Final Report 2000-05

Research

Differential GPS Based Control of a Heavy Vehicle

Minnesota Guidestar

Technical Report Documentation Page

1. Report No.	2.	3. Recipient's Accession No.
MN/RIC-2000-05		

4. Title and Subtitle	5. Report Date
Differential GPS Based Control of a Heavy Vehicle	January 1999
	6.

7. Author(s)

Lee Alexander Max Donath

8. Performing Organization Report No.

9. Performing Organization Name and Address

University of Minnesota
Department of Mechanical Engineering
111 Church Street SE
Minneapolis, MN 55455

10. Project/Task/Work Unit No.

11. Contract (C) or Grant (G) No.

(C) 74708
(G) Work Order #59

12. Sponsoring Organization Name and Address

Minnesota Department of Transportation
395 John Ireland Boulevard Mail Stop 330
St. Paul, Minnesota 55155

13. Type of Report and Period Covered

Final Report 12/97 – 1/00

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract (200 word limit)

This report describes development of technologies that can assist a driver with safe lanekeeping in the event that the driver is incapacitated due to fatigue. We use a Differential GPS system to sense the position and velocity of the vehicle since it appears that this method offers adequate precision with a low enough infrastructure cost to make the system practical in most rural settings. We have concentrated on heavy vehicles (a class 8 truck tractor) partly because of the more favorable economics associated with an installation of this type of system on a commercial vehicle, and partly because of the commercial driver's higher exposure to conditions that engender drowsy driving. Two potential applications of the steering, throttle and brake controllers developed in the course of this research are described. The first is a virtual rumble strip that vibrates the wheel whenever the vehicle drifts out of its lane. The second is a system that senses the erratic steering that presages loss of consciousness and then takes control of the vehicle, pulling it over to a safe stop.

17. Document Analysis/Descriptors

Lateral control
Longitudinal control
GPS
Driver fatigue

18. Availability Statement

No restrictions. Document available from:
National Technical Information Services,
Springfield, Virginia 22161

19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	74	

Differential GPS Based Control of Heavy Vehicles

Final Report

Prepared by

Lee Alexander, Max Donath

Department of Mechanical Engineering
University of Minnesota
111 Church Street SE
Minneapolis, MN 55455

January 1999

Published by

Minnesota Department of Transportation
Office of Research Administration
First Floor
395 John Ireland Boulevard, Mail Stop 330
St Paul, Minnesota 55455

Acknowledgments

The authors would like to express appreciation to the Minnesota Department of Transportation, the Center for Transportation Studies, the ITS Institute for their financial and staff support, and the Office of Minnesota Road Research for the use of the Mn/ROAD test facility for the duration of the research described herein. Jack Herndon and the rest of the staff at Mn/ROAD were especially cooperative.

We would also like to acknowledge the role of the Navistar International Transportation Co. who made the International Navistor 9400 rig available to us for our research.

We would like to thank the members of the Intelligent Vehicles Lab and the Robotics Lab at the U of M for their enthusiasm and encouragement.

Last but not least, a heartfelt thanks to the 400hp Cat 3406 diesel engine that so patiently idled away the hours it took to debug the control programs developed during the course of this research.

EXECUTIVE SUMMARY

Approximately one fourth of the fatalities on rural roads are caused by some form of loss of control in which the vehicle leaves the lane. Most such incidents are due to driver fatigue, drowsiness and inattention. Our long term goal is to be able to sense when loss of driver control is imminent and then have the control system actively intervene by keeping the vehicle in its lane until it is safe to steer the vehicle over to the side of the road and bring it to a stop. Our near term goals are to demonstrate technologies which assist the driver with safe lanekeeping. This document describes the control subsystems including the sensing and control of steering, throttle and brakes, and their experimental validation on a full-size truck tractor.

We use a Differential Global Positioning System (DGPS) to sense position and velocity since it appears that this method offers adequate precision with a low enough infrastructure cost to make the system practical in most rural settings. A basic set of lateral and longitudinal control algorithms are developed and implemented on a standard PC compatible computer. Experiments performed with a full sized (class 8) truck tractor are documented. These experiments were performed at the Minnesota Road Research Project (Mn/ROAD) test track, a pavement testing facility run by the Minnesota Department of Transportation (Mn/DOT).

The longitudinal (throttle and brake) controller is somewhat limited in performance due to the fact that we currently are not sensing which gear the transmission is in. Nonetheless, it is shown that a proportional-derivative (PD) control algorithm with the addition of a feedforward term is able to maintain speed within one or two miles per hour of the commanded velocity.

This is quite adequate for the intended purpose of automatically driving the vehicle long enough so that it can be brought to a stop at a safe place on the shoulder.

A number of lateral (steering) control algorithms were implemented and tested. The lateral position error is calculated based on the difference between the current DGPS sensed position of the truck and the lane centerline determined from a previously acquired map database. We examined both *look down* algorithms (which use no preview of the road ahead) and a pursuit algorithm. To both algorithms we added a damping term which feeds back the position of the steering wheel. The lookdown lateral controller did not do well in entering sharp curves. The *pursuit algorithm* that we examined uses successive DGPS position readings to calculate the heading of the truck. By using the heading calculated in this manner, along with the lateral position, and the position of a point on the desired path ahead of the truck, an angle was computed that steered the front wheels toward the point on the path ahead. The addition of the damping term based on steering wheel position eliminated a tendency towards instability at higher speeds. This controller performed well over a large range of speeds on straight sections and while negotiating curves, but was limited to 50 mph due to test track constraints.

The steering algorithms that we examined did not use a rate gyro to sense the yaw rate of the vehicle. In this report, we intentionally avoided the use of a rate gyro in order to determine the achievable performance with a lower cost suite of sensors. It may become necessary to use a gyro to help stabilize the steering controller in the future when our experiments reach speeds in excess of 50 mph.

Two applications were developed. The first was a *virtual rumble strip* that gives the driver haptic feedback through the steering wheel when the vehicle leaves the lane. The virtual rumble strip vibrates the steering wheel in a manner similar to that caused by the real rumble strips installed on the shoulder of some highways, but no attempt is made to steer the vehicle. A virtual rumble strip, which does not require physical modification to the pavement, can be incorporated while driving on undivided highways to keep drivers from drifting into oncoming traffic.

The second application monitors the steering performance of the human driver and actively intervenes if the system senses a prolonged period of erratic driving. A confidence factor is proposed for triggering intervention. The system then activates the automatic lateral and longitudinal controllers which keep the vehicle in its lane until it is steered over to a safe area on the shoulder of the road and parked.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Objectives—The SAFETRUCK Program

The research described in this study involves the lateral (steering) and longitudinal (throttle and braking) control of a full size truck tractor. The primary position sensor we use is a Differential Global Positioning System (DGPS) receiver. The motivation for this work is the fact that, according to the National Highway Traffic Safety Administration's Fatality Analysis Reporting System, one fourth of all 1997 fatalities in the United States were the result of single vehicle run-off-the-road crashes where the driver has either fallen asleep or lost control for some other reason. Our current work and the topic of this report is to demonstrate that real time DGPS can be used successfully in a control system to keep a vehicle in its lane and, if necessary, pull it over to the shoulder of the road and bring it to a safe stop. The problem of sensing when exactly the driver has lost the ability to safely drive is beyond the scope of this report. We are not proposing to drive a vehicle autonomously if the driver is capable and alert, but rather to add a backup system that would operate in cases where the driver has for some reason become impaired and is unable to control the vehicle. We are also concurrently involved in collision avoidance research using a concept called the Virtual Bumper [Schiller 97]. The controllers we develop in this report are also used to execute commands generated by the Virtual Bumper.

There are a number of institutions doing research in the area of autonomous vehicle control. These researchers have used position sensors ranging from computer vision systems to magnetic nails embedded in the pavement. None of these sensing systems, however, are well suited for deployment in Minnesota. Magnetic nails are expensive to install in the pavement and may create road maintenance problems when exposed to repeated freeze-thaw cycles. Computer vision systems would not be able to operate when visibility is obscured by snow. We believe that DGPS provides an accurate sensing technology, unaffected by poor visibility, that can be implemented with a relatively low infrastructure cost and is therefore one of the most promising positioning sensing systems for intelligent transportation systems.

In the work described here, we assume the truck is alone on the road. Other related research on radar and DGPS communication presently underway in our group would address the issue of collision avoidance with other vehicles on the road.

The goal for the work described in this paper has been to develop a basic control system using a minimal cost suite of sensors and the simplest control algorithms possible. This system will then be used as the foundation for further research and will be enhanced as experimentation with SAFETRUCK continues.

1.2 Report Outline

This introductory chapter concludes with an overview of the facilities used to perform experiments: the Minnesota Road Research Project (Mn/ROAD) test roads, the Navistar truck tractor used as a test vehicle, and the DGPS receiver used as a position sensor. Chapter two deals with control of longitudinal velocity, starting out with an analysis of some complications associated with using successive DGPS position measurements to calculate speed, and continuing through the development of throttle and brake control algorithms. Chapter three covers the implementation of several of steering control algorithms. Chapter four describes the hardware and software used in the experiments. Finally, chapter five discusses two possible applications of the sensing and control technology developed in the earlier chapters.

1.3 Minnesota Road Research Project

The experiments described herein were performed at the Mn/ROAD test road facilities of the Minnesota Department of Transportation (Mn/DOT) in Otsego, Minnesota. Operated by the Office of Minnesota Road Research, this facility, which opened in 1994, is a pavement testing facility with two separate test roadways. One of these is a 3.5 mile long section of freeway parallel to I-94 that regularly carries all northbound I-94 traffic. The other roadway is a 2.6 mile long loop called the Low Volume Road that is not open to the public and is therefore a location for the experiments with a computer controlled truck.

The Low Volume Road, shown in Figure 1.1, is a two-lane road consisting of two parallel straightaways. Each straightaway consists of several 500 ft long segments constructed with different types and thicknesses of asphalt, concrete, and gravel. The straightaways are connected with loops consisting of an 82 degree curve in one direction and a 262 degree curve in the other direction. The radius of all curves is 275 ft at the centerline. These curves are fairly sharp and are atypical of most rural roads. The abrupt changes in curvature at the points labeled a, b, c, d, e and f in Figure 1.1 present a significant challenge for the lateral control algorithm.

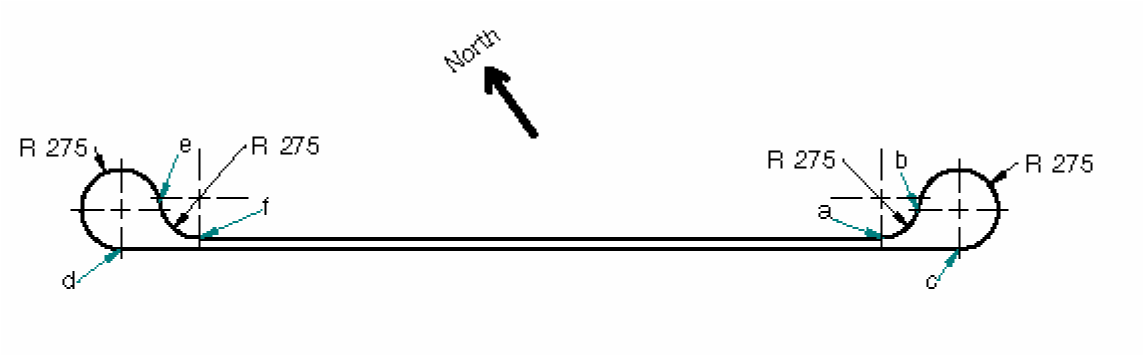

Figure 1.1: Layout of the Low Volume Road test track at the Minnesota Road Research Center.

1.4 International Navistar 9400 Experimental Testbed

The vehicle we are experimenting with is a full sized (class 8) truck tractor designed to pull a semitrailer hauling freight over long distances. It is manufactured by the Navistar International Transportation Company in Fort Wayne, Indiana. We will refer to this vehicle as *SAFETRUCK*. It is a conventional truck tractor layout with one steering axle in front and two tandem load carrying axles with dual wheels in the rear. A *fifth wheel* hitch is positioned over the tandem axles and is used to connect the tractor to a semitrailer.

The truck is powered by a Caterpillar 3406B six cylinder, 400 hp diesel engine controlled by Caterpillar's Programmable Electronic Engine Control (PEEC) system. The PEEC system uses a five volt 600 Hz pulse width modulated signal to control throttle position.

The transmission is the one component on our test vehicle that would not normally be seen on a typical over-the-road truck. In order to reduce the complexity of our control mechanisms, we specified a six speed Allison Heavy Duty automatic transmission instead of the more typical 13 speed manual transmission most class 8 trucks use.

We have modified the electrical system of the Navistar rig to accommodate the extra power requirements of our experiments. The stock 12 volt alternator has been replaced with a much larger 28 volt unit. This higher voltage is used to power an inverter that furnishes 120 volts AC and a voltage splitter that supplies the 12 volts used to charge the stock truck batteries and to power the normal electrical functions of the truck. A separate set of two 12 volt batteries connected serially supplies the inverter when the engine is not turning the alternator. This setup provides a range of AC and DC voltages with which to power our experiments.

1.5 Differential GPS

The Global Positioning System (GPS) consists of 24 earth orbiting satellites and a set of monitoring and control stations on the ground. The satellites orbit at an altitude of 10,898 nautical miles circling the earth approximately once every 12 hours [Novatel 95]. At any given time, there are between 6 and 12 satellites above the horizon at any point on the earth's surface. Each satellite broadcasts a stream of data that includes the precise time that the signal left the satellite. A receiving station on the ground can calculate the distance to a satellite by processing the signal from that satellite. If the distance to several satellites is known, then the position of the receiver can be calculated, since the position of the satellites are known. For a variety of reasons (timing errors, propagation delays in the ionosphere, the deliberate dithering of the signals by the military referred to as selective availability, etc.) the accuracy of this calculation (called a *single point fix*) is limited. Errors on the order of fifty meters are typical for a single point fix.

To obtain the much greater accuracy needed to control the position of a vehicle on a road requires the sophisticated implementation of a simple idea. That simple idea is to add another ground based receiving station (the monitor station) at a known location in the same general area as the receiver at the location being determined (the roving station). The monitor station compares the location it calculates from the satellite signals with its known location and determines the difference between them. This position error or *differential correction* is then transmitted to the roving station to correct the position that it is calculating based on the same satellite signals. Using a differential correction based on analyzing the phase of the GPS carrier frequencies, (usually referred to as carrier phase correction), the currently used Novatel RT-20 DGPS is typically capable of dynamically tracking the location of the Navistar truck tractor to well within 20 cm [Bodor et al. 97]. In order to achieve this level of accuracy a Real Time Kinematic (RTK) method is used. More on this subject can be found in [Novatel 95].

CHAPTER 2

Longitudinal Control

2.1 Overview

This chapter describes the control of throttle and braking systems on the SAFETRUCK experimental testbed. There are two goals we wish to accomplish here. First of all we need to be able to regulate velocity when the SAFETRUCK has assumed automatic control and is in the process of "searching" for a safe place to park and when it subsequently pulls over to the side of the road and stops. We expect that a digital map on board the vehicle would identify such safe areas for pulling off the road. The second objective is to develop low level hardware and software that will be used in the future by the higher level software of the Virtual Bumper. In both cases the quantity directly controlled by the system we describe here will be the longitudinal velocity, not the longitudinal position, of the truck.

2.2 Literature Review

Over the last 30 years or so there has been an extensive amount of research published on the lateral and longitudinal control of vehicles, but the quality of the information reported is sometimes less than what one might hope for. Shladover [Shladover 95] presents an excellent overview of the literature over these years (up to 1995) and also gives this caveat:

> "There is a serious paucity of real vehicle test data reported in the literature. Most papers have been based on analyses and computer simulations, without experimental verification. This is in large part because of the serious investment of time and resources needed to implement AVCS experimentally, but nevertheless it significantly limits the usefulness of much of the literature. Even the papers that include some experimental results tend to be rather vague about quantitative values, and rely more heavily on plots of test results that give the reader only a

> qualitative sense of what has been accomplished. This makes it difficult to assess the true value of the reported results or to compare the results achieved by different researchers. Because of proprietary concerns, most of the existing experimental data have not even been reported publicly."

There are two additional reasons why a large part the longitudinal control literature is not pertinent to the work reported in this paper.

First, we do not currently have the ability to sense any of the internal states of the engine and transmission that would enable us to utilize an analytical longitudinal model of the Navistar's dynamics. In the future, we hope to upgrade and then gain access to the vehicle data bus, which should at least let the control computer determine which gear the transmission is in. This information along with the sensing of manifold pressure, etc. will enable us to design a more sophisticated throttle controller.

Second, most of the recent work done in the field of longitudinal vehicle control concentrates on the difficult problem of maintaining stability in a platoon of several vehicles traveling at high speed with very small intervehicle spacing. Since the focus of this paper is on controlling the speed of a single vehicle, we have a much simpler problem to solve. Concurrent work on the virtual bumper concept here at the University of Minnesota deals with the interaction of several vehicles and will be reported separately.

In the early 1970's [Garrard and Kornhauser 73] designed a longitudinal controller for an automated transit system. They used a first order lag to approximate the power unit (engine and brakes) of a transit vehicle and then optimized the control law that provided input to that power unit by minimizing a quadratic performance index. They used headway error, velocity error, acceleration error and rate of change of propulsive force as the system states that were minimized in the optimization. After the control law had been calculated, they then assumed that the measurement of acceleration error and of rate of change of propulsive force would be difficult to measure so they designed an observer to estimate these quantities from measurements of position and velocity errors. Their simulations showed that the controller they designed would satisfactorily control the vehicle during merging maneuvers, emergency stops, and normal mainline operation.

[Hedrick et al. 91] describe a nonlinear throttle and brake controller designed using a multiple surface sliding control method. In simulations of a two- car platoon, the controller kept the spacing error to within 4 cm. Simulations of a four-car platoon revealed that the spacing error was amplified toward the rear of the platoon. This problem was remedied by communicating the lead car's velocity to the rest of the platoon and using it as a feedforward term in their controllers. This strategy resulted in attenuation of the spacing errors rather than amplification. In [Rajamani et al. 98] a high level sliding surface controller similar to the one above was augmented with a low level controller designed using feedback linearization techniques. This low level controller utilized an experimentally derived map that used manifold pressure and engine speed to predict the engine torque for a given throttle setting. The resulting controller was successfully used at the National Automated Highway Systems Consortium demonstration in San Diego in August of 1997, where it controlled a platoon of eight passenger cars performing maneuvers typical of those required on an automated highway system.

[Fancher et al. 93] used typical tractor-semitrailer parameters to simulate a proportional-integral (PI) throttle controller maintaining headway. No brake control was attempted. The simulation showed that the PI controller performed adequately for gross vehicle weights from 30,000 lb to 80,000 lb.

[Xu and Ioannou 94] designed an adaptive throttle controller to compensate for changes in the load carried by a passenger car. They used a simple first order model for the longitudinal dynamics (assuming that the car was in a “fixed gear state”) to design the controller. Experiments were performed on an actual vehicle and the results showed almost no steady state error.

More recently, researchers at UCLA have investigated the problems associated with controlling heavy vehicles that have delays in their brake actuation systems. Due to the long distance the pneumatic control signal has to travel to get to the valve that controls the rear brakes on a semitrailer, there is a pure time delay of about 200 milliseconds followed by a roughly first order lag with a rise time of about 300 milliseconds [UMTRI 1997]. In [Yanakiev and Kanellakopoulos 96] the UCLA group present an adaptive proportional-integral-quadratic (PIQ) control law. They attempt to control the headway in a platoon of trucks in which they stabilize the platoon by using a variable time headway and a variable separation error gain in the calculation of the error that their PIQ controller tries to minimize. They define this error as the sum of the relative velocity between the leading and following vehicle and a design constant (the separation error gain) times the separation error. In order to accommodate the actuation lag in the air brake system they present an adaptive backstepping controller in [Yanakiev and Kanellakopoulos 97a]. The backstepping controller is able to successfully control the platoon using smaller headways that the original PIQ controller. They further enhanced their controller in [Yanakiev and Kanellakopoulos 97] by adding a first order linear model, to approximate the vehicle dynamics, which functions as a predictor of the errors in velocity and intervehicle spacing. Simulations showed that the controller with the predictor allowed smaller headways and gave smoother control during normal driving, but was less robust during extreme maneuvers.

In [Rajamani et al. 97] the problem of overall system reliability is addressed. A diagnostic system is designed that uses a set of nonlinear observers to identify faults in the sensors and actuators used in a longitudinal control system. Combinations of sensor readings and observer estimates are processed and it is shown that any individual faulty component can be identified.

2.3 Sensing speed with GPS

A sensor in any real time control system must meet at least three criteria. First, it must be accurate enough to give the control algorithms a good estimate of the error between where the system is and where it has been commanded to go. Second, the output of the sensor must come at a fast enough rate so that the state of the system does not change too much between samples. Third, there should not be too much internal delay (latency) between the time the sensor measures the system and the time that it reports that information to the controller. In this section we will evaluate how well DGPS fulfills these three requirements.

Most real time GPS systems have the internal ability to calculate speed. The Novatel RT-20 system we are using, for instance, has two different output logs that include velocity and heading. Unfortunately these logs do not also include position and for the lateral controller described in the next chapter a reliable, periodic and accurate position signal is of paramount importance. For this reason we use Novatel's P20A log which, according to the manufacturer, gives the best available position computed by the receiver. We then take the distance between two successive DGPS readings and divide by the time between the readings to compute the velocity of the truck. The static and dynamic performance of the RT-20 system has been documented in earlier work we have done [Bodor et al. 97] and [Bajakar et al. 97], however there are two aspects of this DGPS signal that were not covered in that work that are important to consider before we use the signal to control a vehicle.

The first, and most important from the standpoint of calculating velocity, is the length of the time period between DGPS readings. Since we are using this time period to calculate velocity we need to make sure that it correlates properly with the position measurements. The Novatel RT-20 is rated by the manufacturer as being able to provide data at 5 Hz. As Figure 2.1 shows, however, the period between successive readings arriving at the serial port of the control computer from the DGPS receiver is not always close to 0.2 second when the receiver has been initialized to provide data at this rate. To make sure this variability was the result of latencies in the DGPS system not the control computer, another computer was programmed to send out a simulated DGPS string at a regular rate and the timing test shown in Figure 2.1 was repeated. With all the control tasks running on the control computer the largest difference in period measured was approximately one millisecond as opposed to over fifty milliseconds using the real DGPS signal. There is clearly a variable latency in the RT-20 signal when it is running in continuous mode.

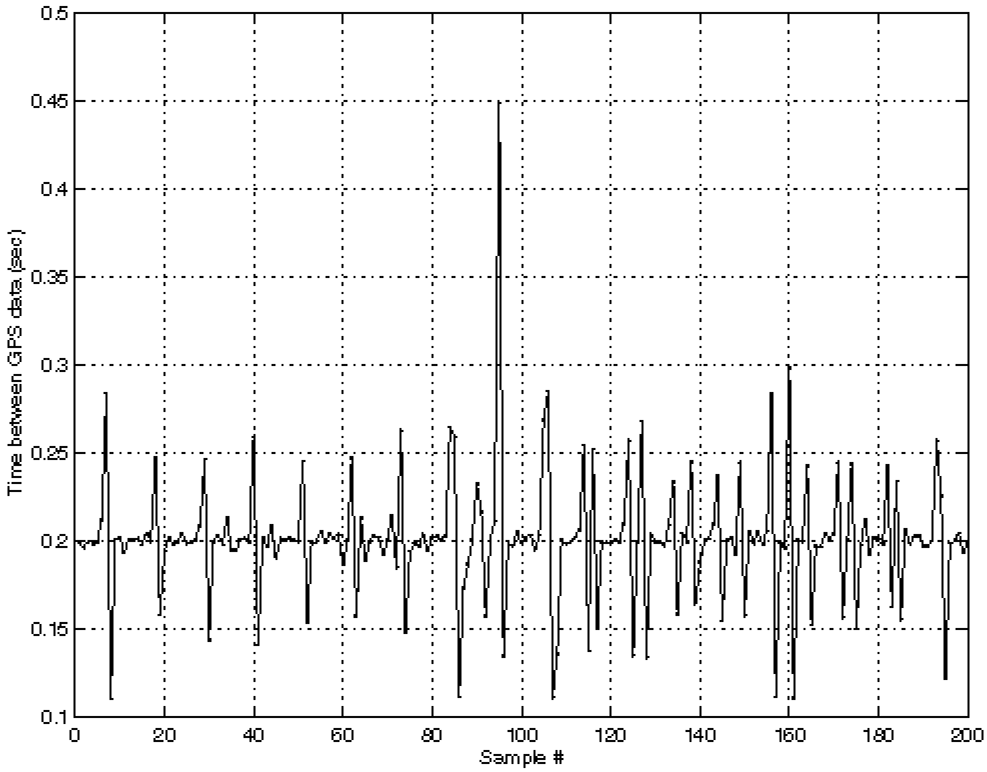

Figure 2.1: Variable time interval between "5Hz" RT-20 position signals for 200 consecutive samples.

The string of data from the DGPS receiver has, in addition to the position measurements, a time stamp that increases in increments of exactly 0.20 seconds if no readings have been lost. At sample 95 in the Figure 2.1 there has been a long enough delay in calculating a position that one entire reading has been skipped. The DGPS time stamp will increase by 0.40 seconds increment in this situation.

In any case, this internal time stamp correlates with the time that the DGPS unit calculates its position. It is the difference between these internal time stamps that should be used to calculate velocity, not the difference between the time that successive positions actually arrive over the serial port from the RT-20 system.

The second property of the RT-20 signal we need to keep in mind concerns the “age” of the differential correction. The Novatel RT-20 system computes a differential correction in real time at a rate of about once every two seconds. This can be seen in Figure 2.2 where the correction typically ages for 8 or so samples before resetting. As the correction ages the positional accuracy is diminished. This results in a fairly regular pattern of noise in the signal at the same rate that the correction is updated. This roughly 0.5 Hz error pattern in the DGPS signal must be considered in the design of any digital filters used in the system.

2.4 Throttle Control

2.4.1 Nonlinearities in Throttle Response

In spite of some significant nonlinearities in the response of the truck to changes in throttle position, the throttle turns out to require the least complicated algorithm of all the subsystems we are investigating here. The nonlinearities arise from the fact that the Caterpillar 3406B PEEC engine in the International Navistor 9400 has some internal control algorithms of its own, and are due to the coupling between these internal engine control algorithms and the Allison automatic transmission. Figure 2.3 graphs the steady state speed attained at a range of throttle settings on a level road without a trailer attached. The bottom line on the graph (marked with x’s) shows the steady state velocity attained as the throttle is increased. The speed increases linearly from about 4mph at idle (0%) to 10 mph at a throttle setting of 40%. At the 40% position the transmission will automatically shift into second gear creating a slightly greater load on the engine. The PEEC engine responds by increasing its torque output even though the external control signal applied is constant at 40%.

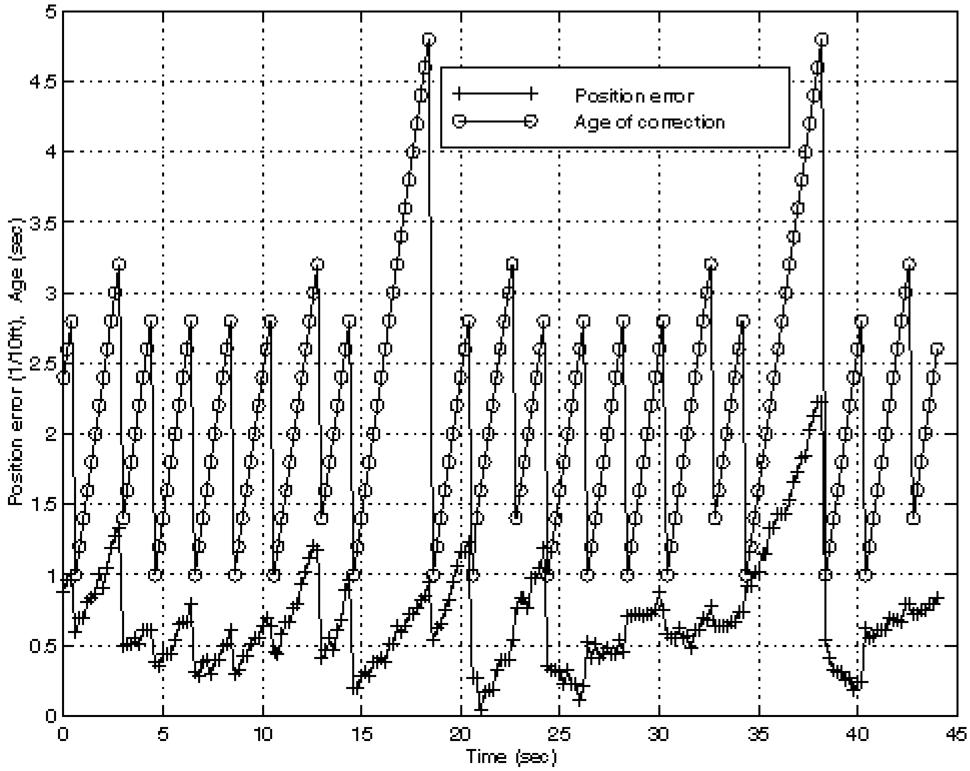

Figure 2.2: The effect of the *age* of the differential correction on positional accuracy. The varying amount of time it takes to calculate a differential correction is due to various real world disturbances such as multipath signal reception. As the correction ages, the accuracy of the position fix is reduced.

This acceleration is represented on the graph as a vertical line from 10 mph to 52 mph at a throttle setting of 40%. From this point the throttle was reduced and the points along the top line in Figure 2.3 (marked with small o's) were recorded as the truck slowed down. The throttle remained at a constant value until the truck achieved a steady state velocity for each data point. Although the test track was not long enough to record the complete cycle in one test run, the results would be the same if it was long enough.

The difference in velocity for identical throttle settings is partly a result of a reduction in friction in the engine and transmission when the transmission is in a higher gear at a given speed. In this test the truck tractor was not pulling a trailer so this internal friction consumed a large fraction of the total power required to keep the vehicle moving. We would expect less difference when the truck is pulling a loaded trailer, but that experiment will have to wait for a test on a road long enough to accelerate to test speeds with a heavy load.

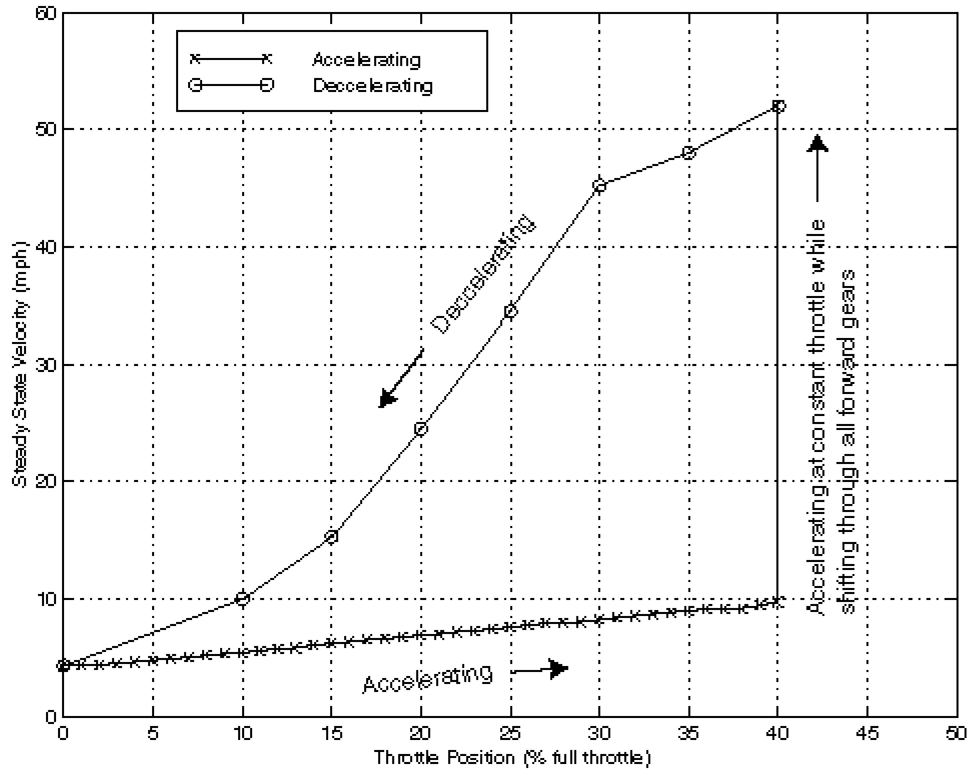

Figure 2.3: Terminal velocity reached at various throttle positions.

In spite of this large nonlinearity a slightly modified proportional-derivative (PD) control algorithm with one feedforward term does an adequate job of tracking velocity commands. The modification involves the derivative term which we compute from the change in velocity of the truck from one sampling period to the next, not the change in the velocity error between two samples as is typical for a PD controller. This minor modification eliminates a quick burst of throttle when there is a step increase in the velocity command. We discuss the feedforward term in the next section. The throttle control loop runs at the same speed as the DGPS system, nominally 5 Hz, but with the same variability in timing as shown in Figure 2.1. No ill effects have been observed associated with this slightly irregular control loop timing. We have also placed a maximum limit on throttle position of 50% since most of our experiments are done without a trailer attached and half throttle is more than enough.

2.4.2 Throttle Feedforward Term

A feedforward term can be used if we know in advance that a change in a command to a control system will require a corresponding change in the control system's output and we have some idea of the relationship between the change in the command and the required change in the output. In the case of our throttle controller we know that, in the absence of other disturbances, a higher velocity command will require a higher throttle setting. What we need is a function relating velocity to throttle position and that is just what we have in Figure 2.3. If we switch the X and Y axes around and assume that the transmission is already in the appropriate gear for the desired velocity then the set of data points represented with o's on the graph suggest a function that will map velocity commands to throttle position. Figure 2.4 shows a third order curve fitted to these points. The equation for this curve is

$$y = 0.000870x^3 - 0.076619x^2 + 2.585309x - 9.569971$$

where y is the throttle command (percent of full throttle) and x is the desired velocity in miles per hour. Although no data has been taken at more than 40% throttle (the test track is not long enough to come to a steady state speed between curves), the curve does intersect the 100% throttle position close to the actual top speed of the Navistar. It is therefore likely to be a good fit for the entire range of throttle travel.

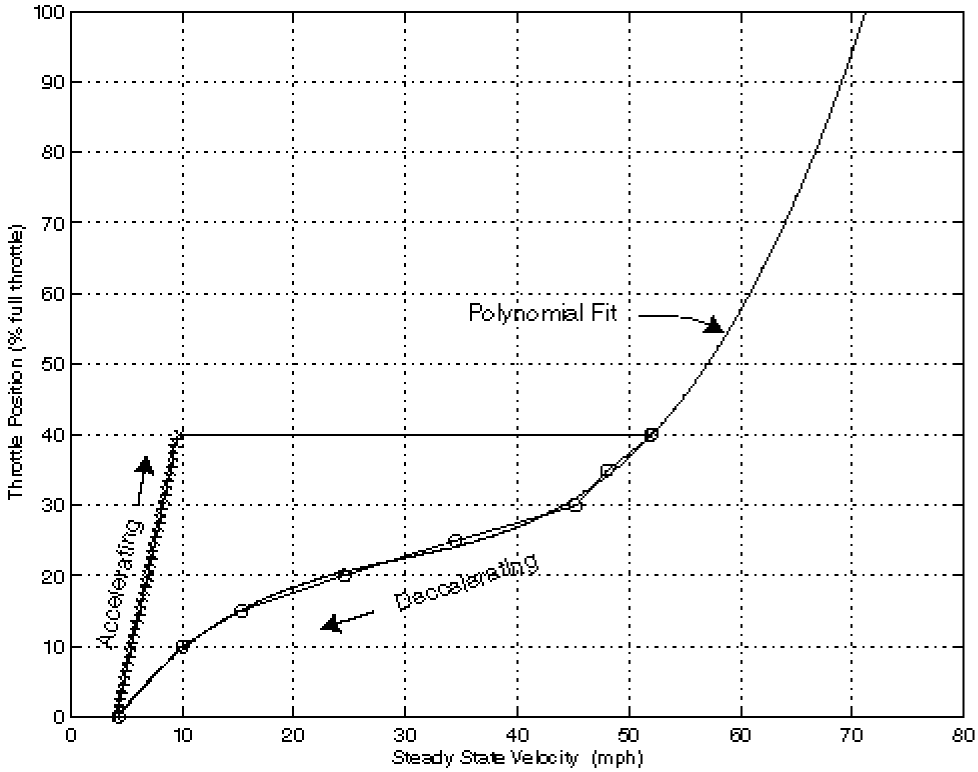

Figure 2.4: Throttle required to maintain a range of forward speeds. This is the same data as Figure 2.3 with the axes reversed. The curved line through the right most data points is a third order curve fit that provides us with a throttle feedforward term.

2.5 Brake Control

The air brakes installed on modern heavy duty trucks are sophisticated and complex pneumatic systems that reliably convert air pressure into the mechanical force required to quickly decelerate many tons of rolling machinery. Although the details of an air brake system are many and varied, the basic idea is that the brake pedal sends out a pneumatic signal to a valve positioned close to the brake actuator. This valve meters air pressure from a supply tank that is also in close proximity to the actuator. The pressure from the supply tank pushes a piston in the actuator that either turns a cam or pushes a wedge that forces the brake shoes into contact with the brake drums. With this arrangement only a relatively small volume of air has to flow all the way from the valve at the brake pedal to the valve near the actuator when the brakes are applied. Since the much larger volume of air needed to move the piston in the actuator is stored close to the actuator, the brakes can respond much faster than they otherwise would. There still is, however, a noticeable delay between the time the pedal is pushed and the brake shoes are activated. Typical delays are on the order of three tenths of a second for brakes on the tractor, and six tenths of a second for the more distant brakes on the trailer [UMTRI 1997].

In order to actuate the brake system automatically without interfering with its manual operation we have added a linear actuator that pulls a cable attached to the back of the brake pedal. The linear actuator consists of an electric motor driving an acme threaded screw. A timing belt connects the motor to the screw. Although this setup avoids having to modify the internals of the factory brake system (and incurring the attendant liability), it has the disadvantage of introducing a large amount of stiction into the system that makes precise positioning of the brake pedal difficult. We experimented with several methods of overcoming this stiction, including adding dither to the control signal, sending a voltage spike when the actuator was stuck in a position away from the commanded position, and using feedforward term to compensate for the progressive resistance of the brake pedal return spring. In the end, the most successful approach was a combination of these along with an outer control loop that sends step commands to an inner PD loop that does the positioning. The outer loop is, like the throttle controller, synchronized with incoming DGPS position data at a nominal 5 Hz. This outer loop compares the velocity command with the velocity computed from DGPS position data and, based on the difference between them, picks a brake position command out of a small table. This position command is sent to the inner brake control loop. The inner loop is a normal PD loop where the derivative term is a gain multiplied by the error between commanded and actual brake position. Since the outer loop sends a position command that changes in fairly large discrete steps, the derivative term generates a voltage spike at each new commanded position that helps “kick” the actuator into motion.

The linear actuator controlling the brakes is a relatively fast acting mechanism compared to the other parts of the system and it therefore requires a faster control loop. Our experience with this actuator has shown that there are stability problems at control rates under 10 Hz. The brake actuator system becomes stable at about 15 Hz with useful controller gains, and little further improvement in performance is noted at rates above 20 Hz. We therefore chose an inner control loop running at 30 Hz to make sure that the sampling rate does not limit the gains with which we can experiment.

2.6 Longitudinal Control Results

Figure 2.5 shows a block diagram of the combined throttle and brake controller. In order to assure that the brake and the throttle are not both on at the same time the lookup table in the outer brake control loop will not activate the brakes until the velocity command is at least 5mph less than the sensed speed of the vehicle. Another nonlinearity that the lookup table must accommodate is the fact that the transmission and torque converter create quite a strong forward force when the velocity is close to zero and the transmission has shifted into first gear. Since we currently can not directly control the transmission, the brake position must be increased beyond that required to slow the truck from higher speeds in order to bring the truck to a complete stop.

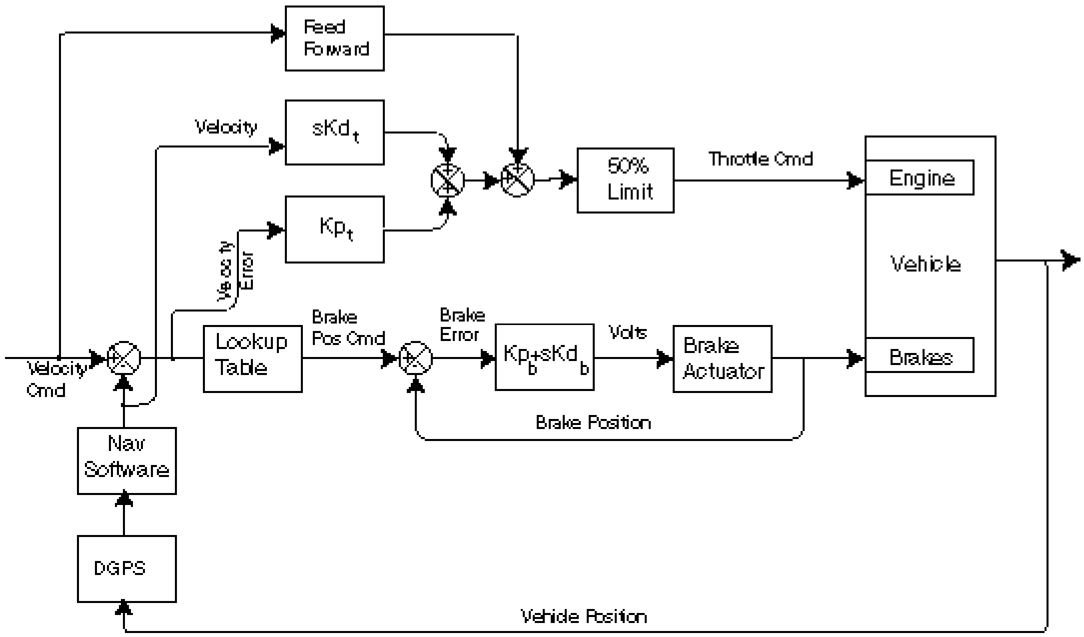

Figure 2.5: Block diagram of the longitudinal control system

The velocity command originates from a higher level task such as the Virtual Bumper or possibly from a speed limit in the map database that also contains the location of the lane data used by the lateral controller. Figure 2.6 shows the response of the system to a velocity trajectory input from the keyboard of the control computer. The top graph shows the actual speed of the truck following changes in the velocity command. The middle graph shows how the throttle controller responded with throttle position commands sent to the engine. The bottom graph shows the outer brake loop sending abrupt commands to the brake actuator and the position to which the actuator moved. Note that the brake actuator moves only a small distance between the position where the brakes are barely on and the position where maximum braking occurs. This data set also shows some dither added to the brake command when the velocity command is zero at the beginning and end of the experimental run.

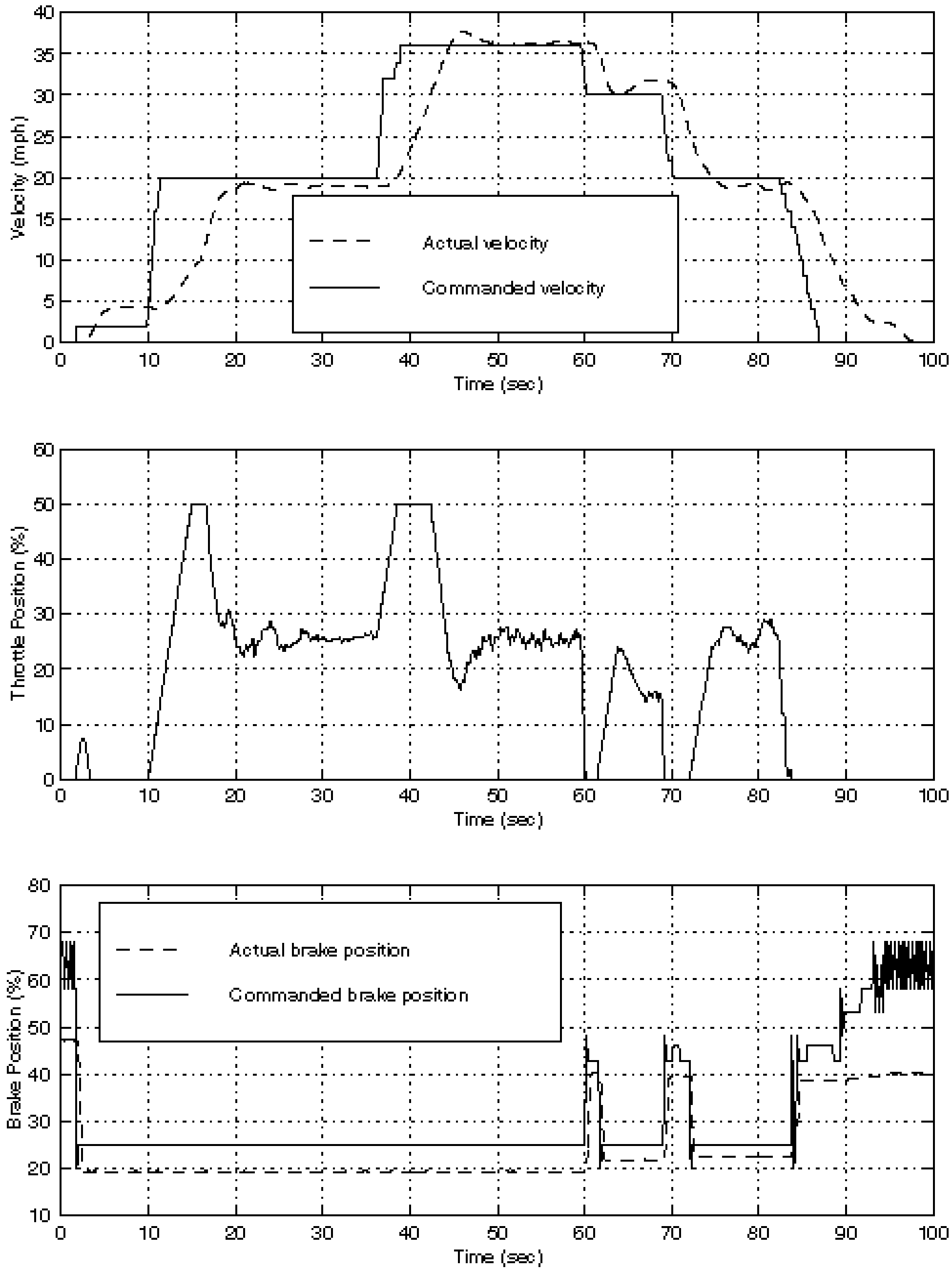

Figure 2.6: These three graphs show the throttle and brake controllers following a commanded trajectory

2.7 Conclusions

We have shown that it is possible to control the longitudinal velocity of a heavy vehicle using DGPS as the primary velocity sensor. Although a more complete set of sensors would be required for the more accurate positioning required to maintain the small headways involved in platooning, our current implementation will be adequate to automatically drive the truck to a safe parking spot and pull over and stop. In concurrent experiments we have used this controller to accept velocity commands from the virtual bumper and it appears to be adequate for that task as well.

CHAPTER 3

Lateral Control

3.1 Overview

In this chapter we will discuss control of the steering system. Once again we stress that the objective of the work done in this report was to use a low cost suite of sensors. There is therefore, no attempt to incorporate a yaw rate gyro into the system. After a brief review of related work done by other researchers in the field we will examine the results of four fairly simple steering control algorithms. The first of these is a PD control loop acting on the lateral deviation of the vehicle from its commanded position in the lane. We refer to this as a *lookdown* controller since it does not use any information about the direction of the road ahead, just the information that a driver would acquire by "looking" straight down at the road beneath the current position of the vehicle. We will see that a pure look-down PD algorithm will keep the truck within the lane boundaries at low speeds, but there is a lateral oscillation that diverges as the forward velocity is increased. The second controller will use an additional damping term to reduce the lateral oscillation and allows the truck to maintain its lane position at higher speeds on straight roads. Entering a sharp curve, however, creates a disturbance that this look-down controller cannot handle. The third controller we investigate is the so called *Pursuit algorithm* where the vehicle is continually steered towards a point on the desired path ahead without explicit regard for its current position in the lane. We then combine the best features of the look-down and pursuit controllers into a hybrid approach. Finally, we will discuss the inner control loop that uses steering position commands generated by the above algorithms to actually move the steering wheel.

3.2 Literature Review

Lateral control of automobiles has been a topic of research since the late 1950's when General Motors and RCA experimented with a vehicle for what they called the "Electronic Highway" [Zworykin and Flory 58]. In the 1960's and 70's a substantial amount of work was done at Ohio State University where [Fenton et al. 76] documented a wire guided passenger vehicle controlled by an analog computer that accurately maintained lateral position at speeds of up to 80 mph. As is the case for longitudinal control, [Shladover 95] gives a comprehensive overview of the research done prior to 1995.

Most of the current work in automated highway vehicle control is being done by Partners for Advanced Transit and Highways in California(PATH). PATH researchers at Berkeley have been controlling vehicles using magnetic plugs in the pavement sensed by magnetometers on the front (and sometimes also on the rear) of passenger cars. [Peng and Tomizuka 90] describes their use of frequency-shaped linear quadratic (FSLQ) control theory with the addition of a feedforward term. The FSQL feedback loop design method explicitly includes the ride quality as a performance index. In this paper they use a simple feedforward calculation that does not include preview of the road curvature ahead. They estimate the cornering stiffness at the front and rear axles using a simple lateral model and a least squares algorithm with yaw rate data from a gyro and lateral acceleration data from an accelerometer.

In [Peng and Tomizuka 93] they add preview of the road ahead by combining the curvature of the approaching road with the superelevation of the lane to form an "effective curvature" and solve the linear quadratic control problem that minimizes the effect that the disturbance from the effective curvature creates. Most of the results in this paper are presented as simulations, but they do mention that experiments show that the vehicle can not track the lane without preview.

In [Pham et al. 94] more PATH supported work at Berkeley combined the lateral and longitudinal control of vehicles using a nonlinear method called surface shaping. Their simulations showed good tracking performance with the claim that minimal tuning of the control parameters was required.

Look-down reference controllers without preview are analyzed in [Guldner et al. 96]. This paper gives an excellent overview of the lateral control problem. They conclude that a look down controller will not be able to control a vehicle at highway speeds. In a subsequent paper [Patwardhan et al. 97] some of the same authors reiterate the limitations of lookdown controllers at higher speeds and propose a *virtual drag link* controller that is essentially the same idea as the *pursuit algorithm* we use at the University of Minnesota [Morellas et al. 97].

In simulations of the lateral control of a wire guided commuter bus, [Ackermann and Sienel 90] proposed sensing the yaw rate of the bus with a gyro and feeding it back to the controller. Their analysis showed that the resulting control system was more robust to changes in mass and velocity than the fixed gain controller they started with. Our experiments with the SAFETRUCK have also demonstrated the stabilizing effect of yaw rate feedback from a gyro [Morellas et al. 97]. Ackermann also assisted in the design of a sliding mode lateral controller in [Guldner et al. 94]. In [Ackermann et al. 95] he compared the nonlinear sliding mode controller to a linear controller designed using what they call a *Parameter Space Approach*. Their simulations showed that the nonlinear controller did a slightly better job of maintaining lateral position, but had more of a tendency to oscillate the steering wheel.

Most of the published work on vehicle lateral control makes use of the single track (bicycle) model of a four wheel vehicle developed in [Riekert and Schunck 40]. Recently we added one more axle to the rear of the bicycle model to represent the second drive axle of a tandem axle truck, and then attempted to verify experimentally that this three wheeled single track model had a steering angle to yaw rate response similar to our International Navistor 9400 [Alexander et al. 97]. The results were not encouraging and for that reason the controllers that are presented in this paper have been developed experimentally rather than analytically. One reason why we were not successful in matching the single track model to the experimental data is that we assumed that the compliance in the steering linkage from the steering wheel to the steered tires was negligible due to the obvious stiffness of the large links in the system. In [UMTRI 1997] it is reported that "despite the relatively large stiffnesses, the small angular deviations they permit may be quite significant with respect to vehicle directional response." We also used a fairly large steering input in those experiments (about 5.5 degrees at the front tires) and in [Roos et al. 97] they found that "The measurements have shown that a vehicle's response to a small steering angle input depends on the test conditions and that it differs from the vehicle's response to medium steering wheel angle inputs." Small steering inputs are, of course, what we use to maintain our position in the lane, so a model based on the response to small inputs might be more useful for lane keeping controller design. At this point we have no reason to disagree with [Guldner et al. 96] in their assessment that the Riekert—Schunck model is sufficient for normal steering using small steering angles as long as this angle is sensed at the front wheels, not at the steering wheel.

3.3 A Simple Look Down Controller

3.3.1 A Basic PD Controller

One of the simplest possible lateral control algorithms is a standard PD control law acting on the error between the actual and the commanded lateral position in the lane. Since we are using DGPS as our sole position sensor we have to deal with a lag that is on the order of 0.2 seconds in the feedback portion of our control loop. To keep the system stable with that lag, we need to have a controller with a fairly high damping ratio or in other words a relatively large derivative gain with respect to the proportional gain. Figures 3.1 and 3.2 graph the results of two different test runs with respective derivative gains of 1.0 and 2.0 radians per lateral ft per sec combined with a proportional gain of 0.4 radians per foot of lateral deviation. In both cases the system goes unstable when the forward velocity is increased to 20 mph. The test run with the larger derivative gain has smaller oscillations at 10 mph and 15 mph, but diverges faster at 20 mph. This tendency to lose stability at higher speeds is a result of the vehicle being more sensitive to steering input as speed increases (a given steering angle gives a faster lateral response). It is evident in Figure 3.2 that the derivative component contains a significant amount of noise at a gain of 2.0. In fact if we increase either the proportional or the derivative gains much beyond these values the system will be unstable even at 10 and 15 mph. Lower gains result in larger oscillations. This is a good demonstration of the tradeoff between sensitivity and complementary sensitivity and it means that we have reached a limit as to how fast we can drive with a simple PD controller acting only on the somewhat noisy lateral position information from our RT-20 DGPS system.

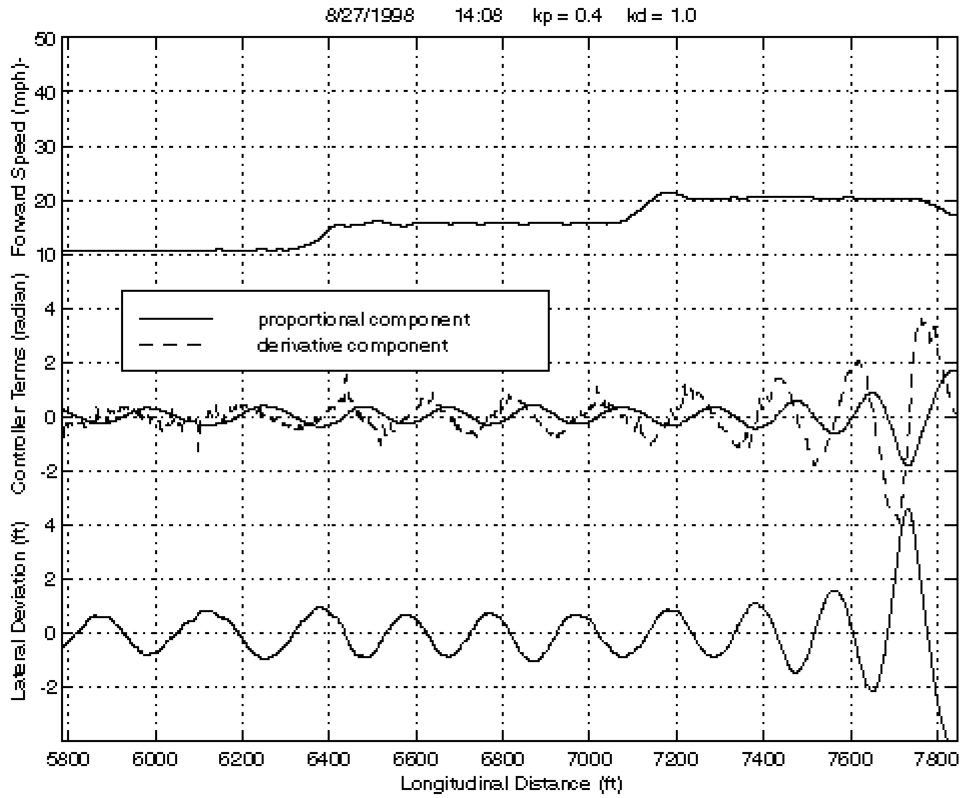

Figure 3.1: Lane keeping performance using a simple PD control law with a proportional gain of 0.4 steering wheel radians per foot of lateral deviation and a derivative gain of 1.0 steering wheel radians per foot/sec of lateral velocity. The top line on the graph is the forward velocity of the truck in mph. The middle set of data is the output of the proportional and derivative components in the controller. The bottom line on the graph is the lateral position of the truck in the lane. With this set of controller gains there was a fairly constant oscillation to about one foot either side of the centerline of the lane until a speed of 20mph was reached. At that speed the oscillations increased slightly for about 400ft then quickly diverged.

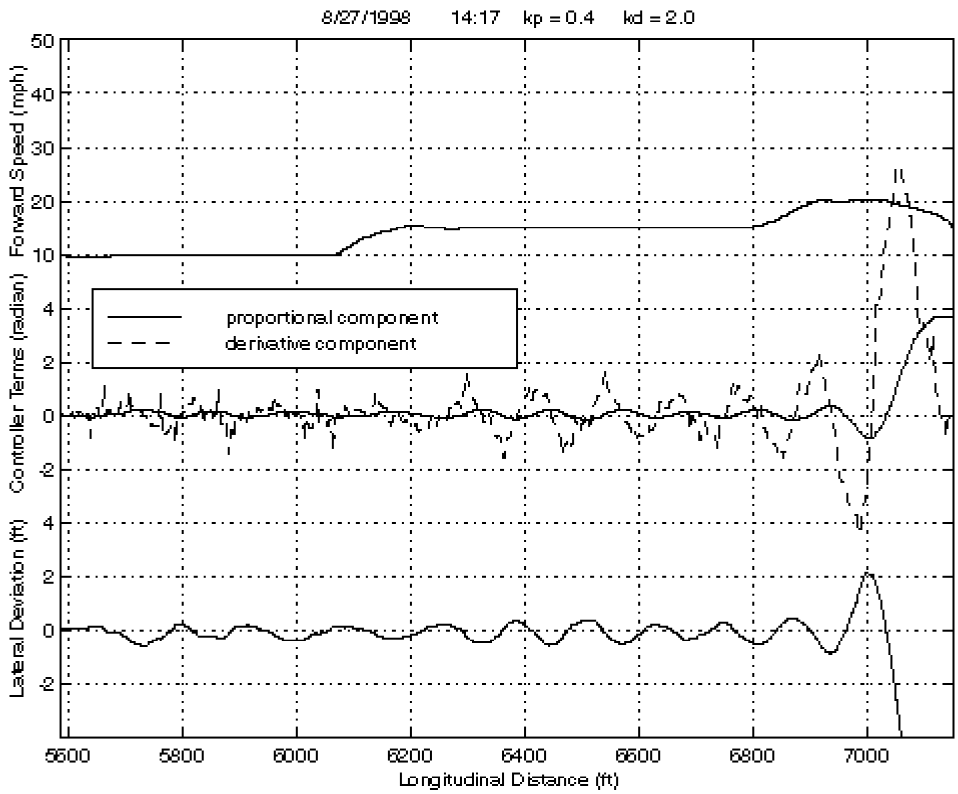

Figure 3.2: Same as Figure 3.1 except the derivative gain has been increased to 2.0 radians per ft/sec. The oscillations have been damped a little, particularly at 10 mph, but they diverge faster when the speed is increased to 20 mph. Note that the derivative components in the middle set of data is now dominated by noise spikes. If the derivative gain is increased any further these noise spikes destabilize the system even at lower speeds.

3.3.2 An Additional Damping Term

There is one simple enhancement we can make to the look-down PD controller without adding any additional sensors to our system that further damps oscillations yet is not affected by noise from the DGPS. Like the brake control system described earlier, the steering system requires an inner control loop. This inner loop is closed on the position of the steering wheel which is sensed by a relatively noise-free potentiometer mounted on the steering column. Since the position of the steering wheel correlates with the curvature of the path the truck is following, and the curvature of the path at a given speed is proportional to the lateral acceleration of the truck towards the center of the arc of that curved path, we can correlate steering position to the lateral acceleration of the truck across the road, as long as lateral deviations from the center of the lane remain small.

Having suggested that the steering wheel position is related to the lateral acceleration of the vehicle with respect to the lane, we can now attempt to use it as another damping term in addition to the derivative component in our PD controller. We will call it a DD term since lateral acceleration is the second derivative of lateral position. Like the D (first derivative) component that is used to attenuate the lateral velocity of the vehicle, the DD component will be used to attenuate the lateral acceleration.

Figures 3.3, 3.4 and 3.5 show the results that we obtain from test runs using the DD component in addition to the P and D components.

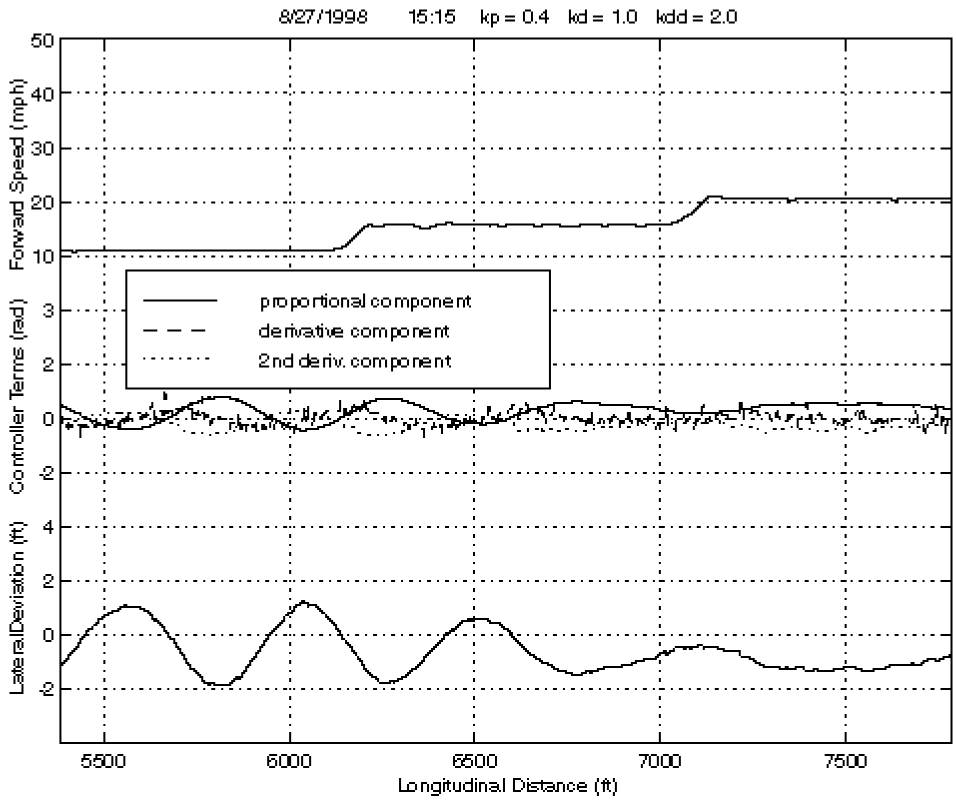

Figure 3.3: Low speed lane keeping performance using the PD-DD controller.Note that at 10mph the oscillations are larger than those of the PD controllers above, but they become smaller as the velocity increases. At slow speeds the truck responds slowly to steering input, but at higher speeds the truck is a faster, more sensitive system so the extra damping improves performance.

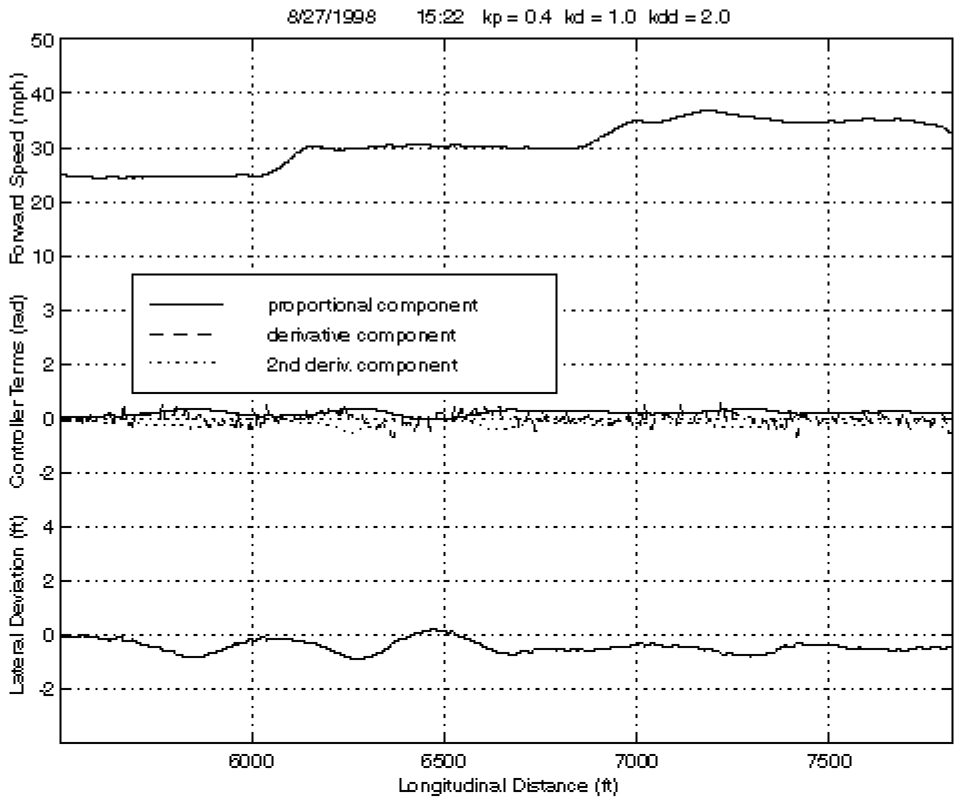

Figure 3.4: Medium speed lane keeping performance using the PD-DD controller. The oscillations are reduced as the speed increases. There is no tendency to destabilize as forward speed increases.

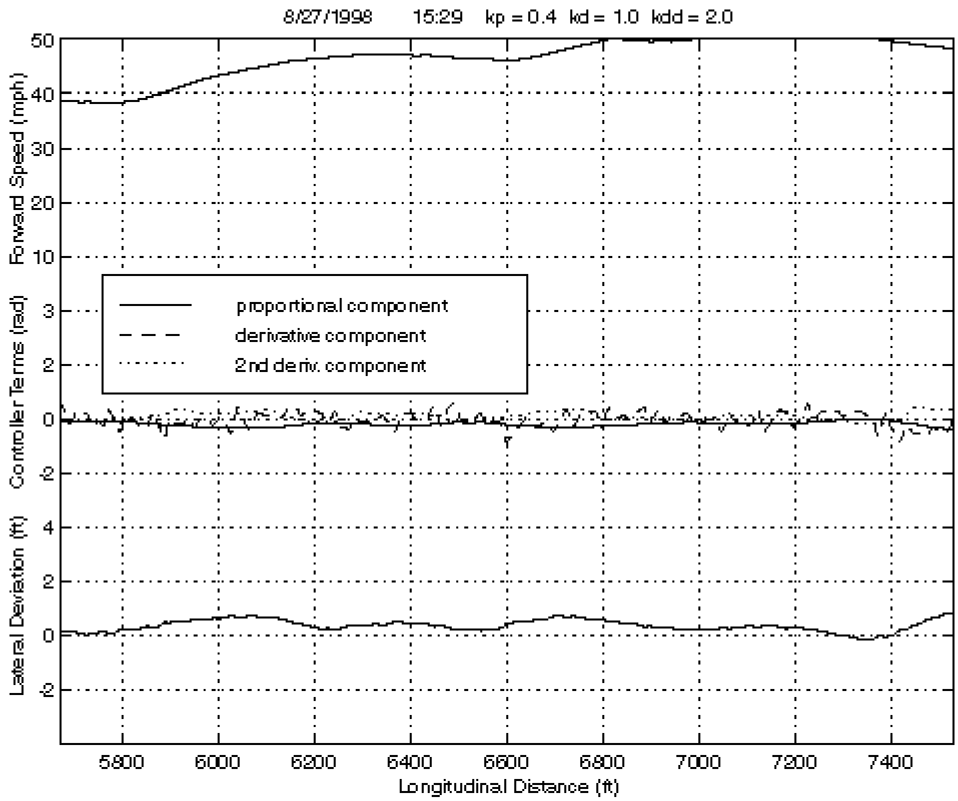

Figure 3.5: High speed lane keeping performance using the PD-DD controller. This graph shows that a lookdown controller is capable of steering a truck at highway speeds.

We should note that the correlation we are using between steering wheel position and path curvature is based only on the obvious kinematics of the vehicle. At higher speeds, the kinetics of the system become relatively more important with respect to the kinematics and we have to be more concerned with the effects of slip angle at the tires, compliance in the steering system, and other dynamic effects that are less important at lower speeds. Backlash in the steering system also becomes more significant as forward velocity increases since it limits the accuracy with which the sensed steering wheel position correlates with the position of the steered wheels on the front axle. As our research moves from the test track to the highway, it is possible that the DD component will have to be calculated using a yaw rate gyro and a speed signal instead of the steering position sensor.

Although the performance of our PD-DD controller is acceptable on straight roads, its performance entering curves is disappointing. The abrupt changes in curvature on the Mn/ROAD test track consistently sent this version of our steering controller into significant oscillations that no amount of tuning would mitigate. To enable the truck to enter curves smoothly, we need to look ahead of the truck to get a preview of what is coming. A lateral control algorithm that does this is discussed in the next section.

3.4 The Pursuit Algorithm—A Look-Ahead Controller

The principle of the pursuit algorithm is very simple: find a point on the desired path a certain distance ahead of the vehicle and turn the front wheels to point in that direction. Figure 3.6 illustrates the geometry of the pursuit algorithm. The pursuit algorithm is closely related to a PD control algorithm, but it has the advantage of smoothing out step changes in road curvature.

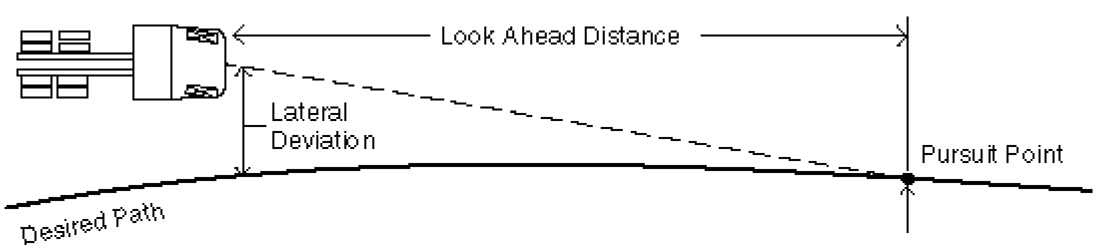

Figure 3.6: The Geometry of the pursuit Algorithm. The idea is very simple, just point the front wheels at the pursuit point which lies on the desired path ahead of the truck.

The pursuit algorithm, like the PD algorithm described at the beginning of this chapter, will keep the truck within its lane at slow to medium speeds, but there is always a significant oscillation and that oscillation will tend to diverge at higher speeds. Figure 3.7 shows the results of steering with the pursuit algorithm on a straight road at a forward velocity of 20 mph. Two different look ahead distances were used in this example. Shortening the look ahead distance corresponds closely to increasing the proportional gain in a PD controller. In Figure 3.7 we see that shortening the look ahead distance from 100 ft to 75 ft has the same destabilizing effect as raising the proportional gain too much in the lookdown PD controller.

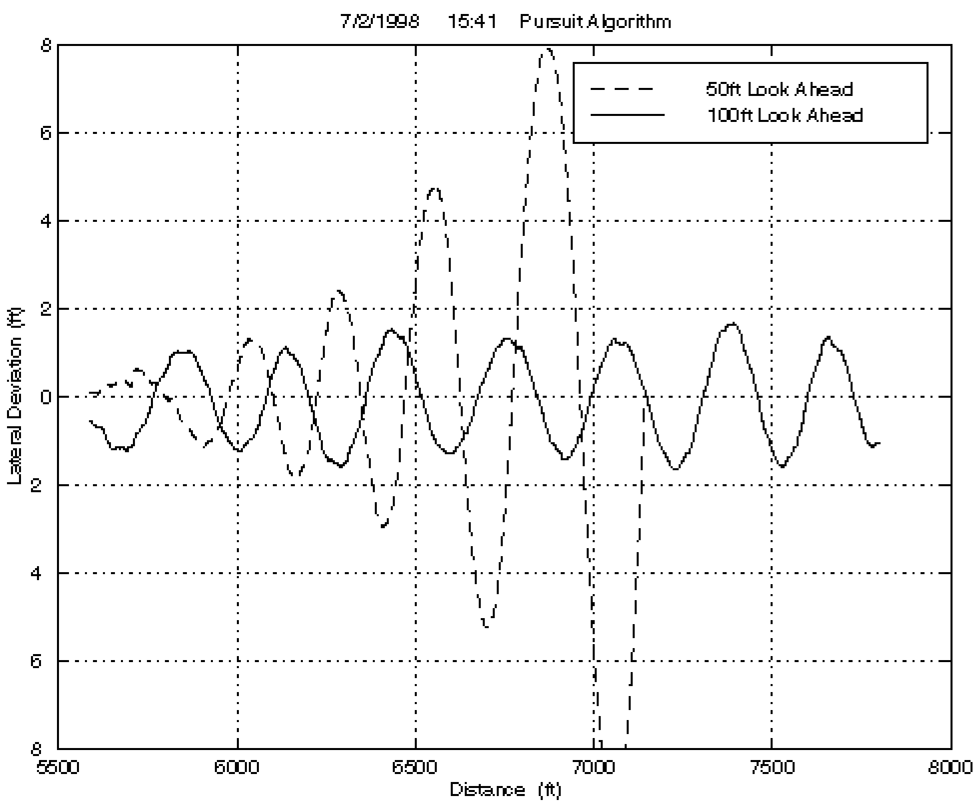

Figure 3.7: Lane keeping performance of the pure pursuit algorithm at two different look ahead distances on a straight roadway.

We now have one control law—the pursuit algorithm, that will allow the truck to enter and exit curves gracefully, and another control law; the PD-DD algorithm, that does a good job of steering the truck on straight sections of road. In the next section, we will combine the two together to create a controller that has the best qualities of both.

3.5 Combining Elements of the Pursuit and Look-down Algorithms

To build a hybrid controller from the pursuit and the PD-DD algorithms, we simply replace the proportional component in our PD-DD control law with the Pursuit angle calculation. Figure 3.8 shows the results of a test run on a straight road using a look ahead distance of 75 ft along with derivative and second derivative gains similar to those used in the PD-DD controller above. This algorithm controls the truck as well as the PD-DD algorithm on straightaways. We will now look at some of the complications involved in negotiating curves.

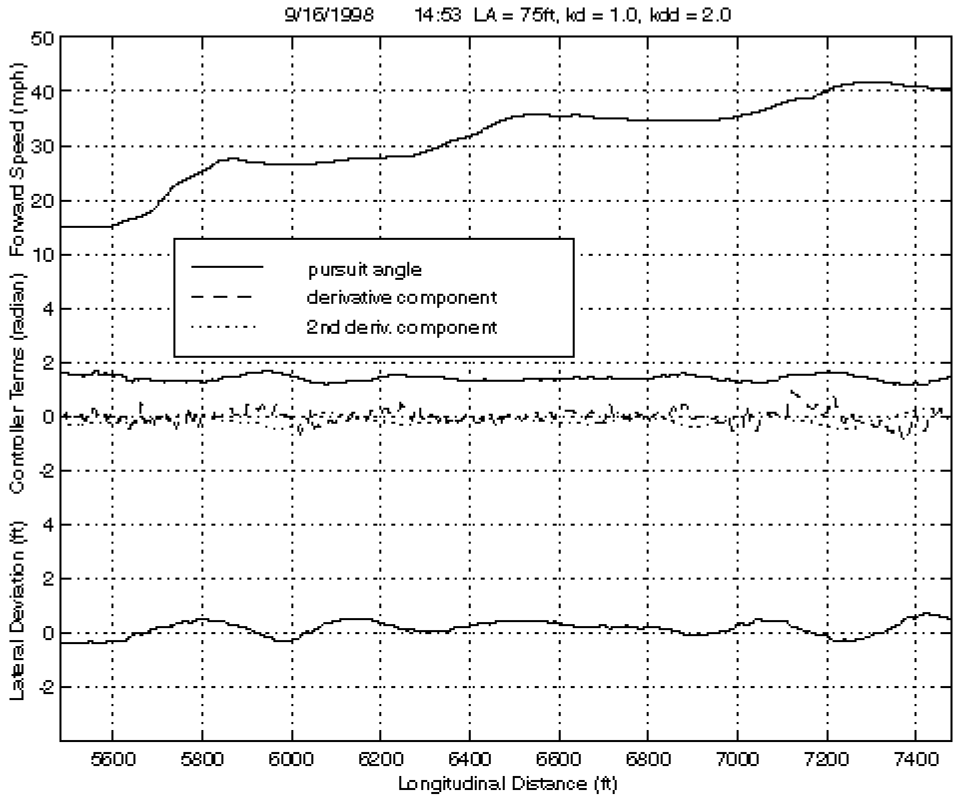

Figure 3.8: Lane keeping performance of the pursuit algorithm on a straight road with derivative and 2nd derivative components included.

3.6 Offtracking and Offsetting Corrections

One complication that results from using a controller that aims at a point on the desired path at some distance in front of the truck is that the actual path the truck follows will be to the inside of the desired path on curves. This is the same problem the driver of a long bus faces while negotiating a tight corner. The rear tires of the bus always point at the front tires and therefore travel in an arc inside of the arc made by the front tires. This phenomena is called offtracking and in this section, we will show geometrically how to correct for it and move the truck back out to the center of its lane.

Another complication we address here is that currently our map database contains points on the centerline of the road. To keep the truck in the center of its lane we need to compensate not only for the offtracking caused by pointing at the pursuit point ahead, but also for the offset to get from the centerline of the road to the middle of our (usually) 12 ft wide lane. The geometry of this situation is illustrated in Figure 3.9. In order to follow the desired path the front wheels, represented in the diagram as one wheel at point *a*, must aim along the tangent to the path which is the line from *a* to *c*. The pursuit point is labeled *p* so the angle returned by the pursuit algorithm will be the sum of ζ and λ and we need to subtract λ to get back to the correct steering angle ζ. To calculate λ3#we can use two successive applications of the cosine law, first on triangle *abp* to find the distance *ap*, then on the triangle *acp* to find λ.

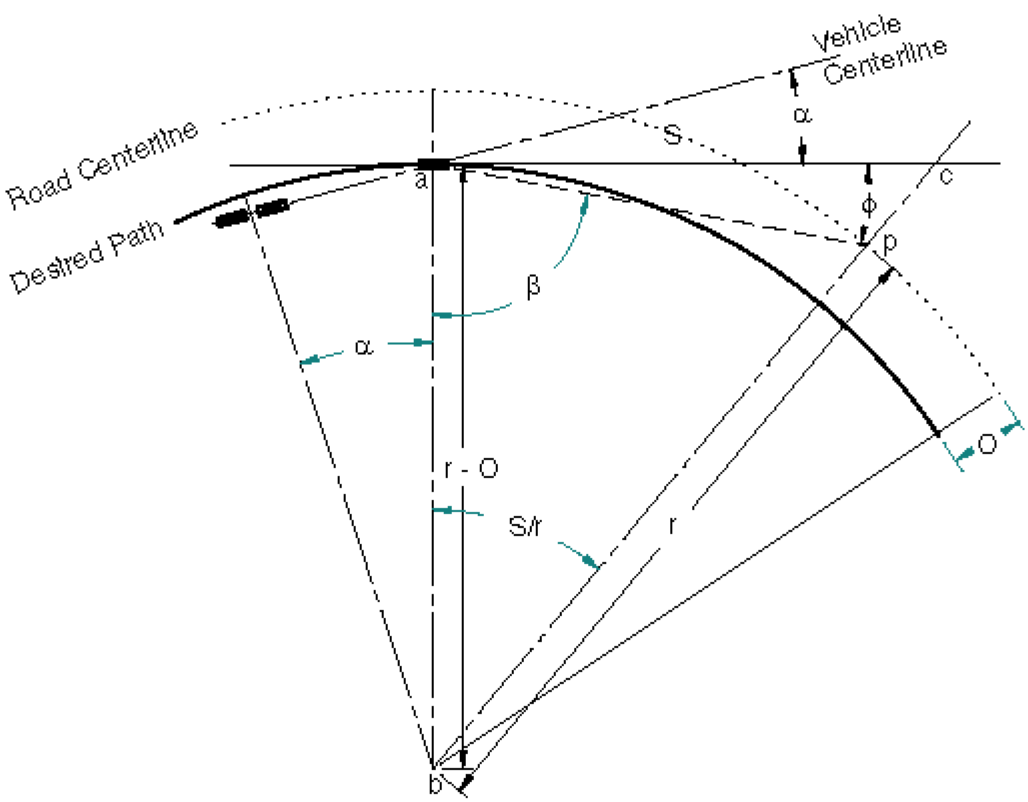

Figure 3.9: Offtracking and lane offset correction. In order to stay in the middle of the desired lane of travel the pursuit angle must be reduced by angle λ. *O* is the offset from the centerline

of the road to the middle of the lane. *S* is the look ahead distance measured along the centerline.

The results of these calculations are that

$$ap = \sqrt{ab^2 + pb^2 - 2\,ab\,pb\cos(S/r)}$$

and that

$$\lambda = \arccos\left(\frac{ap^2 + (r - O)^2 - r^2}{2ap(r - O)}\right)$$

After incorporating the calculation of λ into the hybrid controller and using it to correct for the lane offset and pursuit offtracking we finally have a lateral control algorithm that will steer the truck around the entire test track. On a straight road the results are similar to Figure 3.8 since there is no offtracking to correct for on a straight road. Performance of this algorithm negotiating curves is depicted in Figure 3.10. Note the locations of points *a*, *b* and *c* from the map in Figure 1.1. These points mark the places that the curvature of the track changes abruptly.

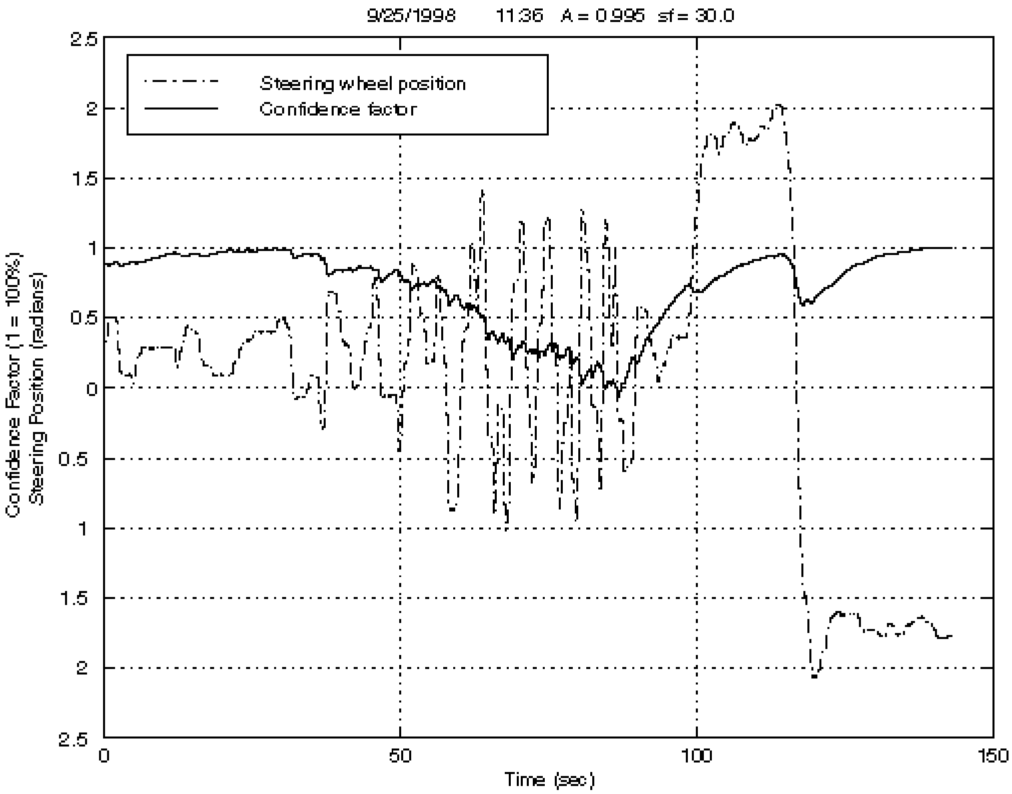

Figure 3.10: This diagram shows the results of using our hybrid controller, including pursuit offtracking corrections, to negotiate the S shaped curve and loop at one end of the MnRoad test track at a speed of 22 mph. The line labeled *left curve* and *right curve* represents the curvature of the track. Points a, b and c are locations of abrupt changes in curvature and are also shown on the map of the MnRoad track in Figure 1.1. Although there are still slight perturbations at the locations where the curvature changes, the truck stays well within the lane at all times.

3.7 Inner Steering Control Loop

Up to this point, we have used several control laws to compute a position which moves the steering wheel. We now consider the steering control subsystem that takes the steering position command as its input and computes a voltage to send to the steering motor that is located on the steering column. We refer to this part of the control system as the *inner loop* because it is a separate control loop running at a faster rate than the *outer loop* that we have described in the previous sections.

This inner loop consists of another PD control algorithm acting on the error between the position command from the outer steering control loop and the actual position sensed by a potentiometer on the steering column. Typical proportional gains are in the area of 1.5 volts per radian and typical derivative gains are about 1.5 volts per radian per second. A slightly lower proportional gain (1.3 volts per radian) will smooth out the path the truck follows down the straightaway, and a slightly higher proportional gain (1.7 volts per radian) will keep the truck closer to the center of the lane on the curved sections.

After the inner loop has calculated a voltage using the PD algorithm it filters out any large jerks that were ultimately caused by noise in the DGPS system. Finally the inner loop limits the absolute value of the output voltage to be less than 1.3 volts. This assures that the human driver will always be able to overpower the automatic system. Figure 3.11 shows a block diagram of the entire lateral control system.

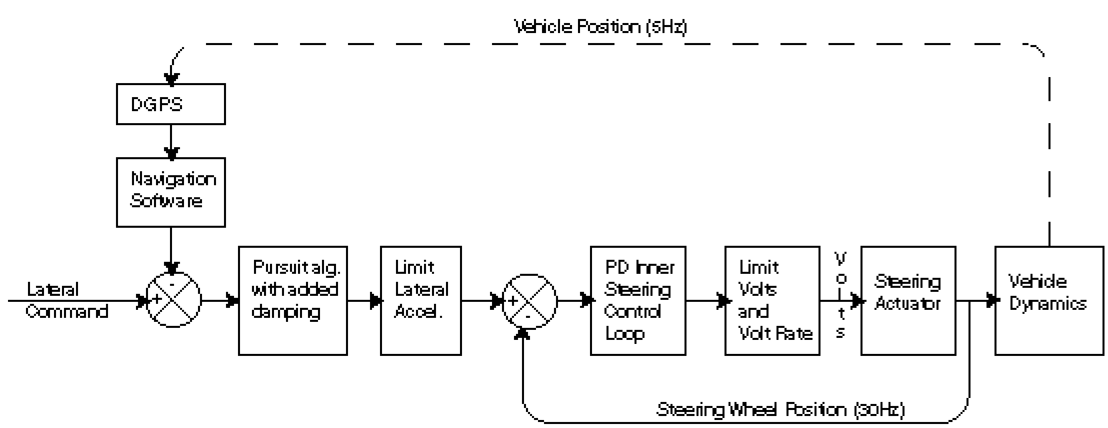

Figure 3.11: Block diagram of the lateral control system.

3.8 Conclusions

We have shown that through the use of fairly simple steering control algorithms, we can use DGPS position signals to keep a vehicle in its lane while traveling at speeds up to 50 mph. The length of the Mn/ROAD test track limited the distance we could travel at higher speeds so more experimentation is in order before we can claim to have a system robust enough for general use at highway speeds. At this point, we see no reason why the control algorithms described above will not work at higher speeds, but it is likely that some gain scheduling will be required to lower the controller response as the speeds increase. It is also possible that the extra stability that a yaw rate gyro can provide at higher speeds will offset the extra cost and complexity that it entails. In the near future, we plan to add an inertial measurement unit to the system that will include both rate gyros and accelerometers along with a Kalman filter. This system should significantly increase the performance of the SAFETRUCK's lateral control system.

CHAPTER 4

Computer Hardware and Software

4.1 Laptop PC and Parallel Port I/O System

The hardware setup developed to support the experiments described in this thesis has been designed to use a generic PC as the control computer. No additional I/O cards are necessary, all communication with external sensors are handled through the standard serial and parallel ports. The DGPS position signal is received directly through the serial port and all other sensing and control signals are handled by a National Instruments DAQ-PAD 1200 I/O device connected to the parallel port.

Currently we are using a Dell Inspiron laptop computer. The model we use has a 266 MHz Intel Pentium processor, 80 MB of ram memory and a 4GB hard disk drive. The performance of this machine is more than adequate for the control tasks involved in the SAFETRUCK research we are doing. We typically find that we use less than 10% of the available processor time when the truck is being automatically controlled and most of that time is spent printing information to the screen. In the future, we intend to add a more sophisticated graphical user interface and we assume that that will use up most of the remaining computing power we have available on this machine. Figure 4.1 shows a basic schematic diagram of the hardware used in this system.

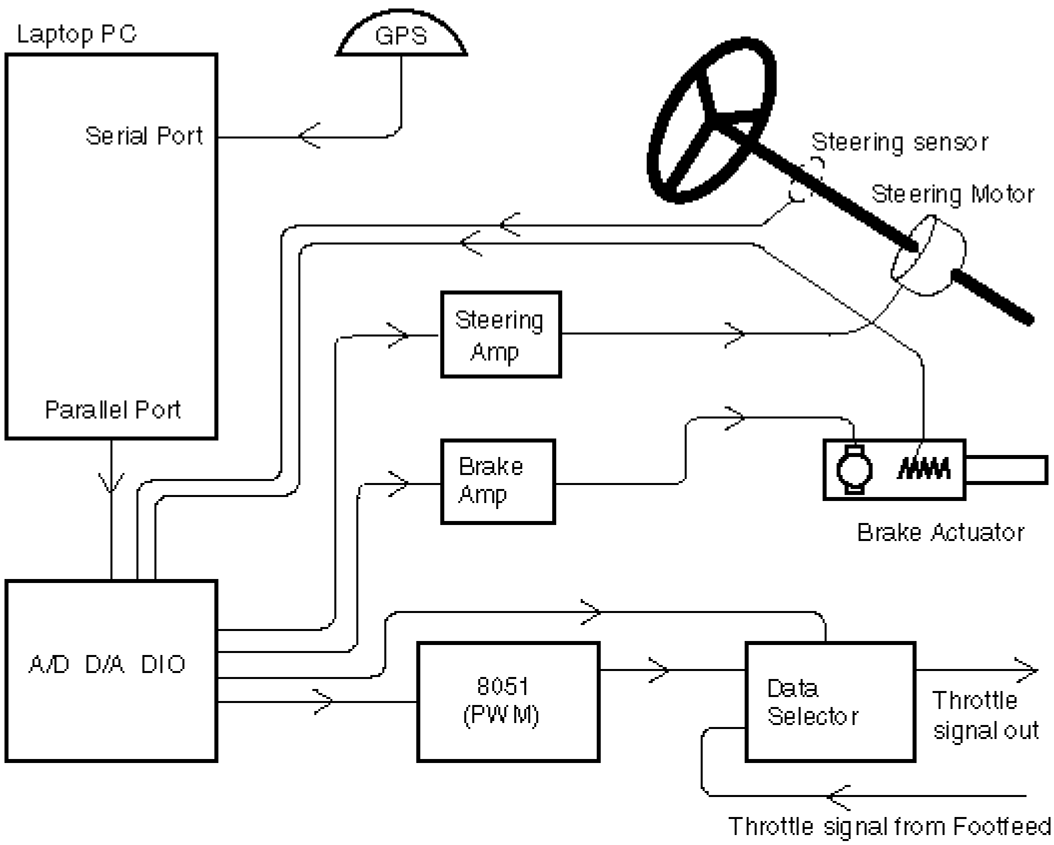

Figure 4.1: Schematic diagram of the PC based control system used in the experiments. A more detailed wiring diagram would also show kill switches for the throttle, brake and steering subsystems.

The National Instruments (Austin, Texas) DAQ-PAD 1200 is an analog and digital I/O unit that communicates to a host computer through a standard bi-directional parallel port. There are 8 analog to digital converter inputs with 12 bit resolution, two 12 bit digital to analog outputs, 24 lines of TTL compatible digital I/O, and three 16 bit counter timers. National Instruments provides a programming library and the necessary drivers to access the DAQ-PAD.

4.2 Real Time Operating System

For the majority of the experiments we have described here we used a real time multitasking operating system named RT Kernel version 4.5 from On Time Informatik GmbH in

Hamburg, Germany. RT Kernel 4.5 is supplied as a set of 16 bit programming libraries for MS-DOS based C compilers from either Borland or Microsoft. The *main()* function of a C program using RT Kernel loads the real time kernel as part of its initialization process. From that time on, the program has access to a fairly complete set of real time multitasking tools including functions to manage tasks, semaphores, mailboxes, message passing, interrupts and a selection of scheduling options. RT Kernel programs compile to a single .exe file that will run on most PC compatible computers.

Included with RT Kernel are several modules containing source code for high resolution timing, serial communication, the Novell IPX networking protocol and other miscellaneous services. The timing module, for instance, has the important ability to reset the DOS system interrupt to a programmer specified value between 0.5Hz and 8000 Hz. We set this interrupt rate to 30 Hz and use it to run our control loops at a constant rate.

Although RT Kernel has the ability to do preemptive multitasking, we use it in a cooperative multitasking mode. This makes it easier to use non-reentrant DOS system calls and other third party programming libraries (such as the National Instruments data acquisition libraries) that have not been designed to be used in a multitasking environment. The slightly higher interrupt latency that we get using cooperative rather than preemptive scheduling is not a significant factor in our implementation.

4.3 The Map Database

The position of the Mn/ROAD test track is currently represented in this system as simple list of points spaced 25 ft apart along the centerline of the 2-lane road. For each point, we include the latitude and longitude to the same eight decimal place accuracy that the Novatel RT-20 system provides, and the curvature of the track at that location. There are a total of 551 of these map points around the 2.6 mile track.

Although the survey data for the Mn/ROAD track is furnished in the Wright County coordinate system, which is based on the number of feet north and east of a certain point in the county, we chose to do all of the mapping and calculating for the work detailed in this paper using latitude and longitude directly. This saves the processing time that would otherwise be needed to convert the degrees of latitude and longitude furnished by the DGPS system into county coordinates, and it also leaves us with a more general navigation system that will work anywhere in the world.

The 25 ft spacing was chosen as a trade off between processing speed and accuracy. When the points are closer together, there are more of them to search through when we are looking for the closest one in our navigation algorithm, but when the points are farther apart, the straight line segments between them do a poorer job of representing curves. The decision to use 25 ft was made during initial testing which was done using a 20 MHz 80386 based computer. With the newer, faster computer we are now using, it is likely that a smaller distance between map points would be closer to the optimum. Other additions to the map that we are currently working on include local speed limits and the position of signs and guardrails along the side of the road.

Eventually we hope to make use of higher accuracy map databases, which we are developing. When they become available, we will need software to access that data real time. Work is in progress to address this issue.

4.4 Software Modules and Control System Tasks

The software for this system is written in C and split into six modules that each contain the tasks and supporting functions for a certain portion of the SAFETRUCK controller. These six modules are:

Drive.c This module contains the *main()* function that initializes the system, spawns the other tasks and monitors the keyboard.

RecvGPS.c Contains the serial port interface that receives data from the GPS system, and the task that parses that data into the latitude, longitude, etc.

Navigate.c Takes the position information parsed above and finds the location and velocity of the truck with respect to the map database.

Steer.c Uses the location and velocity to calculate steering position and the voltage to send to the steering amplifier.

Throttle.c Computes the duty cycle for the PWM throttle control signal based on the difference between commanded and actual velocity.

Brake.c Controls the position of the brake actuator based on the difference between commanded and actual velocity.

Figure 4.2 illustrates the flow of data through the system. In the following sections we will examine each of these modules in more detail.

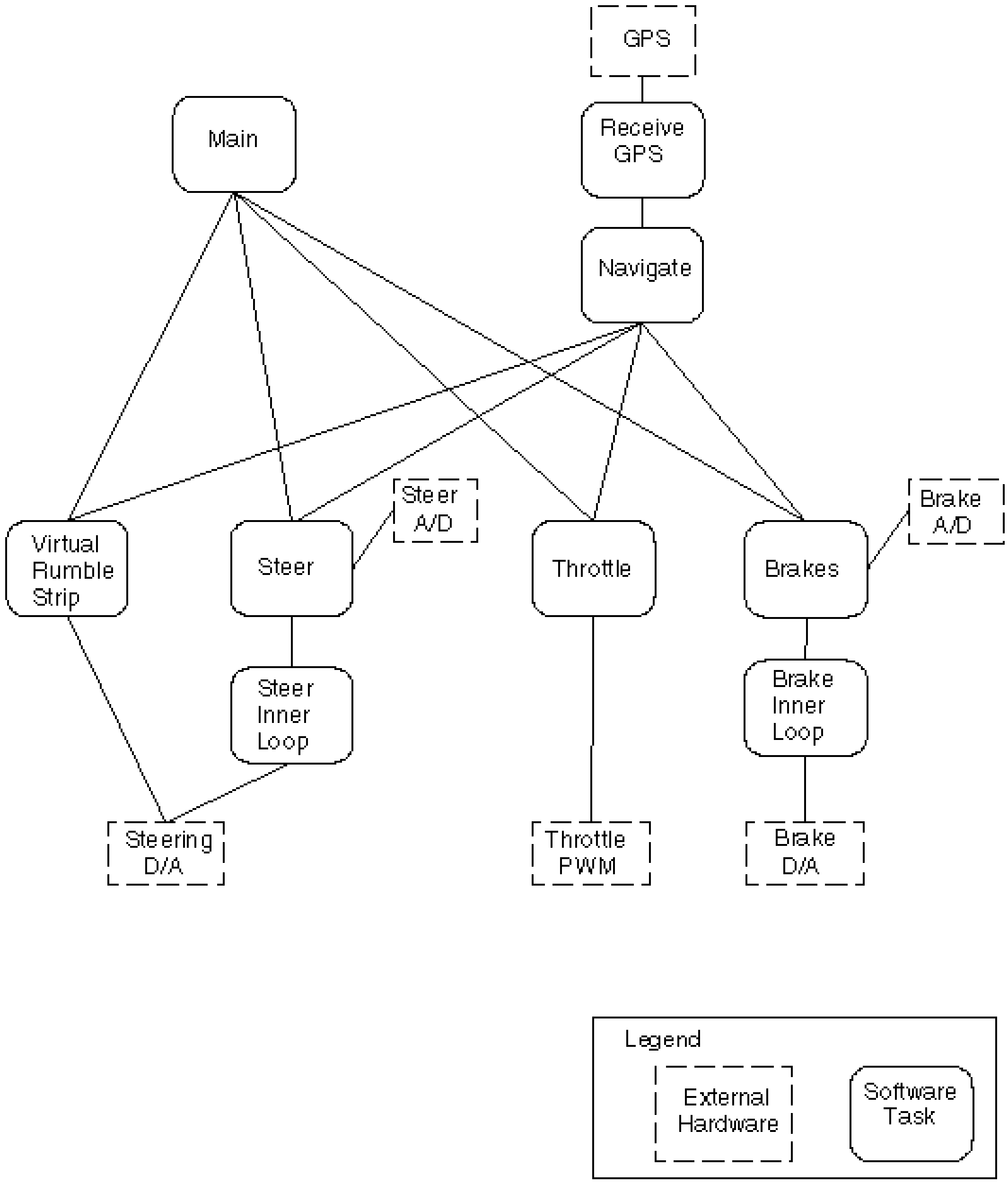

Figure 4.2: Data moves through the various tasks from the top of this

4.4.1 Module Drive.c

The *Drive* module contains the code that initializes all the other tasks, declares global variables, sets up mailboxes for intertask communication, monitors the keyboard for operator commands, and generates velocity and lateral position commands when it is time to execute a pull over and stop maneuver. The *Drive* module contains four functions:

main()

This function is the standard *main()* function common to all C programs. The first action it takes is to initialize RT Kernel by replacing DOS interrupt vectors, initializing some internal data structures, setting RT Kernel's clock to 0, and performing some other low level operations to convert DOS into a real-time operating system. After that the other RT Kernel modules are installed. These modules include an interrupt driven serial port driver, a high resolution timer, a windowing system that lets separate tasks share the screen, and a keyboard driver that releases the processor to other tasks when it is waiting for a key press. Next, the *main* function resets the system interrupt rate to 30Hz from the 18.2Hz that is standard for PC's. 30 Hz has been found to be an acceptable rate for our control loops. Six mailboxes are then initialized to carry messages between various tasks and a semaphore is created to protect calls to the DOS file system when logging data to disk. The ten tasks that perform the various control and data handling operations are started next and a range of priorities are assigned to them. The *main* task then enters a loop during which it monitors the keyboard for commands from the keyboard. These commands include:

F1:	Change operating mode to *Manual*.
F2:	Change operating mode to *Virtual Bumper*.
F3:	Change operating mode to *Automatic*.
l:	Turn on data logging to disk.
o:	Turn off data logging to disk.
Up arrow:	Increase the forward velocity command.

Down arrow: Decrease the forward velocity command.

Right/Left arrow: Adjust the offset of the steering potentiometer.

Space bar: Terminate the program in an orderly manner.

Once the space bar is hit, the *main* function sets a global flag to broadcast to the other tasks that it is time to cleanup and shutdown.

CPULoad()

This is a low priority task that returns an approximate measure of the amount of processor time the entire program is using.

open_logfile()

When triggered by the *l* key this function opens a disk file named log.dat and enters the date and time on the first line. This file can then be used by any other tasks to store arbitrary data for later analysis. Entering an *o* at the keyboard stops data logging and closes the file.

pull_over_and_stop()

This is a function that is called by the *main* task when the system has determined that it is time to pull over and stop. A description of how the system might decide to pull over and stop is given in the next chapter. Once that decision has been made this function monitors the forward velocity and lateral position of the vehicle and generates commands to slowly move the truck to the side of the road and stop.

4.4.2 Module RecvGPS.c

Receiver()

After initializing the serial port this task receives characters from the GPS unit. When it receives the '$' character that marks the start of the RT-20 data string it starts putting characters into an array. When the newline character that marks the end of the data string arrives, this task adds a string terminator (a null character) to the end of the array and sends it via a mailbox to the ParseGPS task.

ParseGPS()

This task receives the GPS data string from the Receiver task, breaks it into its component data values, then fills a data structure with the following information:

- GPS time
- Age of the differential correction
- Latitude
- Longitude
- Number of satellites being received
- RT-20 solution status
- Fix status (single point or differential)

send_GPS_init()

This function is called from *main()* to:

1. set up the RT-20 receiver to accept information from the differential radio over the RT-20's com1 port.

2. Send the P20A log out the RT-20's com2 port to the control computer at a continuous rate of 5Hz.

verify_gps()

ParseGPS() uses this function to make sure that the latitude and longitude it received are within a reasonable distance of the Mn/ROAD facility. If not, there has likely been an error in the serial port transmission and the bad data is not passed on to the rest of the program.

4.4.3 Module Navigate.c

Navigate()

This is the task that uses the latitude and longitude received from the DGPS system to find the position of the truck with respect to the map database. It is sometimes referred to as the *Navigator*. After reading the map from disk into main memory, *Navigate()* waits for the first gps data structure to arrive from *ParseGPS()*. It then calls the function *find_init_index()* which searches the map data to find the map index that is closest to the truck. After that *Navigate()* enters a loop that is synchronized with the arrival of new GPS data. In this loop, the precise location of the truck with respect to the map is calculated along with the rest of the information required by the nav data structure:

- GPS time
- Lateral position with respect to the road
- Longitudinal position along the track
- Curvature of the road in front of the truck
- Pursuit angle
- Speed in mph

The position of the truck is calculated by analyzing the triangle created by the current DGPS reading and the map points ahead of and behind the closest map point (the current map index) to the truck. The curvature is a field in the map database associated with each map index. The program returns a weighted average of the curvature in front of the truck out to a distance equal to the look ahead distance to the pursuit point. This averaging eliminates abrupt changes that would otherwise cause significant oscillations in the steering controller.

The pursuit angle is calculated by subtracting the vehicle's heading from the bearing to the pursuit point. The heading is the angle that a line makes from a previous position to the current position and the bearing is the angle from the current position to the pursuit point on the desired path ahead. Since heading is calculated from the position of two DGPS readings, there is a tendency to introduce noise if the truck is driving slowly and two successive positions are too close together. A small lateral error in one or both positions can then create a large error in heading calculation. To minimize this problem, we keep a bounded buffer of recent latitude and longitude readings and search back through them until we find one that is at least ten feet behind the current location and use that one to compute the heading angle.

The speed in mph is calculated from the distance between successive gps position readings and then put through a low pass filter to remove noise. Care must be taken so that this filter does not create a phase lag long enough to destabilize the longitudinal control algorithms.

4.4.4. Module Steer.c

The steer module contains the tasks that form the inner and the outer steering control loops, a function that computes a feedforward term for the outer control loop, and a task that implements the Virtual Rumble Strip.

Steer()

This task runs in a loop that is synchronized with the reception of new DGPS data from the Navigator. It receives a copy of the nav data structure that contains current position with respect to the track, local curvature, and forward speed. Three control terms and a correction to the pursuit angle from Navigator are calculated here. All terms are in units of radians of steering wheel position. The proportional term is a gain (usually between 0.25 and 0.5 radians per ft) multiplied by the lateral deviation from the commanded position. The derivative term is a gain (usually about 0.8 radians per ft per sec of lateral velocity) multiplied by the difference between two successive lateral positions and then divided by the difference in time between them. The second derivative term is a gain (typically between 1.0 and 2.0) multiplied by the difference between the current steering wheel position and the position calculated by the feedforward function described below. These terms are all combined with the pursuit angle and then limited to a total magnitude that depends on how fast the truck is driving: the faster the forward velocity the smaller the permitted steering angle. The resulting angle is put into a mailbox for receipt by the inner steering control loop.

Steer_inner_loop()

This task receives a steering position command from the Steer() task above, compares it to the actual position of the steering wheel and generates a voltage that controls the steering motor. It runs in a 30Hz feedback loop that reads a voltage from a potentiometer on the steering column to sense steering wheel position, converts that voltage to a position in radians, and then subtracts that position from the commanded position to get the steering position error. The voltage that is to be sent to the steering amplifier is computed from this error by a normal PD control law then the voltage is limited in two ways. First, the rate of change is limited to filter out spikes arising from GPS noise and step changes in commanded position. Second, the absolute magnitude of the voltage is limited to make sure that the steering motor never creates enough torque to overpower the human at the wheel. When the global shutdown flag is set, the control loop terminates and the output voltage is set to zero before the task ends.

ff_steer_pos()

This is a function that returns the steering position that would be required to negotiate a curve of a given radius in the absence of other disturbances. The target curvature (the reciprocal of radius) is passed in as a parameter and an intermediate result is calculated using the basic kinematic formula $\iota = WB/r$ where *WB* is the wheelbase of the vehicle, *r* is the radius of the curve and ι is the angle of the steered wheels on the front axle. ι is then multiplied by a steering ratio of 21.5 to get a steering wheel angle and this number is then modified by some experimentally determined constants that fit the conditions we operate in at the Mn/ROAD test track. The result is then passed back to the calling function.

Note that we have not attempted to wholly compensate for any number of real world complications such as backlash and compliance in the steering system, the slip angle of the tires, or the nonlinearities involved with the nonconstant velocity effect of the U-joints on the steering column. A better analysis of the steering feedforward calculation is probably the most significant thing that we could do to improve the performance of the lateral controller on the SAFETRUCK.

Vrumble()

This task implements the Virtual Rumble strip and is the simplest of all the tasks in the system. It gets the lateral position of the truck from the Navigate() task and, if the truck is not in its lane and the operator has set the mode to *Vrumble*, then a 15Hz square wave is sent to the steering motor.

4.4.5 Module Throttle.c

This task compares the velocity command with the velocity sensed by the GPS receiver, calculates a throttle position, then sends a command to the National Instruments DAQ-PAD 1200. This command results in an eight bit integer that is converted into a PWM signal by the Domino microcontroller in the control box (The Domino is an 8052 based microcontroller manufactured by Micromint, Inc., Vernon, CT). The throttle task is synchronized with the navigate task and therefore runs at the same rate (roughly 5 Hz) as the DGPS system. The control law is a simple PD algorithm. Provision has also been made for an integral term with anti-windup correction, but it has not turned out to be very useful on the Mn/ROAD test track where space restrictions limit the distance that we can attempt to maintain a constant speed. An integral gain large enough to have an effect over the short distances we are dealing with tends to make the longitudinal control system significantly less stable.

After the throttle position is computed using the PD algorithm combined with the feedforward function described in section 2.4.2, it is sent through a filter that limits the rate of throttle increase to 10% per second and limits the maximum throttle output to approximately 50% of full throttle. This filter eliminates uncomfortably quick reactions to abrupt changes in velocity commands and keeps the power output at a manageable level when we are not pulling a loaded trailer (even with the 50% limit the engine will easily overpower the parking brakes on the tractor.)

At startup, the throttle task initializes the National Instruments DAQ-PAD 1200 to use one of its digital I/O ports to send the control byte to the Domino. The Domino has been programmed to send out a 600 Hz square wave on one of its I/O pins while it waits for and reads the control byte on eight other I/O pins. The control byte is converted to an *on* and *off* duration that sets the duty cycle of the square wave. The square wave is then sent from the Domino to a CMOS buffer and finally to the engine controller on the truck.

4.4.6 Module Brake.c

This module contains the two tasks *Brake()* and *Brake_inner_loop()*, that control the air brake system on the Navistar.

Brake()

This is the upper level brake control task that receives vehicle speed from the navigator, compares it with the current velocity command and computes a brake position command that is then sent to the lower brake control loop. This task, like the throttle task, is synchronized with the navigate task and therefore runs at the same rate (roughly 5 Hz) as the DGPS system.

As we mentioned in section 2.5, the brake actuator has too much stiction to be accurately controlled with a normal PD algorithm. The most successful method we have found to overcome this stiction is to send position commands to the inner loop that change in fairly large steps. The derivative term in the inner loop will generate a voltage spike when it receives a step change in the commanded position and that spike will help kick the actuator into motion. We use a small look—up table that approximates a proportional controller to generate the step commands. After determining the brake actuator position that just barely starts to engage the brakes we add to that position an amount that is based on the error between commanded and actual velocity. There are increasing position command entries in the table for errors of from 3 mph to 20 mph. The brake position commands are shown in table 4.1. If the error is less than 3 mph, the brake actuator moves the brake pedal to the released position. This gap of 3 mph at the low end of the error range is necessary (due to lags in the total response time of the system) to keep the longitudinal control system from occasionally applying both throttle and brakes at the same time.

Error (mph)	Brake Pos. Cmd.
20	52.0%
10	46.0%
7.5	44.5%
5.0	41.5%
3.0	40.0%

Table 4.1: Brake position commands used to slow the truck for different velocity errors. This table is used when the velocity command is greater than zero.

If the velocity command is 0 mph and the actual speed is below 8 mph, we may need to have the brakes applied even if the error is less than the 3 mph mentioned above. At these low speeds we have to contend with the large forward force generated by the engine and torque converter in first gear. This force increases the closer the actual velocity of the truck is to 0 mph. We deal with this situation with another look—up table that inverts the response of a normal proportional controller by increasing the commanded position for *smaller* velocity errors when the command is 0mph and the actual speed of the truck is below 8mph. The brake positions used to stop the truck from low speeds are shown in table 4.2.

Velocity (mph)	Brake Pos. Cmd.
8	55%
4	60%
2.5	65%

Table 4.2: Brake position commands used to stop the truck (velocity command is 0). The position has to be increased as the velocity gets closer to 0 mph to overcome the increased torque generated by the engine and transmission.

After passing the velocity error through the two look-up tables described above we then limit the rate of change of the commanded actuator position in an attempt to keep the actuator from releasing too fast. If the actuator releases too fast from the full *on* position it can overshoot the brake-off position, hit the end of its travel and jam in that position.

Brake_inner_loop()

This task receives the brake position command from the brake task described above (the outer loop), compares it with the current position of the actuator, computes a new control voltage and sends the proper command out the parallel port to the National Instruments DAQ-PAD 1200. It runs in a loop at a constant 30 Hz rate. After checking the inner brake mailbox for a new position command (it uses the last position sent if there is no new command), this task verifies that the position command is between the maximum and the minimum allowed actuator position. If the command is for some reason beyond one of these limits, then the command is reset to be at the corresponding limit. The control voltage is calculated using a normal PD algorithm and the result is sent to the DAQ-PAD 1200 which sends the voltage to the brake servo amplifier.

When the *main()* task broadcasts a shutdown command to the system, the outer brake control loop will send a command to the inner loop to position the brake actuator in the off position. The inner loop will keep running for several iterations after the shutdown command is received to give the actuator time to move to the off position. The brake control voltage is then set to 0 volts before the inner loop task terminates.

CHAPTER 5

The Virtual Rumble Strip and Active Intervention

5.1 Virtual Rumble Strips

In section 4.4.4, we described a software task that implements the Virtual Rumble Strip. The idea here is not to steer the vehicle automatically, but to use the steering motor to shake the steering wheel when the vehicle is drifting out of its proper position in the lane in the same manner that a real rumble strip embedded in the shoulder of the road does. One advantage to using a virtual implementation is that it is also possible to warn the driver when the vehicle is crossing the road centerline, a location that a real rumble strip cannot be placed in. In figure 5.1, we see the result of an experiment where the truck is intentionally driven out of the lane in both directions. When the lateral deviation from the center of the lane exceeds two feet, the Virtual Bumper task sends a 15Hz square wave to the steering motor causing the steering wheel to oscillate. This serves notice to the driver that the vehicle is not following the road.

In the current version of the software, there is no attempt to automatically steer the truck back into the center of the lane, but implementing that feature would be straightforward. Another useful future enhancement would be to project a point out in front of the truck in the direction of travel and to use the position of that point to trigger the Virtual Rumble Strip. This would give the driver a lane departure warning slightly in advance of actually leaving the lane (equivalent to a warning based on Time—to—Lane—Crossing (TLC), a term often discussed with respect to driver warning and monitoring systems.)

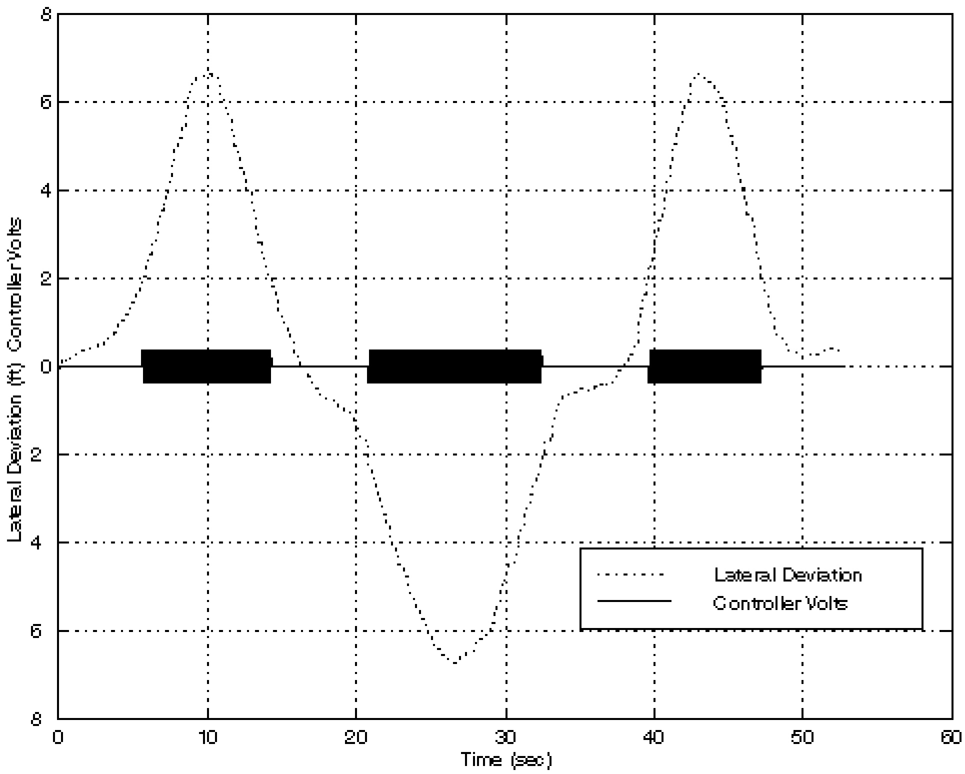

Figure 5.1: Virtual Rumble Strip responding to a lane departure to either side of the lane.

5.2 Virtual Rumble Strips

One of the initial objectives of the SAFETRUCK program was to monitor the performance of a driver in an unobtrusive manner and, if it becomes apparent that he or she is not in control of the truck then the control system would take over and pull the truck over to the side of the road and stop at the next safe parking place. Deciding whether or not a driver is alert and competently in control of the vehicle is a difficult problem that will require a significant amount of human factors research before a reasonably accurate and reliable method is developed. One possible method, and one that we will pursue here, is the result of recent work done in this field by MacInnis Engineering Associates Ltd. in British Columbia, Canada [Siegmund et al. 96]. In a controlled environment on a test track they recorded the pattern of steering wheel movement for seventeen sleep deprived drivers, some of whom actually fell asleep at the wheel. They found a significant increase in both the amplitude and speed of steering wheel movement in the minutes before the driver loses consciousness. We will describe here a qualitative demonstration of this method that will trigger the control system to pull over and stop when the steering input becomes too uneven.

In order to accomplish this task we need to sense the slowly increasing tendency of the driver to make slightly larger and faster steering corrections while at the same time ignoring the occasional quick steering inputs that are a part of normal driving. To do this we will send the absolute value of the steering wheel position, sensed at a rate of 30 Hz, through a low pass filter with a low enough cut-off frequency so that fast but infrequent steering motions are more or less ignored. A phase plot of steering position data for a test run where the truck was steered normally for the first part of the run, and erratically for the second part is shown in Figure 5.2. The larger "orbit" of the erratic steering is what we are trying to sense. Normal straight line driving is shown in the plot as the denser area in the center. The smaller lobes on either side of the large central area are the result of steering normally through a curves to the left and right.

In order compute a number that represents the confidence that the system has in the driver, we first determine experimentally what the output of the filter is during normal driving. We then subtract that value (that represents normal steering wheel movements) from the output of the filter and then subtract that result from a number representing 100% confidence in the driver's alertness. The higher the output from the filter the lower the confidence our system has in the driver. At some low level of confidence, the system takes over and parks the vehicle. The top graph in Figure 5.3 shows the confidence going down as the steering input becomes erratic and Figure 5.4 shows the response of the truck after it loses enough confidence in the driver to pull over and stop.

For the purposes of this demonstration we have chosen some rather arbitrary values for the filter coefficient and the confidence value that triggers the pull over and stop maneuver. Much more research needs to be done on the human factors involved before we can design a system for general use in the real world. In other Canadian research on drowsy driving [Wylie et al. 96] it was found that the steering input required for routine lane keeping varied considerably with traffic and road conditions. We should also note that we are not advocating that the vehicle ever be allowed to overrule the control inputs of the driver. Even if the system we described has lost confidence in the driver, he or she ought to always be able to override the system and drive normally.

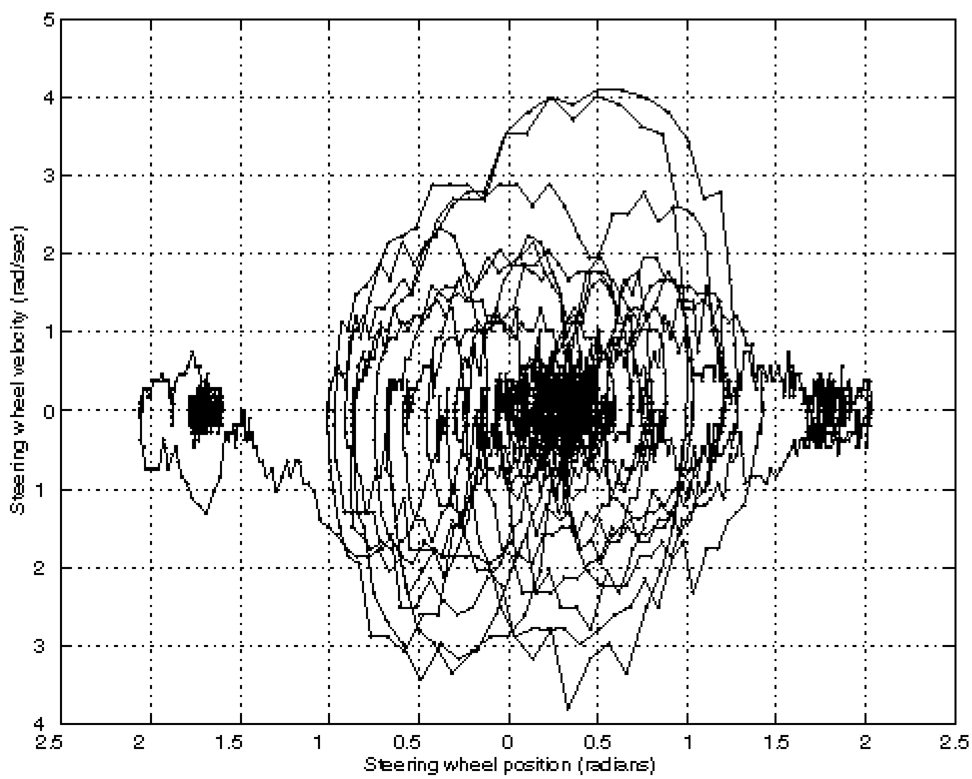

Figure 5.2: Phase plot of steering wheel velocity vs. position for normal driving and for the less smooth response that we will use to simulate a driver falling asleep. Lobes at either side are from cornering in opposite directions.

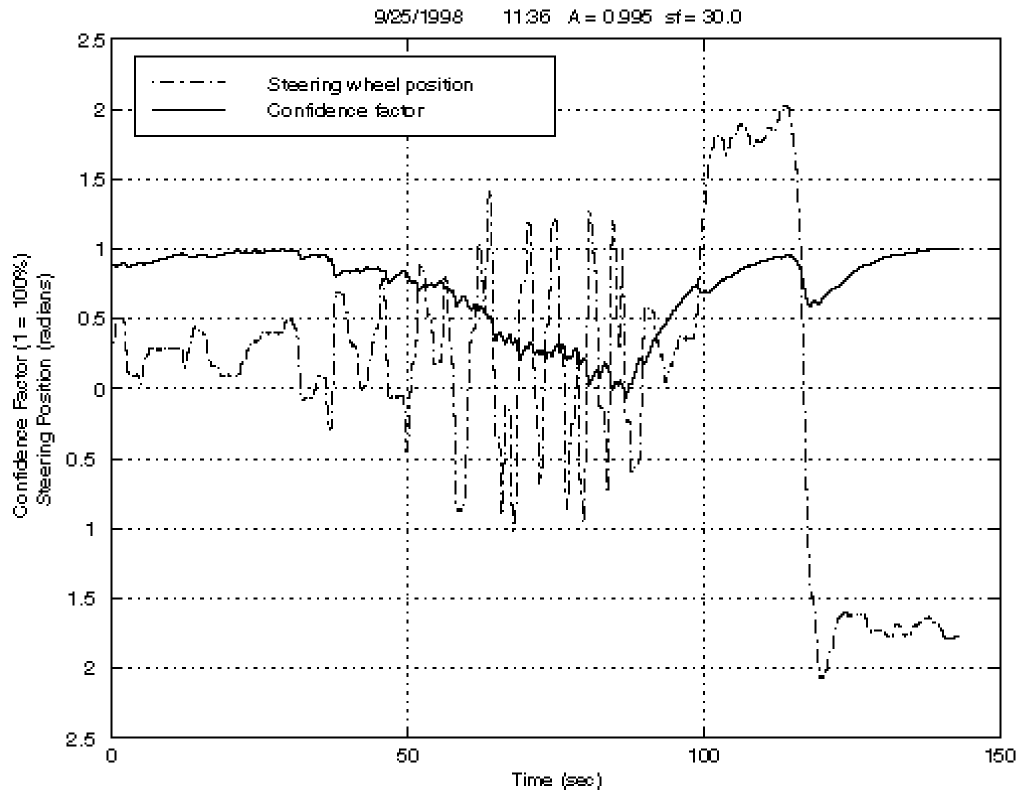

Figure 5.3: The confidence filter response as the truck moves down the straightaway and around the curves in the loop. The truck is being steered normally from 0 to 40 seconds at which time the steering becomes erratic. At 90 seconds the steering action returns to normal. At 100 seconds the truck enters a curve to the right and at about 120 seconds makes a quick transition to a curve to the left. The confidence factor drops during the erratic steering, but stays high during while the truck is being steered normally.

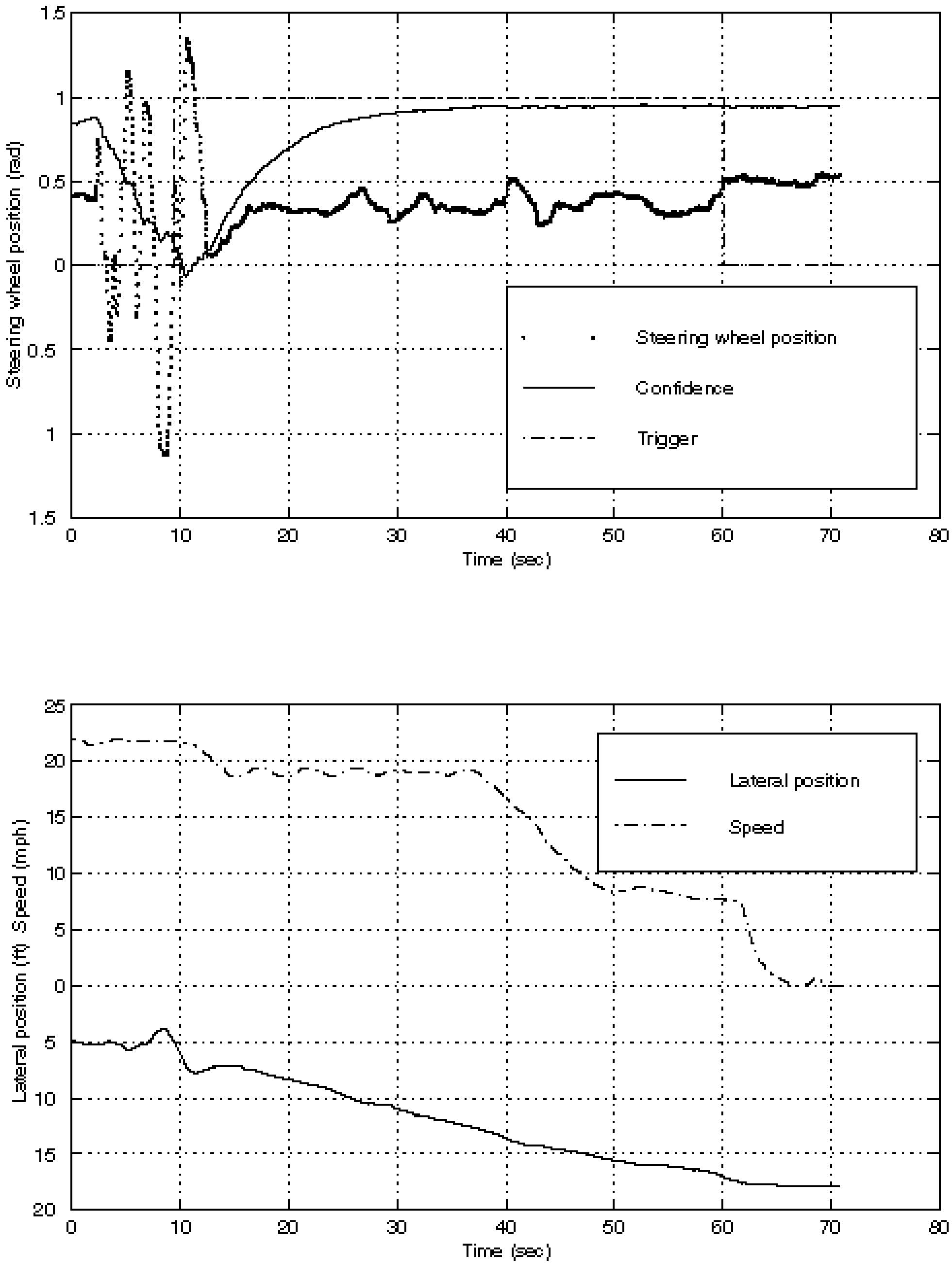

Figure 5.4: SAFETRUCK responding to a loss of confidence in the performance of the driver. After the confidence factor drops to 0.1 in the top graph, the pull-over-and-stop maneuver is triggered and the truck slows down, pulls over to the shoulder and stops as shown in the bottom graph.

5.3 Conclusions

We have demonstrated that it is possible to control a vehicle laterally and longitudinally using DGPS as the only sensor of vehicle position. Two preliminary applications that used this technology were developed to demonstrate the utility of our research in helping to prevent run off the road accidents. At this point, we see no reason why the controllers we have implemented will not work at highway speeds, but all the experiments we have done to date have been on a test track at lower speeds. Work is in progress to deal with the problem of loss of satellite lock when passing under bridges. This will require an inertial measurement unit using a rate gyro and accelerometers to guide the truck until DGPS lock is reacquired. The SAFETRUCK therefore, is still a work in progress.

Bibliography

[Ackermann and Sienel 90] Ackermann, J., and Wolfgang Sienel, "Robust Control for Automatic Steering," Proceedings of the American Control Conference, San Diego, 1990, pp.795-800.

[Ackermann et al. 95] Ackermann, J., J. Guldner, W. Sienel, R. Stienhauser, and V. Utkin, "Linear and Nonlinear Controller Design for Robust Automatic Steering," *IEEE Transactions on Control System Technology*, vol. 3, no. 1, 1995.

[Alexander et al. 97] Alexander, Lee, Max Donath, M. Hennessey, Vassilios Morellas, Craig Shankwitz, "A Lateral Dynamic Model of a Tractor-Trailer: Experimental Validation," Minnesota Department of Transportation, Report number 97-18.

[Bajakar et al. 97] Bajikar, Sundeep, Alec Gorjestani, Pat Simpkins, and Max Donath, "Evaluation of In-Vehicle GPS-Based Lane Position Sensing for Preventing Road Departure", IEEE Conference on Intelligent Transportation Systems, Boston, 1997.

[Bodor et al. 97] Bodor, Robert, Lee Alexander, ChenFu Liao, Sundeep Bajikar, Vassilios Morellas, Max Donath, "Analysis of a Differential Global Positioning System as a Sensor for Vehicle Guidance," Minnesota Department of Transportation, Report number 97-17.

[Du et al. 95] Du, Yufeng., William Schiller, Don Krantz, Craig Shankwitz, Max Donath, "ALX: Autonomous Vehicle Guidance for Roadway Following and Obstacle Avoidance," 1995 IEEE International Conference on Systems, Man and Cybernetics.

[Fancher et al. 93] Fancher, P., Z. Bareket, G. Johnson, "Predictive Analyses of the Performance of a Headway Control System for Heavy Commercial Vehicles," The Dynamics of Vehicle on Roads and Tracks. Proceedings of the 13th IAVSD Symposium, August 1993.

[Fenton et al. 76] Fenton, R., G. Melocik, K. Olson, "On the Steering of Automated Vehicles: Theory and Experiment," *IEEE Transactions on Automatic Control*, vol. AC-21 no. 3, June 1976.

[Flory 67] Flory, Leslie E., "A System for Control of Highway Vehicles," SRI Urban Transportation Alliance Symposium, May, 1967.

[Garrard and Kornhauser 73] Garrard, W. L., and A. L. Kornhauser, "Use of State Observers in the Optimal Feedback Control of Automated Transit Vehicles," *Journal of Dynamic Systems Measurement and Control*, 95, 2, June 1973.

[Guldner et al. 94] Guldner, J, Vadim I. Utkin, and J. Ackermann, "A Sliding Mode Control Approach to Automatic Car Steering", Proceedings of the American Control Conference, Baltimore, Maryland, 1994, pp.1969-1973.

[Guldner et al. 96] Guldner, Jurgen, Han-Shue Tan, and Satyajit Patwardhan, "Analysis of Automatic Steering Control for Highway Vehicles with Look-down Lateral Reference Systems," *Vehicle System Dynamics*, vol. 26, no 4, December, 1996.

[Hedrick et al. 91] Hedrick, J. K., D McMahon, V. Narendran, and D. Swaroop, " Longitudinal Vehicle Controller Design for IVHS Systems," Proceedings of the American Control Conference, vol. 3, pp. 3107-3112, June, 1991.

[Morellas et al. 97] Morellas, Vassilios, Ted Morris, Lee Alexander, and Max Donath, "Preview-Based Control of a Tractor Trailer Using DGPS for Preventing Road Departure Accidents," IEEE Conference on Intelligent Transportation Systems, Boston, 1997

[Nguyen et al. 97] Nguyen, Thanh M., James W. Sinko, Randal C.Galijan, "Using Differential Carrier Phase GPS to Control Automated Vehicles," 40th Midwest Symposium on Circuits and Systems, August 1997.

[Novatel 95] "GPS Card Command Descriptions Manual," Novtel Communications Ltd, 1995.

[Patwardhan et al. 97] Patwardhan, Satyajit, Han-Shue Tan, Jurgen Guldner, "A General Framework for Automatic Steering Control: System Analysis," Proceedings of the American Control Conference, June, 1997.

[Peng and Tomizuka 90] Peng, H., and M. Tomizuka, "Vehicle Lateral Control for Highway Automation," Proceedings of the American Control Conference, San Diego, pp. 788-794, May 1990.

[Peng and Tomizuka 93] Peng, Huei, and Masayoshi Tomizuka, "Preview Control for Vehicle Lateral Guidance in Highway Automation," *Journal of Dynamic systems, Measurement and Control*, vol. 115, no. 4, December 1993.

[Pham et al. 94] Pham, H., K. Hedrick, and M. Tomizuka, "Combined Lateral and Longitudinal Control of Vehicles," Proceedings of the American Control Conference, pp. 1205-1208, Baltimore, Maryland, June, 1994

[Rajamani et al. 97] Rajamani R., J.K. Hedrick, and A. Howell, "A Complete Fault Diagnostic System for Longitudinal Control of Automated Vehicles," Symposium on Advanced Automotive Technologies, 1997 ASME International Congress.

[Rajamani et al. 98] Rajamani R., S. B. Choi, J.K. Hedrick, B.Law, R. Prohaska, and P. Kretz, "Design and Experimental Implementation of Control for a Platoon of Automated Vehicles", Proceedings of the ASME—Dynamic Systems and Control Division, pp. 681-689, 1998.

[Riekert and Schunck 40] Riekert, P., and T. E. Schunck, "Zur Fahrmechanik des gummibereiften Kraftfahrzeugs (in German)," Ingenieur Archiv vol. 11, pp. 210-224, 1940.

[Roos et al. 97] Roos, G., R. Rollet, and R. F. C. Kriens, "Numerical Simulation of Vehicle Behaviour During Straight Line Keeping on Undulating Road Surfaces," *Vehicle System Dynamics*, 27, no. 4, April, 1997.

[Schiller 97] Schiller, W., "The Virtual Bumper—A Control Based Collision Avoidance System for Highway Vehicles," Master's Thesis, University of Minnesota, 1997.

[Shin et al. 91] Shin, D. H., S. Singh, W. Shi, "Partitioned Control Scheme for Mobile Robot Path Tracking," IEEE 1991.

[Shladover 95] Shladover Steven E., "Review of the State of Development of Advanced Vehicle Control," *Vehicle System Dynamics*, vol 24, no. 6-7, pp. 551-595, 1995.

[Siegmund et al. 96] Siegmund G. P., D. J. King, D. K. Mumford, "Correlation of Steering Behavior with Heavy—Truck Driver Fatigue," SAE Paper no. 961683.

[UMTRI 1997] University of Michigan Transportation Research Institute , Ann Arbor, Michigan. "The Mechanics of Heavy Duty Truck Systems (Course notes)," pp. 6-28, 1997

[Wylie et al. 96] Wylie, C. D., T. Shultz, J. C. Miller, M. M. Mitler, R. R. Mackie, "Commercial Motor Vehicle Driver Fatigue and Alertness Study: Technical Summary," Transport Canada, No. TP 12876E.

[Xu and Ioannou 94] Xu, Z. and P. Ioannou, "Adaptive Throttle Control for Speed Tracking," *Vehicle System Dynamics*, vol. 23, no. 4, pp. 293-306, 1994.

[Yanakiev and Kanellakopoulos 96] Yanakiev, D. and I. Kanellakopoulos, "Speed Tracking and Vehicle Follower Control Design for Heavy Duty Vehicles," *Vehicle System Dynamics*, vol. 25, no. 4, pp. 251-276 1996.

[Yanakiev and Kanellakopoulos 97a] Yanakiev, D. and I. Kanellakopoulos, "Longitudinal Control of Heavy Vehicles with Air Brake Actuation Delays," Proceedings o the American Control Conference, Albuquerque, New Mexico, Ju 1997.

[Yanakiev and Kanellakopoulos 97b] Yanakiev, D. and I. Kanellakopoulos, "Longitudinal Control of Automated CHVs with Significant Actuator Delays," Proceedings of the 36th Conference on Decision and Control, San Diego, CA, December 1997.

[Zworykin and Flory 58] Zworykin, V. K., and L. E. Flory, "Electronic Control of Motor Vehicles on the Highways," Highway Research Board Proceedings, 37th Annual Meeting, 1958.