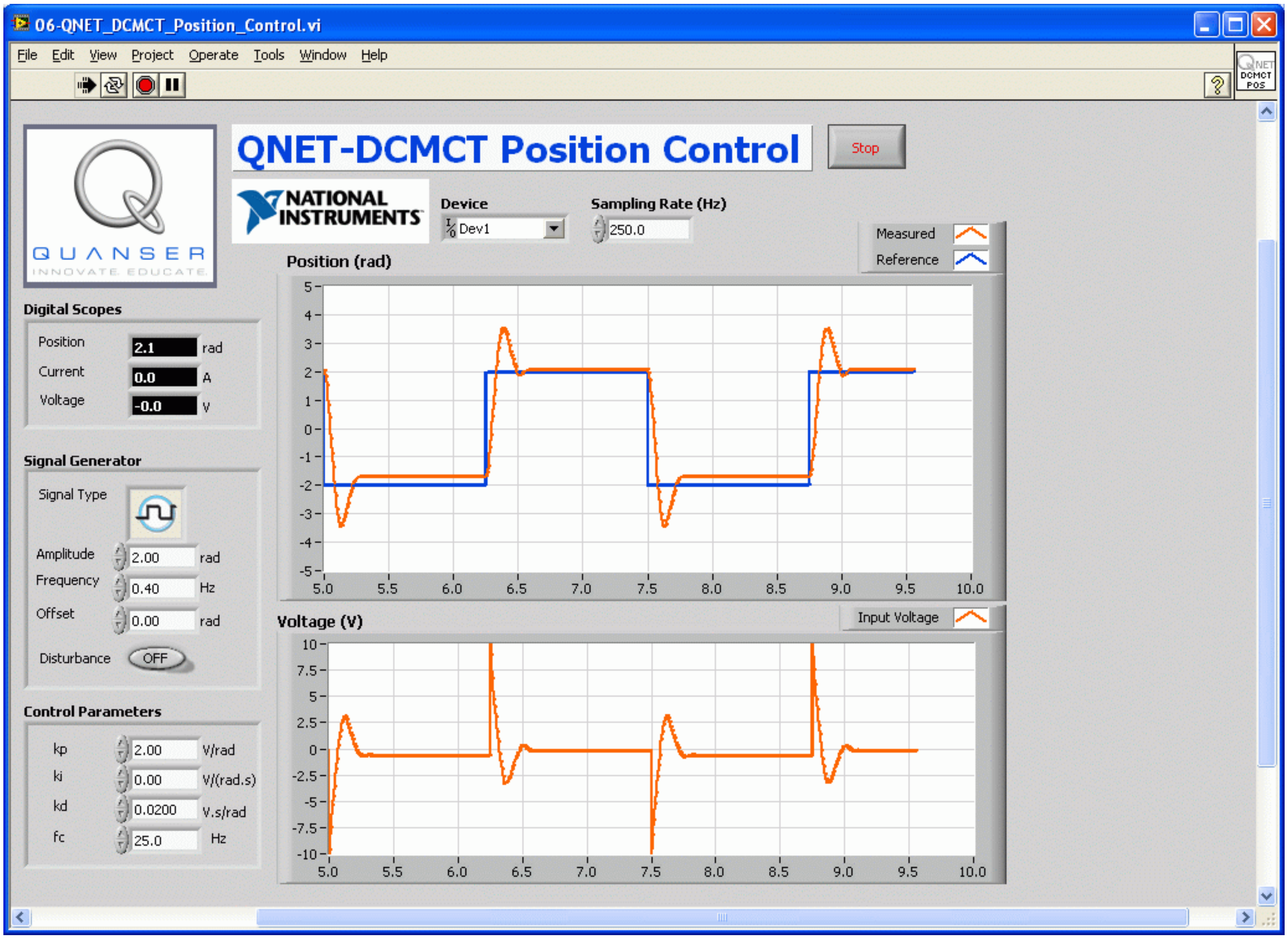

Figure 10: Running the QNET Position Control VI.

4.3.2. PD Control according to Specifications

1. **Exercise 4**: Using the equations in the *Peak Time and Overshoot* section of the QNET Practical Control Guide, calculate the expected peak time, t_p, and percentage overshoot, *PO*, given
   - *zeta* = 0.60
   - *w0* = 25.0 rad/s
   - *p0* = 0.0

Optional: You can also design a VI that simulates the DC motor first-order model with a PD control and have it calculate the peak time and overshoot.

2. **Exercise 5**: Calculate the proportional, k_p, and derivative, k_d, control gains according to the model parameters found in Section 4.1.2 and the specifications above.
3. Run the QNET_DCMCT_Position_Control.vi. You should see the DC motor rotating back and forth.
4. In the *Signal Generator* section set:
   - *Amplitude* = 2.00 rad
   - *Frequency* = 0.40 Hz
   - *Offset* = 0.00 rad
5. In the *Control Parameters* section, set the PD gains found in Exercise 5.
6. **Exercise 6**: Capture the position response found in the *Position (rad)* scope and and control signal used in the *Voltage (V)* scope.
7. **Exercise 7**: Measure the peak time and percentage overshoot of the measured position response. Are the specifications satisfied? If they are not, then give one possible reason why there would be discrepancy.
8. **Exercise 8**: What effect does changing the specification *zeta* have on the measured position response and the generated control gains? See the *Peak Time and Overshoot* section of the QNET Practical Control Guide for more help.
9. **Exercise 9**: What effect does changing the specification *w0* have on the measured position response and the generated control gains? See the *Peak Time and Overshoot* section of the QNET Practical Control Guide for more help.
10. Stop the VI by clicking on the *Stop* button.

4.3.3. Response to Load Disturbance

1. **Exercise 10**: In the *Response to Load Disturbance* section of the QNET Practical Control Guide, the load disturbance to motor position closed-loop PID block diagram is found. Consider the same regulation system, $r = 0$, when $b_{sp}=1$ and $b_{sd}=1$ and show the block diagram representing the simulated disturbance to motor position closed-loop interaction (in this case $T_d = 0$).
2. **Exercise 11**: Find the closed-loop PID transfer function describing the position of the motor with respect to the simulated disturbance voltage: $G_{\theta,Vsd}(s) = \theta(s)/V_{sd}(s)$.
3. **Exercise 12**: Find the steady-state motor angle due to a simulated disturbance step of $V_{sd} = V_{sd0} / s$.
4. **Exercise 13**: A step of $V_{sd} = V_{sd0} / s$ with $V_{sd0} = 3$ V is added to the motor voltage to simulate a disturbance torque. Evaluate the steady-state angle of the motor when a PD controller is used with the gains $k_p = 2$ V/rad and $k_d = 0.02$ V.s/rad. Then, calculate the steady-state angle when using a PID controller with the gains $k_p = 2$ V/rad, $k_d = 0.02$ V.s/rad, and $k_i = 1$ V/rad/s. Enter your numeric answers in Table 14.
   Optional: You can also design a VI that simulates the DC motor first-order model with a PID control and a step disturbance and examine the steady-state angle obtained from the response.
5. Run the QNET_DCMCT_Position_Control.vi. The DC motor should be rotating back and forth.
6. In the *Signal Generator* section set:

- *Amplitude* = 0 rad
- *Frequency* = 0.40 Hz
- *Offset* = 0 rad

7. In the *Control Parameters* section set:
   - *kp* = 2.0 V/rad
   - *ki* = 0.0 V/(rad.s)
   - *kd* = 0.02 V.s/rad
8. Apply the disturbance by clicking on the *Disturbance* toggle switch situated below the *Signal Generator*.
9. **Exercise 14**: Examine the effect of the disturbance on the measured position. Attach a response of the motor position when the disturbance is applied, record the obtained steady-state angle, and compare it to the value estimated in Exercise 13.
10. Turn OFF the *Disturbance* switch
11. In the *Control Parameters* section set:
    - *kp* = 2.0 V/rad
    - *ki* = 2.0 V/(rad.s)
    - *ki* = 0.02 V.s/rad
12. Apply the disturbance by clicking on the *Disturbance* toggle switch.
13. **Exercise 15**: Examine the effect of the disturbance on the measured position. Explain the difference of the disturbance response with the integral action added and compare to the result you obtained in Exercise 13.
14. Stop the VI by clicking on the *Stop* button.

4.3.4. Exercises

Exercise 1: Pure Proportional Control

0	1	2

Exercise 2: PD Steady-State Error

0	1	2

Exercise 3: Adding Derivative Control

0	1	2

Exercise 4: Peak Time and Overshoot

Description	***Symbol***	***Value***	***Unit***
Natural frequency specification	ω_0	25.0	rad/s
Damping ratio specification	ζ	0.6	
Peak time	t_p		s
Percentage overshoot	PO		%

0	1	2

Table 10: Expected peak time and overshoot.

Exercise 5: Design PD Gains to Specifications

Description	*Symbol*	*Value*	*Unit*
Natural frequency specification	ω_0	25.0	rad/s
Damping ratio specification	ζ	0.6	
Steady-state model gain	K		rad/(V.s)
Model time constant	τ		s
Proportional gain	k_p		V.s/rad
Integral gain	k_i		V/rad

0	1	2

Table 11: PD speed control design.

Exercise 6: Designed PD Position Control Response

0	1	2

Exercise 7: Peak Time and Overshoot of PD Response

0	1	2

Exercise 8: Effect of Increasing Damping Ratio

Description	*Symbol*	*Behaviour*	*Unit*
Peak time	t_p		s
Percentage overshoot	PO		%
Proportional gain	k_p		V.s/rad
Derivative gain	k_d		V/rad

0	1	2

Table 12: Effect of increasing damping ratio specification in position control.

Exercise 9: Effect of Increasing Natural Frequency

Description	*Symbol*	*Behaviour*	*Unit*
Peak time	t_p		s
Percentage overshoot	PO		%
Proportional gain	k_p		V.s/rad
Derivative gain	k_d		V/rad

0	1	2

Table 13: Effect of increasing natural frequency specification in position control.

Exercise 10: Block Diagram of PID Simulated Disturbance

0	1	2

Exercise 11: PID Simulated Disturbance Transfer Function

0	1	2

Exercise 12: PD Steady-State Angle

0	1	2

Exercise 13: Evaluate PD and PID Steady-State Angles

0	1	2

Description	***Symbol***	***Value***	***Unit***
Proportional gain	k_p	2.0	V/rad
Integral gain	k_i	1.0	V/(rad.s)
Derivative gain	k_d	0.02	V.s/rad
Simulated disturbance	V_{sd}	3.0	V
PD steady-state angle	$\theta_{ss,d}$		rad
PID steady-state angle	$\theta_{ss,pid}$		rad

Table 14: Motor position steady-state angle due to simulated disturbance.

Exercise 14: Measured PD Disturbance

0	1	2

Exercise 15: Measured PID Disturbance

0	1	2

5. References

[1] QNET User Manual
[2] QNET Practical Control Guide