1 INTRODUCTION

The Quanser HIL Driving Simulator (QDS) is a modular and expandable LabVIEW model of a car driving on a closed track. The model is intended as a platform for the development, implementation and evaluation of a variety of control systems. The QDS consists of a variety of components that are integrated together to create a representation of a vehicle being driven on a track. One possible configuration is shown in Figure 1.1. The model utilizes the Quanser Rapid Control Prototyping Toolkit (RCP) to facilitate hardware-in-the-loop interfacing (HIL). The Quanser 3D Viewer is also used to create an immersive visual environment for testing and evaluating controllers. Students are expected to observe and think critically about the effects of system parameters on not just the discrete plant, but the overall system.

Some examples of the real-world control problems that can be addressed using the QDS include parking assist systems, radar guided cruise control, active suspension, traction control and autonomous navigation.

Figure 1.1: Quanser Driving Simulator

Topics Covered

- Basic data gathering using LabVIEW™
- PD speed control of a DC motor
- PI position control of a DC motor
- Autonomous vehicle navigation

Note: This workbook contains a single independent laboratory experiment as an introduction to the QDS platform. If you are interested in the complete QDS platform, please contact info@quanser.com

2 VEHICLE SPEED CONTROL

2.1 Introduction

2.1.1 Electronic Throttle Control

Electronic throttle control (ETC), traction control and cruise control have become standard features on modern cars. More recently, with the advent of radar-guided cruise control, and pre-crash systems electronic vehicle control units have begun to play an increasingly critical role in the real-time control of vehicle speed. Though the implementation of these systems can be complex, the essential system can be viewed as a closed-loop speed controller that sends throttle commands to the engine, shown in Figure 2.2.

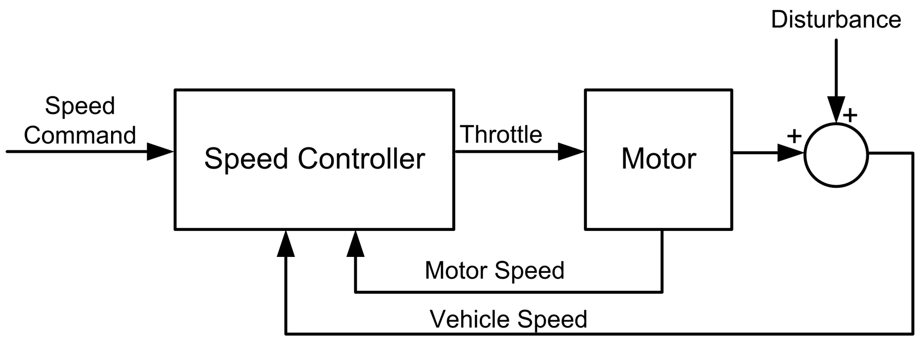

Figure 2.2: ETC controller

2.1.2 DC Motor Speed Control

For this laboratory, you will use the QDS in conjunction with a Qube Rotary Servo to develop a Proportional-Integral (PI) speed controller to regulate the speed of the simulated vehicle. This exercise is analogous to developing an ETC for an electric vehicle, shown in Figure 2.3. In this case, the disturbance to the speed of the vehicle is the slope of the road, which is converted into a voltage and added to the control command. The controller that you will design takes the actual speed of the Qube motor as the derivative of the encoder measurements, and the commanded speed from the internal driver controller in the QDS. The PI controller then outputs the motor voltage, V_m, which is added to the slope of the road and sent to the Qube. The slope of the road is provided by the sensors inside the QDS.

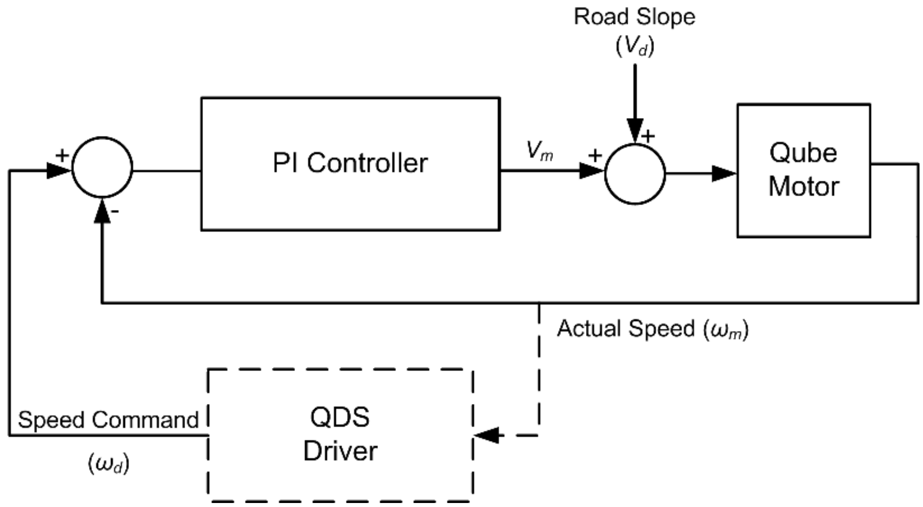

Figure 2.3: QDS speed controller

2.2 Background

2.2.1 PI Control

The PI controller is one of the most common control algorithms. It combines the error reduction of proportional control, with the offset elimination of integral control. The proportional term tracks the instantaneous error, while the integral term controls the offset by tracking the total error over time. The longer there is error in the system, the more the integral term tries to compensate. Though the lack of a derivative term can cause overshoot and a longer settling time, for systems with simple dynamics the algorithm can provide near-optimal performance with no steady-state error.

In the time-domain, the linear behavior of a PI controller can be described by:

$$u(t) = k_p(r(t) - y(t)) + k_i \int_0^t (r(t) - y(t))\, dt \tag{2.1}$$

where u(t) is the control signal, r(t) is the reference, and y(t) is the measured process output.

Question 2.1
From Equation 2.1, determine the PI controller transfer function in the Laplace domain.

Answer 2.1
If you let $e(t) = r(t) - y(t)$ then the controller transfer function can be stated as:

$$\frac{U(s)}{E(s)} = C(s) = k_p + \frac{k_i}{s} = \frac{K_c(s+z)}{s} \tag{Ans.2.1}$$

where K_c is the controller gain and z is a zero.

□□□

Question 2.2
What is the steady-state error of the closed-loop system.

Answer 2.2
The use of an integral term will eliminate any steady-state error.

□ □ □

2.3 Speed Controller Implementation

2.3.1 QDS Speed Control Implementation

Experiment Setup

The Quanser Driving Simulator - Speed VI shown in Figure 2.4 is used to implement the closed-loop PI speed controller on the QDS platform. The goal of this section is to observe the response of the actual system in the context of the driving simulator, and observe the effects of the PI controller on the performance of the vehicle.

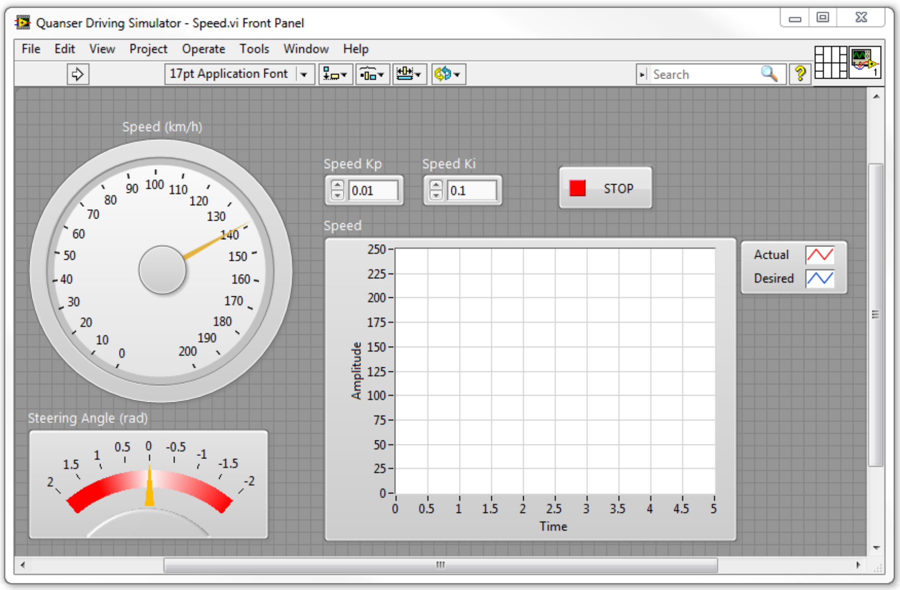

Figure 2.4: Quanser Driving Simulator with HIL Speed Control

Note: If you are using the QUBE-Servo for NI myRIO, please begin by opening the *Cruise Control - myRIO* LabVIEW project in the myRIO folder. The control subsystems are in the *QUBE-Servo Speed Control - myRIO* VI deployed to the NI myRIO, and the simulation parameters in the *Cruise Control* VI.

Question 2.3
If the car has 18 inch rims and 1 inch tires, derive the value of *Car to Wheel* in the Speed Control subsystem block diagram. This gain converts the car speed in m/s to wheel rotational speed in rad/s.

Answer 2.3
The total diameter is 20", which in meters is 0.508 m. The conversion is therefore

$$1\, m/s = 2\pi \frac{1}{0.508}\, rot/s = \left(\frac{2}{0.508}\right) rad/s \qquad \text{(Ans.2.2)}$$

and the value of the gain block is therefore 3.937

□ □ □

1. Enter a k_p gain of 0.3 into the *Kp* gain control, and a value of 0 into the *Ki* gain control.
2. Run the VI.

3. Observe the performance of the car as it makes a lap of the track.

4. Enter the k_p and k_i control gains used in the QUBE-Servo Speed Control Lab into the gain controls.

5. Run the VI.

Question 2.4
What do you observe about the performance of the vehicle in both cases?

Answer 2.4
Without integral gain the vehicle visibly slows when climbing hills, and is unable to compensate for steady-state errors because there is no disturbance rejection.

Question 2.5
Is the controller able to track the desired speed effectively?

Answer 2.5
The controller should be able to track the desired speed quite well, as shown in Figure 2.5.

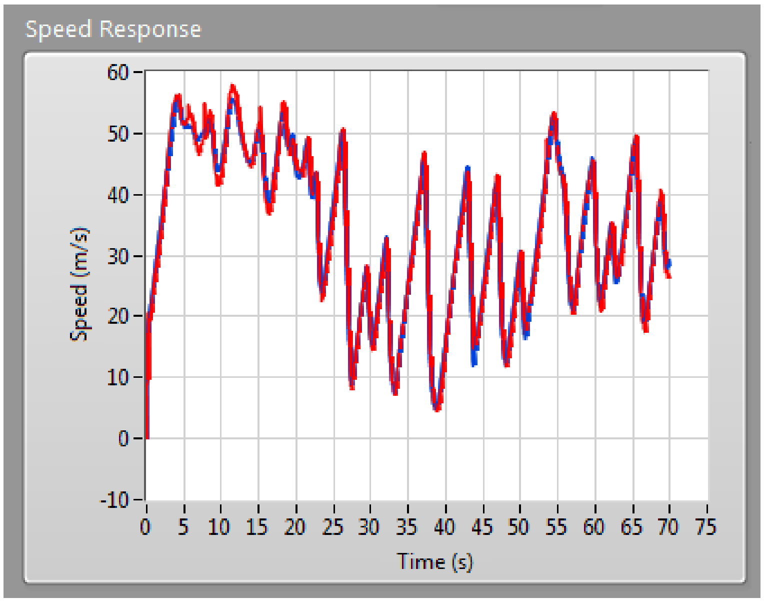

Figure 2.5: QDS speed control response

6. On the block diagram, change the magnitude of the *Slope Gain* constant to 0.

Question 2.6
How well is the controller able to track the speed command if the disturbance is eliminated?

Answer 2.6
The controller should be able to track the desired speed very well compared to the response in Figure 2.5, especially near the beginning of the track where the controller may have struggled with the significant elevation changes. A sample response is shown in Figure 2.6.

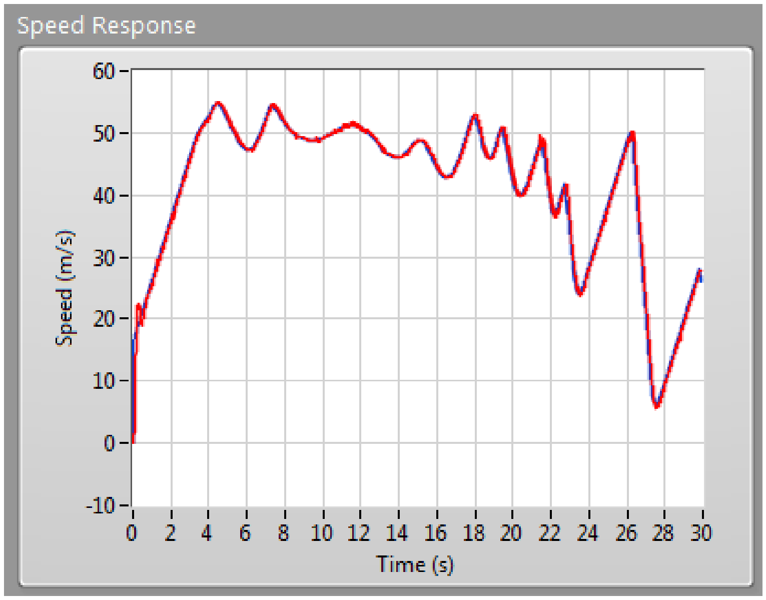

Figure 2.6: QDS speed response without disturbance

□□□

7. On the block diagram, replace the output of the Autodriver subsystem with a constant of 50 m/s.
8. Reset the *Slope Gain* block back to the original gain of 10. Rebuild and re-run the simulation.

Question 2.7
How well is the controller able to compensate for the slope disturbance? How does this compare to the setpoint tracking performance?

Answer 2.7
The controller is not able to compensate for the slope disturbance as well as it matches changes in the desired speed. This is because the control gains were tuned based on the response specifications, not the slope disturbance. A sample response is shown in Figure 2.7.

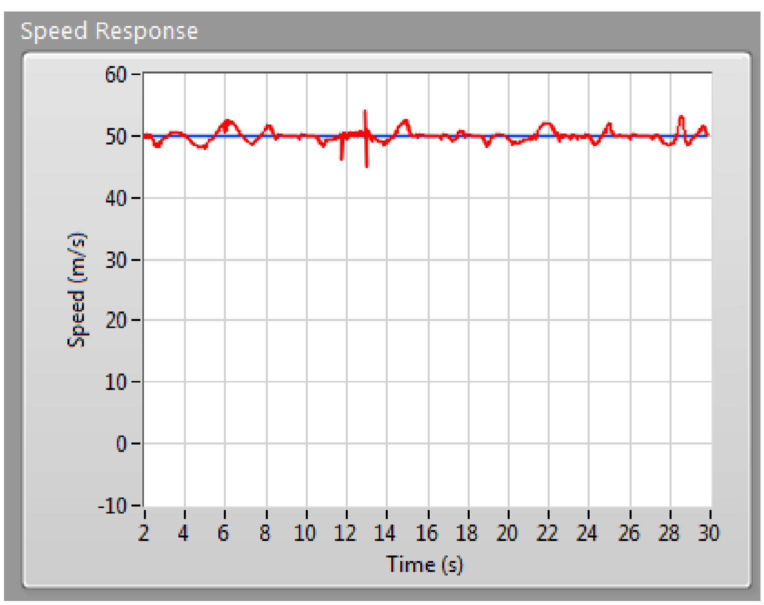

Figure 2.7: QDS disturbance response

□□□

9. If necessary, re-tune the controller gains to achieve the desired performance.

Question 2.8
Record the final control gains and response plots.

© 2014 Quanser Inc., All rights reserved.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. Quanser Inc. grants the following rights: a) The right to reproduce the work, to incorporate the work into one or more collections, and to reproduce the work as incorporated in the collections, b) to create and reproduce adaptations provided reasonable steps are taken to clearly identify the changes that were made to the original work, c) to distribute and publically perform the work including as incorporated in collections, and d) to distribute and publicly perform adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. These rights are granted subject to and limited by the following restrictions: a) You may not exercise any of the rights granted to You in above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation, and b) You must keep intact all copyright notices for the Work and provide the name Quanser Inc. for attribution. These restrictions may not be waved without express prior written permission of Quanser Inc.