Design and Evaluation of Cooperative Adaptive Cruise Control System for Heavy Freight Vehicles

by

William Grant Apperson

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 14, 2019

Keywords: Vehicle Dynamics,Vehicle Control, Heavy Vehicles, Fuel Savings, RADAR, CAN

Approved by

David M. Bevly, Chair, Professor of Mechanical Engineering
John Y. Hung, Professor of Electrical and Computer Engineering
Peter D. Jones, Professor of Mechanical Engineering

Abstract

This thesis describes the design, implementation, and evaluation of a Cooperative Adaptive Cruise Control (CACC) system for heavy freight trucks that seeks to provide a platform for future cooperative control and estimation schemes. The freight trucking industry is the primary method of transporting goods in the United States. Approximately 70 percent of all transported goods travel by freight trucking. It is estimated that the freight trucking industry in the US spends close to 10 billion dollars on fuel each year. Most of the industrys time and fuel is consumed on long interstate corridors. Due to the high volume of these vehicles, there is huge potential benefit for a collaborative scheme of control. This control algorithm can help vehicles reduce fuel consumption, emissions, driver fatigue, and traffic congestion. By reducing the inter-vehicle spacing and automating throttle and brakes, all of these goals can be accomplished. This control system is known as CACC and is particularity valuable for freight vehicles as they can see significant value from reduced air drag by staying within the wake of a preceding truck.

CACC systems have been in development for many years now and are beginning to be tested on real-world convoys. There is still some potential savings to be gained through controller optimization. To validate new control and estimation techniques a vehicle testing platform is required. The primary contribution of this work is in the development of a platform for future research and validation as a proof of concept. The system is comprised of a communication network between vehicles, a low-level brake and throttle controller, a range estimation scheme and a cascaded gap controller. Testing results from this system both in simulation and on-highway driving are presented and show fuel savings of approximately 2-3 percent. While the fuel savings achieved under this work are not as high as predicted, it is expected that with further controller optimization will yield results in line with that of

other researchers. The vehicle platform developed, however, was shown to be stable and will provide a good basis for future research.

Acknowledgments

I would like to thank Dr. Bevly for the opportunities he as given me to study and work alongside him. I would never have considered graduate education without his influence. I would like to thank Dan Pierce for being a great roommate and for always being a great sounding board for ideas. I specifically want to thank him for his help with the estimation scheme implemented in this work. The controller designs put forth in this thesis would not have been possible without the accurate measurement he was able to help me obtain. I would like to thank TARDEC (particularly Scott Heim) and the Federal Highway Administration for their continued funding and interest in this space. I am thankful for Patrick Smith who worked hard when helping test the system. I look forward to seeing what he will do with the system in the future. I also need to thank James Johnson and all the people at NCAT for all the time devoted to testing and evaluation this system. I should probably also thank my wife Courtney for all of her support. Without her nagging this thesis would not exist. Finally, I would like to thank Celine Dion for all the music that she has put out over the years and for inspiring me to reach new heights.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Terminology

ACC	Adaptive Cruise Control
CACC	Cooperative Adaptive Cruise Control
CAN	Controller Area Network
CFD	Computational Fluid Dynamics
Convoy	Group of vehicles using CACC
DSRC	Dedicated Sort Range Communication (Inter-Vehicle Communication Protocol)
DRTK	Dynamic Base Real Time Kinematic (Differential GPS algorithm)
GPS	Global Positioning System
Headway	Spacing between vehicles (measured by linear distance or time)
J1939	SAE CAN Protocol for Heavy Vehicles
J2735	SAE DSRC Protocol for Inter-Vehicle Communication
NCAT	National Center for Asphalt Technology (Opelika, Alabama)
PVT	Position, Velocity, Time
V2V	Vehicle to Vehicle Communication
V2I	Vehicle to Infrastructure Communication

Chapter 1

Introduction and Background

In recent years road throughput limitations, increasing fuel cost and emissions have gained much attention. At the same time, automotive manufacturers are investing heavily in research and development of vehicle automation systems, primarily to increase safety and reduce driver fatigue. This increase has greatly advanced the development of many low-cost sensors specifically in the passenger vehicle space. These advances are also having an impact on the heavy freight industry. While there are great advantages in terms of safety and driver fatigue, the heavy freight traffic on roadways can also see great benefits in reduction in emissions and fuel consumption Figure 1.1 shows the percent drag reduction caused by drafting from CFD analysis presented in [1]. These results were compared to SAE fuel testing results of a CACC system. The CFD analysis shows the drafting effects reach a steady-state drag reduction at 100ft. This is due to the type of CFD model used which does not completely capture the effects of a dissipating wake. The total fuel savings were consistent with these results though the following truck showed a slightly different trend. These fuel savings could have a significant impact on the freight industry. Within the transportation industry, trucking represents approximately 70% of all value transported, making this the primary mode of transportation of goods within the United States [2] . By decreasing the total fuel consumption of this form of transportation significant reduction in carbon emissions can also be achieved.

The large frontal area and the long chassis of a heavy freight vehicle with a van trailer leads to a well-defined airflow profile which produces a large wake. Other vehicles can take advantage of this wake to reduce their total drag if the inter-vehicle spacing is small enough. However, this diminishing gap between vehicles has raised two main concerns.

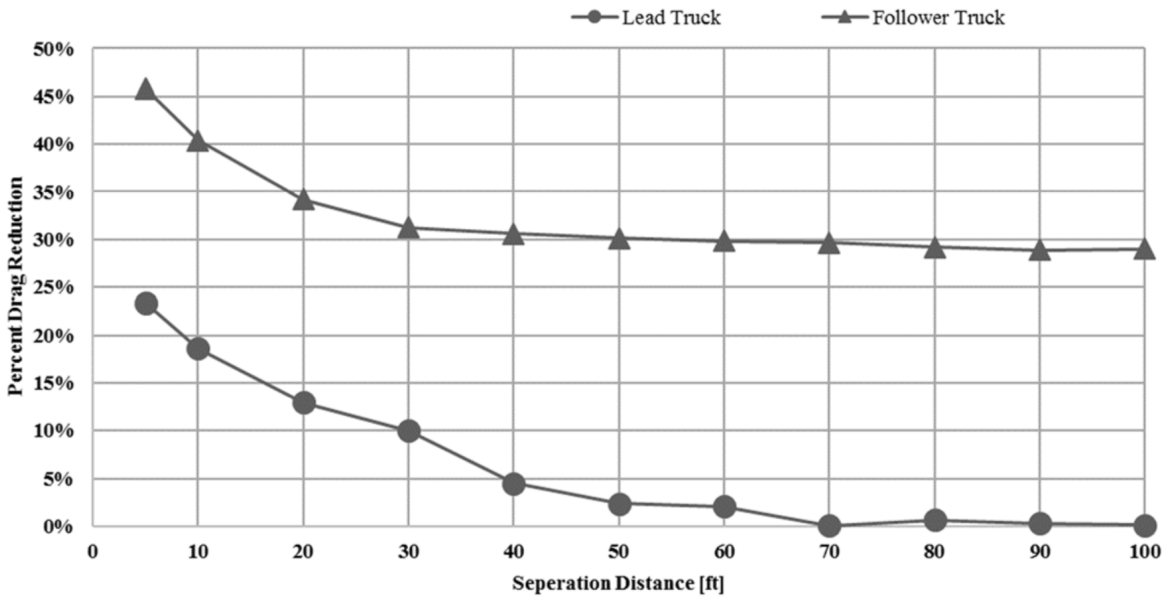


Figure 1.1: Predicted Drag Reduction Relative to Inter-Vehicle Spacing

Firstly, drivers experience a diminished field of view beyond their vehicle, and secondly the reduced gap naturally decreases the available time for a driver to react to changes in a preceding vehicle's velocity. Thus, unaided drivers are unable to safely operate vehicles at these close spacings. In fact, many states in the US have "tailgating" laws that seek to penalize drivers for following at unsafe distances. For heavy freight vehicles, this is normally between 200 and 300 feet. At this spacing the drafting effects are significantly lower.

The inability of drivers to perform these tasks safely and their potential benefits makes this an area ripe for automation. In fact, there has been a lot of interest in automation of the inter-vehicle spacing both in passenger and heavy freight vehicles. These systems are usually called adaptive cruise control (ACC) systems and they seek to augment traditional cruise control systems by modulating the set speed based on a preceding vehicle when present. The following vehicle's speed is decreased to that of its leader and the control system seeks to maintain a safe headway in time. This headway can be described as the inter-vehicle distance divided by the vehicle velocity. For passenger vehicles this is normally designed to be around one second [3]. At highway speeds this leads to a gap of 30 meters at 30 meters per second (98 feet at 65 Mph) between passenger vehicles. Due to the increased mass and

the lower bandwidth of brake actuation, the design specification of ACC systems in heavy freight vehicles is set around 2.5 seconds. This gap in time leads to an inter-vehicle spacing of 75 meters (approximately 250 feet at 65 Mph). Again, at these spacings the benefits of drafting are greatly diminished.

Cooperative Adaptive Cruise Control (CACC) systems are developed to help mitigate the effects in lower-level control actuation lag by adding a communication layer to the architecture. These systems are capable of reducing the gap to 10 meters (approximately 30 feet). At these gaps the benefits of drafting are considerable [1]. This thesis seeks to develop a CACC system for freight vehicles. All of the required components for such a system were developed under this work and will be described in detail in this thesis. The system designed here provides a foundation for future research in the automated cooperative convoy space.

1.1 Prior Research

The field of automated longitudinal control for both passenger and freight vehicles has been an area of significant research over the past 30 years. The potential benefits for driver comfort and safety have led to widespread adoption of these automated systems on passenger vehicles. The trucking industry has started to see significant growth in this area as well due to the same reasons mentioned for passenger vehicles with the addition of drastic fuel savings due to drafting. The references that follow serve as a foundation and inspiration for much of the work presented in this thesis.

1.1.1 Adaptive Cruise Control Systems

Adaptive cruise control systems were a heavy area of research in the 1990s. Many systems were developed the help relieve driver fatigue and improve safety [4]. One of the key principles that came from these systems is the importance of string stability [5]. This string stability effectively limited the gap between vehicles due to lower-level vehicle control lag. Control schemes in [6] ensure string stability by attempting to produce a minimum impact

on surrounding vehicles through aggressive lower-level control. These ACC control schemes are becoming more prevalent in passenger vehicles today and are readily available.

1.1.2 Cooperative Adaptive Cruise Control Systems

Following the development of Adaptive cruise control systems Automated Highway systems were proposed [7]. These systems used special lanes and a communication network between vehicles for small gap following. As the design of these systems progressed, they began to be tested on systems not limited by specific lanes. These systems became known as Cooperative Adaptive Cruise Control (CACC) Systems and were meant to further increase string stability and reduce the inter-vehicle spacing [8]. Ploeg at PATH demonstrated a convoy with six passenger vehicles [9] showing both simulation analysis and validation through real world testing.

1.1.3 Inter-Vehicle Spacing Estimation Schemes

Convoy control algorithms rely on accurate range measurements between the controlled vehicles. A fusion of differential GPS and INS systems for this purpose is shown in [10]. Building on the research and combining it with RADAR measurements in a cascaded filter is presented in [11]. Essentially this algorithm seeks to improve raw RADAR tracking data with measurements from differential GPS.

1.1.4 Longitudinal Control Schemes of Heavy Vehicles

As automation has become more common on passenger vehicles the heavy vehicle industry followed with specific control algorithms designed for their use [12]. The unique challenges of automated longitudinal control for freight vehicles are outlined in [13] detailing the effects of actuation latency and control saturation. Hedrick et. al. further developed ACC system for heavy vehicles [14]. In recent years these control algorithms have become increasingly complex as MPC control techniques were developed for heavy vehicles as well [15]. These

systems are rarely validated with real world testing due to the expense and complexity of integration and limitation of current road safety laws. Thus a platform for developing and testing these algorithms can help prove the feasibility of optimal control techniques.

1.2 Contributions

The primary contribution of this thesis is the development of a real-time system for validation and testing of convoy control schemes on heavy vehicles. The system developed in this thesis provides a robust estimation scheme and a stable controller. It provides a good foundation to be built on by future researchers. In brief, the primary contributions of this thesis are

- Development of hardware and software integrated vehicle platform for validation
- A stable control scheme for heavy freight vehicles
- An estimation scheme for fusing GPS, RADAR and vehicle state measurements for accurate inter-vehicle spacing measurements.
- Validation of algorithms with multi-vehicle convoys

1.3 Thesis Layout

Chapter 2 provides a detailed derivation of the dynamic model that is used for controller design. System model parameter measurements and estimation are presented. Additionally, the system is validated against the Peterbilt trucks used in this thesis. Utilizing the model validated in this chapter a simulation environment is then developed utilizing MATLAB, SIMULINK and a high-fidelity vehicle simulation software package, TruckSim. The design of this simulation environment and its validation against the model is also presented in Chapter 2. Prior to any discussion on control, it is necessary to discuss the development of an inter-vehicle spacing estimation scheme. Chapter 3 presents an explanation of the

fusion algorithm that combines vehicle state information, GPS PVT measurements, and RADAR tracks into a robust measurement to be used for control. Next a control scheme is presented along with an investigation of string stability. Chapter 4 shows the derivation of the final control scheme and its validation in the simulation environment. Chapter 5 presents a detailed view of the primary contribution of this thesis, namely a real-time hardware and software implementation for validation of the estimation and control schemes developed. In this chapter specific hardware components and their limitations are discussed. Safety features and engineering tools are also presented. Finally, a look at the software architecture and its robustness and higher-level safety functions are shown. Chapter 6 discusses the testing and validation of the control and estimation schemes on the Auburn Universities NCAT test track and US Highway 280. Lastly, Chapter 7 gives an overview the systems overall performance and presents recommendations for improvements and next steps.

Chapter 2

System Modeling

Before discussing the controller architecture or simulation of heavy truck convoys, a dynamic system model must be defined. This chapter describes the derivation of the model use the Newton-Euler method for describing equations of motion. In this chapter the assumptions required for model simplification are described along with parameter estimation techniques and basic model validation against the instrumented Peterbilt 579s described in Chapter 5. It should be noted that some basic algebraic steps were omitted in this chapter for the sake of brevity; however, the complete derivation is presented in the Appendix A1.

2.1 Development of Linear Vehicle Driveline Model

The model used to describe the longitudinal dynamics of the vehicle is described in this section. The derivations here are based on the previous work presented in [13] [16] [17]. In order to simplify the complex vehicle dynamics to its essential elements several assumptions are made to remove the highly nonlinear components. These assumptions are as follows:

- No slip between the tire and road
- The tractor and trailer are assumed to be rigidly linked
- The driveline is assumed to be engaged (i.e., there is no slip between the flywheel and the transmission or the torque converter is locked)
- The engine dynamics are assumed to be a first-order system (these lower-level controllers are set by OEM component suppliers and are not easily altered)

These simplifications make the overall model slightly less accurate; however, they also make the control design much more straight forward. For most highway driving these suppositions are not violated, and thus it is acceptable to assume the vehicles are operating within these regimes. Figure 2.1 shows the two vehicle bodies and their associated coordinate frames that will be discussed in this thesis. Before moving on to the model development, it is useful to discuss common coordinate frames and forces that will be used for discussion throughout this thesis.

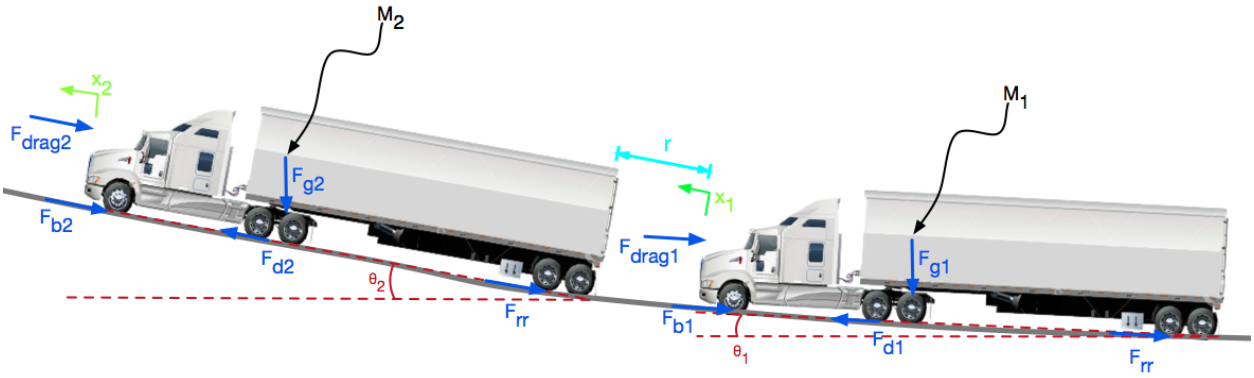


Figure 2.1: Schematic of Two Heavy Freight Vehicles

The forces acting on the bodies in order of their potential magnitude are

- F_b -The braking force applied by foundation brakes.
- F_d -The driveline force, the effective force applied by the engine including the positive engine torque produced with throttle applied and the negative torque produced by the engine retarder or Jake brakes.
- F_{drag} -The force of aerodynamic drag upon the body
- F_g -The force of gravity applied by a non-zero grade (θ) on the bodies with masses M_1 and M_2
- F_{rr} The rolling resistance is the force acting on the bodies opposing their motion due to tire deformation

Two forces are ignored in this diagram and those are the drive line friction and damping forces. The driveline damping forces are included in the driveline free body diagrams further on in this section and are represented in the final equation of motion. The driveline friction forces, however, are not included in the final equation of motion and are assumed to be lumped in with the rolling resistance F_{rr}. The friction forces are shown in the steps along the process for completeness, but are removed once the equations are further manipulated.

The primary outputs for each vehicle are the position of each vehicle (x_1 and x_2), their respective velocities ($\dot{x}_1$ and $\dot{x}_2$ which may also be written as v_1 and v_2) and the acceleration of each vehicle ($\ddot{x}_1$ and $\ddot{x}_2$ or a_1 and a_2). From these we can define a relative measurement "range" which may also be described as the inter-vehicle spacing or gap. The range between the vehicles r is effectively difference in the positions of the two vehicles minus the length of the proceeding vehicle (L).

$$r = (x_2 - L_2) - x_1 \tag{2.1}$$

The range rate is simply the time derivative of the range and can be thought of as the difference in the relative velocities of the two vehicles.

$$\dot{r} = \dot{x}_2 - \dot{x}_1 = v_2 - v_1 \tag{2.2}$$

The combined dynamics of these two vehicles are made up of the dynamics of two individual trucks. To simplify the discussion, the dynamic equations for a single truck are derived. The free-body diagram for a single vehicle is shown in Figure 2.2. The forces acting on the vehicle are the same as described above.

With the assumptions mentioned previously, the model can be resolved into a single degree of freedom first-order differential equation. The derivation of this simplified model follows in the next section.

For a single truck body, the equation of motion is described in Equation (2.3):

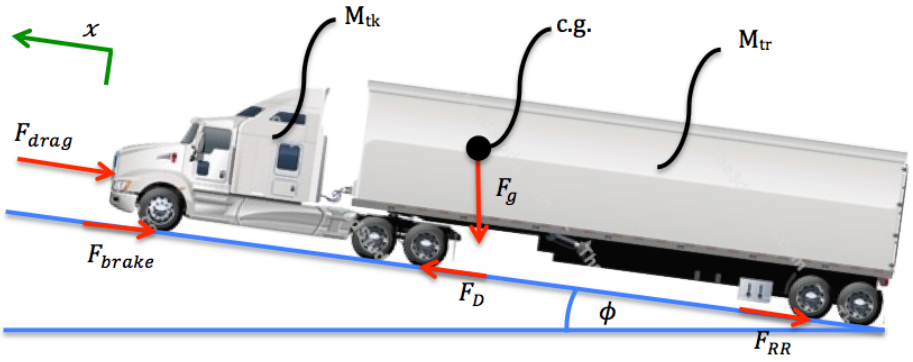


Figure 2.2: Single Truck Free-Body Diagram

$$\nwarrow \sum F_x = M_{total} * \ddot{x} = F_D - F_{brake} - F_{RR} - F_{drag} - M_{total} * g * sin(\phi) \tag{2.3}$$

Where the total mass is simply the combination of the tractor and trailer.

$$M_{total} = M_{tk} + M_{tr} \tag{2.4}$$

It is important to note that the drag force is proportional to the velocity squared i.e., as speed increases the force that the engine must overcome increases exponentially. This is why there are large fuel benefits to drafting in a convoy. This drafting effectively lowers the drag coefficient (Cd) and thus lowers the total force the engine must overcome. The equation for air drag is shown in Equation (2.5)

$$F_{drag} = \frac{C_D * \rho * A * V^2}{2} \tag{2.5}$$

Note that this non-liner equation must be linearized to use standard tools for solving differential equations. The solution will be linearized about some nominal operating velocity V_0, in practice this is the set speed of the convoy. Using Taylor series expansion for linearization Equation (2.5) becomes:

$$F_{drag} \approx -\frac{C_D * \rho * A * {V_0}^2}{2} + C_D * \rho * A * V_0 * V \tag{2.6}$$

In order to find the drive force in terms of engine torque and transmission gear, the derivation must be incorporated. After this inclusion, the acceleration of the vehicle can be written in terms of the engine torque applied. To resolve the effective engine force at the wheel free body diagrams of each of the interlinking mechanisms must be drawn. Starting at the engine, Figure 2.3 shows the inertia, damping and friction torque as well as the torque produced by the engine. The reaction force of the transmission on the engine is also shown. From this the second equation of motion is derived and shown in Equation (2.7).

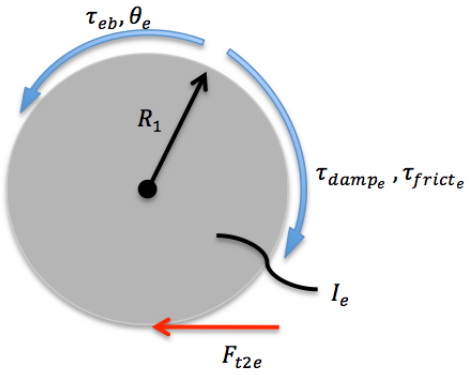


Figure 2.3: Engine Flywheel Free-Body Diagram

$$\circlearrowleft \sum M = I_e * \ddot{\theta}_e = \tau_{eb} - \cancel{\tau_{frict_e}} - \tau_{damp_e} - F_{t2e} * R_1 \tag{2.7}$$

The next linkage between the engine and the wheel is the transmission and drive shaft shown in Figure 2.4. The associated friction, damping and reaction forces are shown with the transmission's inertia. Summing the moments about the bodies central axis yields Equation (2.8)

$$\circlearrowleft \sum M = (I_t + I_{ds}) * \ddot{\theta}_t = -\tau_{frict_t} - \tau_{damp_t} + F_{e2t} * R_2 - F_{diff2ds} * R_3 \tag{2.8}$$

The final free-body diagram is that of the differential and wheel. Figure shows the ground reaction force and the forces propagating from the engine. The final damping and

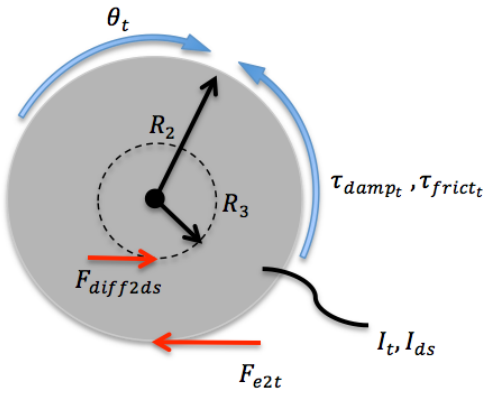


Figure 2.4: Transmission Free-Body Diagram

friction forces are also noted. Again, summing the moments around the drive axle gives the last equation of motion shown in Equation (2.9.)

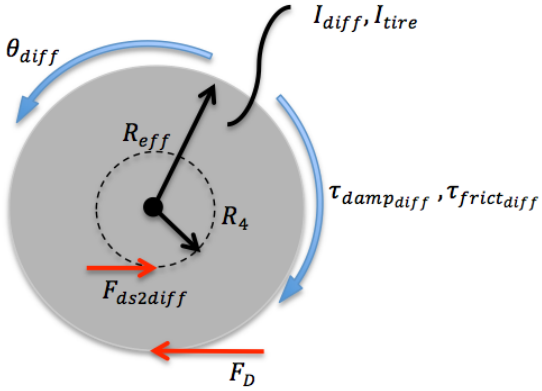


Figure 2.5: Differential Free-Body Diagram

$$\circlearrowleft \sum M = (I_{diff} + I_{tire}) * \ddot{\theta}_{diff} = -\tau_{fricts_{diff}} - \tau_{damp_{diff}} + F_{ds2diff} * R_4 - F_D * R_{eff} \quad (2.9)$$

With the reaction forces shown in Equations (2.7 - 2.9)it is now possible to resolve the engine torque in the body frame as the driveline force. To give a better understanding of the linkage of all of the previous bodies, Figure 2.6 is presented.

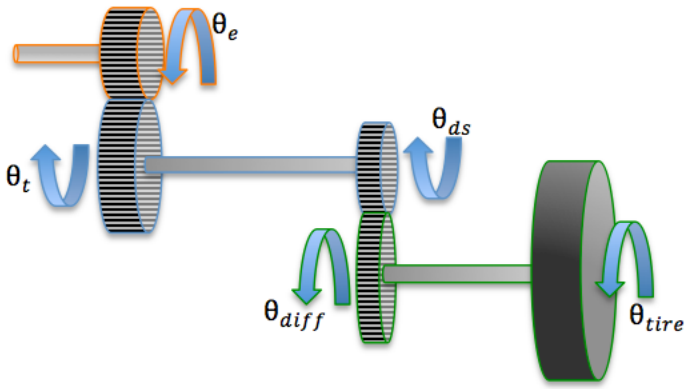


Figure 2.6: Spin-Rate Linkage Diagram

Lastly, to resolve the lever arms of the resultant forces into terminology typically used in vehicle dynamics discussions Equations (2.10 and 2.11) show the gear ratios of the transmission and differential respectively.

$$N_T = \frac{R_2}{R_1} \tag{2.10}$$

$$N_{diff} = \frac{R_4}{R_3} \tag{2.11}$$

With the assumption of no slip mentioned previously, all of the spin rates of the individual links can be related directly to the vehicles speed. This is useful as it is now possible to write all of the previous equations of motion into one first order linear differential equation. Before doing so it is useful to bring out a few lumped parameters and define them separately. Equation (2.12) shows the effective mass of the whole vehicle including the driveline inertias. It is interesting to note the large impact on the effective mass of the two gear ratios N_T and N_{diff}, as they augment the effective mass by the square of their magnitude.

$$M_{eff} = M_{total} + \frac{((I_t + I_{ds} + I_e * {N_T}^2) * {N_{diff}}^2 + (I_{diff} + I_{tire}))}{{R_{eff}}^2} \tag{2.12}$$

The second parameter that can be pulled out is the effective damping. The combines all of the effects of driveline damping in the bearings and air drag parameters the account for the variation from the nominal drag. The effective drag is shown in Equation (2.13).

$$B_{eff} = \frac{(b_t + b_e * {N_T}^2) * {N_{diff}}^2 + b_{diff}}{{R_{eff}}^2} + C_D * \rho * A * V_0 \tag{2.13}$$

Combining all of the equations of motion gives a simple equation that describes the acceleration ($\dot{V}$) expected as a function of the current velocity and any input.

$$\dot{V} = \frac{-B_eff * V + \frac{N_{diff} N_T * \tau_e}{R_{eff}} - F_{brake} - F_{RR} + \frac{C_D * \rho * A * {V_0}^2}{2} - M_{total} * g * sin(\phi)}{M_{eff}} \tag{2.14}$$

Regarding the inputs it is important to note that there dynamics associated with both of the primary inputs (engine torque and braking force). For the work presented here, a first-order lag is used for these dynamics. The reason for doing this rather than using the complex modeling suggested in some publications [17] is two-fold. Firstly, the simplification makes controller design much more straightforward and the approximation is accurate enough to achieve the desired results. The second reason is that the method of producing the braking force and engine torque is somewhat abstracted. Meaning, lower-level control of the engine and brakes is not possible and dynamics such as turbo lag and proportioning valve lag cannot be altered. The brake control system and engine control systems are designed by the OEM component manufacturers and they handle all of the lower-level dynamics. The equations for the first order lag for each of these inputs are as follows and shows the dynamics from desired input to actual output.

$$T_c * \dot{F_b} + F_b = F_{des} \tag{2.15}$$

$$T_c * \dot{\tau_e} + \tau_e = \tau_{des} \tag{2.16}$$

These equations are used as transfer functions in the block diagrams used for controller synthesis. The latency for the engine in particular varies based on vehicle speed, however,

the assumption that this control scheme is primarily meant for highway speeds allows us to use a single value for the time constant as the variation in vehicle speed is minimal. The latency in the brake system has appeared to be consistent independent or other vehicle variations.

2.2 Parameter Estimation and Measurements

Before the model developed in the previous section can be useful, it is necessary to first measure or estimate all of the vehicle parameters used in the dynamic equation. Some of these measurements are easy to obtain such as gear ratio which is actually populated on the vehicle CAN bus and can be read directly. Other measurements such as the damping coefficients and inertias cannot be easily measured without using some estimation techniques. The MATLAB system identification toolbox was used to help identify the effective masses and damping values. It should be noted that it can be difficult to separate individual variables as they are lumped in the dynamics of the vehicle. Model parameters estimated for the Peterbilt 579 are shown in Tables 2.1 and 2.2

Table 2.1: Estimated Model Parameters

Parameter	Value
Tractor Mass	8,8759 kg
Trailer Mass	6,350 kg
Engine Inertia	2.80 kg m^2
Transmission Inertia	0.265kg m^2
Drive Shaft Inertia	0.013 kg m^2
Differential Inertia	0.018 kg m^2
Wheel Inertia (all wheels)	1700 kg m^2
Engine Damping	2.1 $\frac{Nm}{s}$
Transmission Damping	1.18 $\frac{Nm}{s}$
Differential Damping	10.1 $\frac{Nm}{s}$
Drag Coefficient	0.79
Drag Frontal Area	8.52 m^2
Coefficient of Rolling Resistance	0.005

Table 2.2: Gear Ratios

Gear/Parameter	Ratio/Diameter
1	11.06
2	10.2
3	7.062
4	4.984
5	3.966
6	2.831
7	2.03
8	1.417
9	1
10	0.74
Differential	2.75
Wheel Radius	0.508 m

2.3 Model Validation

To test the accuracy of the model a heavy acceleration test at full throttle was performed at the NCAT test track using a Peterbilt 579 with an unloaded van trailer. A comparison between the longitudinal acceleration obtained from the vehicle CAN bus and that predicted by the mode derived above are shown in Figure 2.7.

It can be seen from the figure above that the model captures the general trends accurately. In fact, the largest errors occur in the transients during a gear shift. This is due to the violation in the assumption that the driveline is always engaged. This, however, is not a big concern for the convoy controller design because its primary mode of operation will be at relatively constant speeds with few gear shifts.

2.4 Simulation Environment

Before testing the results of the controllers described previously in the real world, a high-fidelity simulation environment is set up to test and tune controllers easily. MATLAB, SIMULINK and a high-fidelity vehicle testing software for heavy trucks called TruckSim

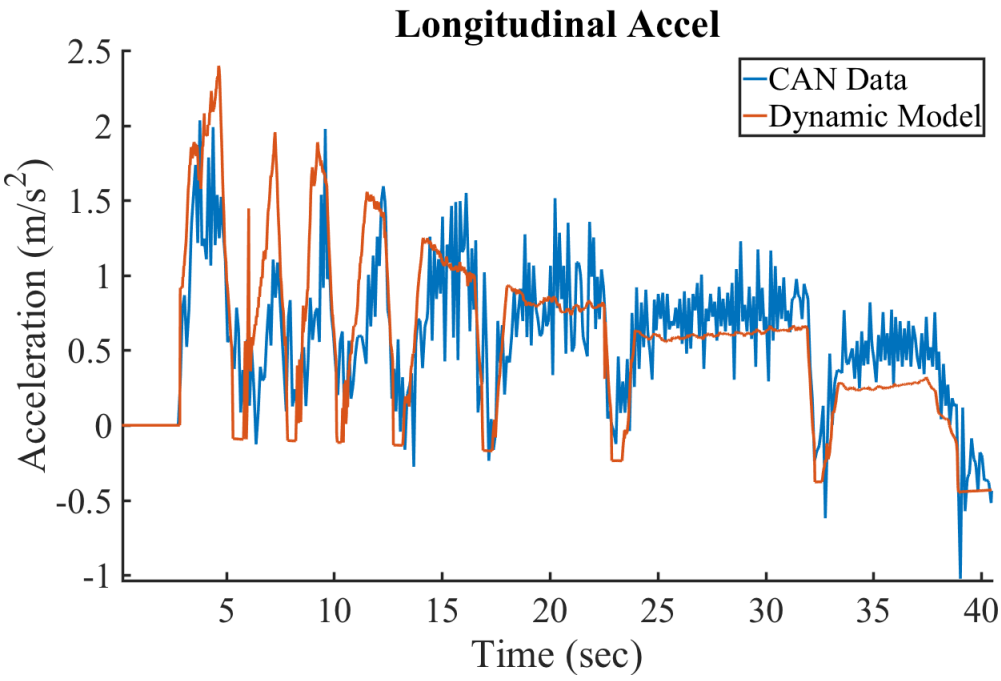


Figure 2.7: Model Validation Against CAN Acceleration Data

were used. These simulation environments were developed to mimic the real world system by ensuring good agreement with the vehicle dynamic model, reasonable sensor noise and latency in communication.

2.4.1 MATLAB Simulation

For ease of controller design MATLAB is utilized as the first phase of simulation testing. A vehicle class was developed to do initial testing of control scheme. This method improves the speed of controller development, as it is easier to first do initial testing in MATLAB. The control algorithm can then be ported to SIMULINK for integration with TruckSim. The non-linear vehicle model described previously is used within the truck class developed. This model was helpful for both parameter estimation and experimental controller evaluation.

2.4.2 TruckSim Simulation

In addition to the MATLAB environment developed, additional software was used to perform a higher-fidelity testing of control and estimation schemes. To simulate the large-scale vehicle dynamics of multiple vehicles TruckSim and SIMULINK were utilized. TruckSim has many built-in vehicle models with different power trains. This allows for the testing on various vehicle types and driving scenarios. Figure 2.8 shows a convoy composed of four vehicles of the same vehicles within the TruckSim and SIMULINK environment developed.

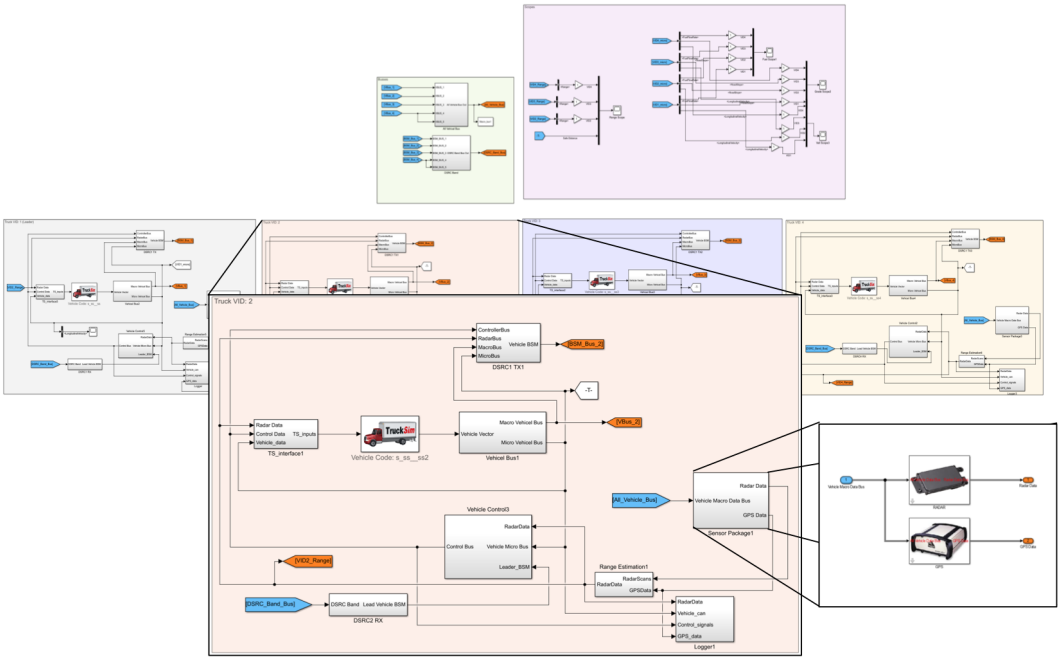


Figure 2.8: TruckSim 4 Vehicle Convoy Diagram (With Detail of a Single Vehicle)

Each vehicle has a given sensor set which includes both a RADAR and GPS simulated block. The GPS block simply generates expected sensor noise for the DRTK range solution. The RADAR sensor is slightly more complex as it includes both the sensor noise expected by the Delphi ESR RADAR and the effects of tracking multiple objects. This allows for the effects of occlusion allowing for testing of vehicle cut-ins and various other complex scenarios.

To ensure that the vehicle models in TruckSim also match the model developed in earlier in this chapter, another constant throttle acceleration run was performed to compare the TruckSim model to the controller model. Figure 2.9 shows the resulting acceleration over time. Again, the model matched well during constant gear acceleration. During a shift there is still some error in the model. However, this is simply due to the fact that TruckSim uses a higher fidelity non-linear model which can capture more complex motion such as shift delays, non-linear torque converter dynamics and non-linear tire dynamics.

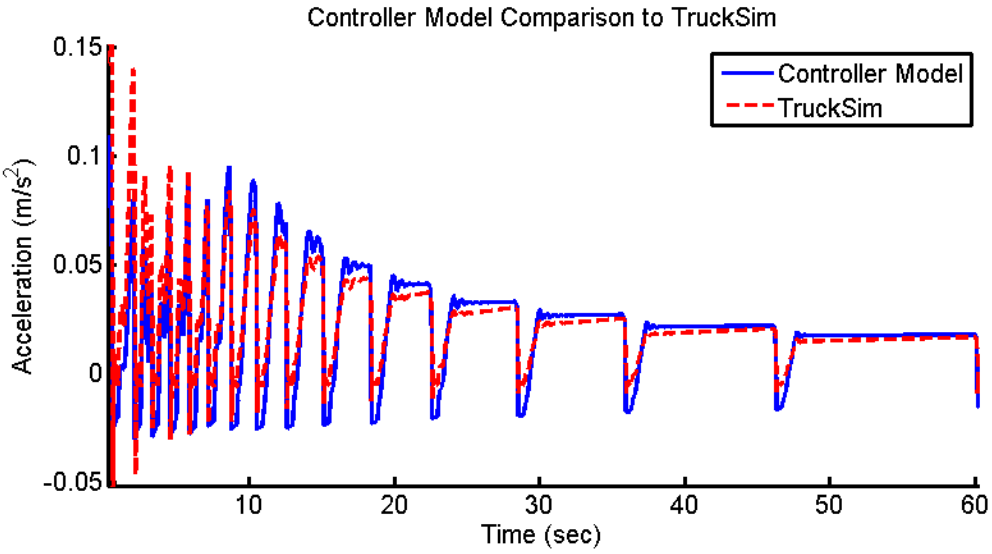


Figure 2.9: TruckSim Model Validation

With good agreement between the controller model and the simulation model, a simulation with multiple vehicles can be performed. As the number of vehicles increases the complexity of the simulation compounds and can become overwhelming. For this reason, sensor blocks and controllers were implemented to make the simulation as extensible as possible. In this way, multiple vehicle types can be tested and sensor measurements can be corrupted to test the robustness of control. Another aspect that was simulated within TruckSim is the effect of drag reduction based on the CFD analysis performed in [1]. This vehicle drag reduction with vehicle gap allows the testing of basic fuel savings predicted by drafting. Figure 2.10 shows the fuel rate of a two-vehicle convoy at various gaps. The transients at the start of the test are simply due to the way in initializes tests with zero throttle

input ,thus, causing a spike in fuel consumption as the controller maintains the gap. This reduction of fuel rate shows the impact of the estimated drag changes due to drafting. The variation in drag force from the nominal free stream drag can be thought of as a disturbance on the controller and is another level of fidelity to ensure accurate testing in the simulation.

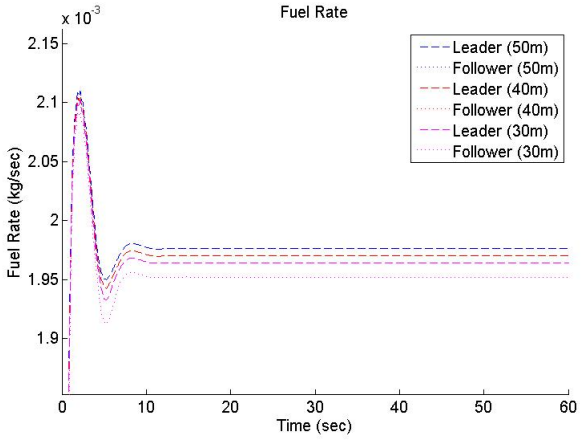


Figure 2.10: TruckSim Fuel Rate at Various Gaps

Further use of the vehicle simulation environment is detailed in Chapter 4 during the initial controller evaluation step. This simulation testing environment allowed a level of confidence when moving controller designs from simulation to validation and testing. There is one primary difference in the simulation environment and the interface utilized on the Peterbilt 579s and that is the lower-level control scheme. In TruckSim the engine and brake interface is in the form of throttle percentage and brake line pressure. In order to make this as close to the real vehicles as possible, similar bandwidths were designed for the lower-level controllers.

Chapter 3

Estimation Schemes

The design of a vehicle headway controller first requires a measurement of the spacing between each vehicle in the convoy. In standard ACC systems, a single-ranging device is used to estimate this gap. Production systems today use sensors such as RADAR or LiDAR to measure this range. However, CACC convoy has the benefit of a communication link between the vehicles allowing more sophisticated estimation techniques.

3.1 InterVehicle Spacing Estimation

The estimation scheme developed in this thesis leverages both RADAR and GPS measurements and fuses them together. The RADAR provides range, range rate and azimuth measurements for each "track" of the RADAR. These RADAR measurements are published at 20Hz but are noisy and have a relatively low resolution. The GPS system by itself can only provide accuracy of approximately 0.5 m when taking the difference between two standard position solutions from a GPS receiver. With two receivers working in concert, however, it is possible to remove some of the common mode errors that degrade the solution of a single receiver. By combining the measurements, a relative position solution can be calculated that is accurate to 2-3cm. Figure 3.1 shows the measurements and their coordinate frames relative to a two trucks.

In contrast to the RADAR, Dynamic Base Real-Time Kinematic (DRTK), a differential GPS solution, has a much slower updated rate with much higher accuracy [18]. This measurement set is combined together in a Kalman filter to give accurate fused measurements at 20Hz.

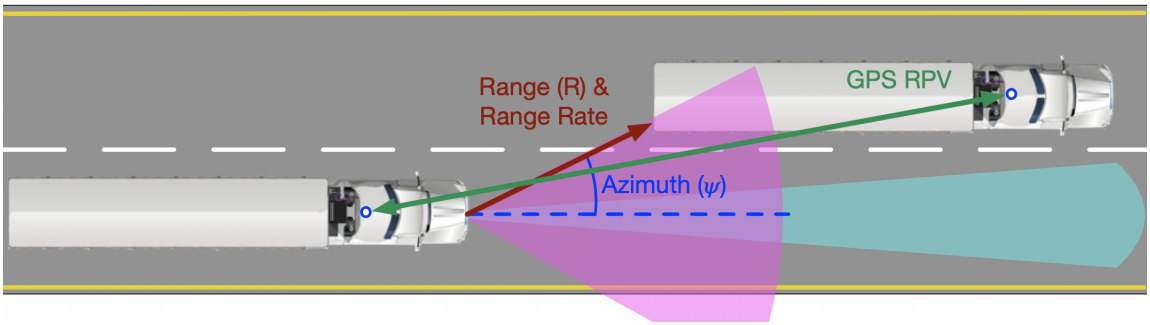


Figure 3.1: Estimation Polar Coordinates

3.1.1 Dynamic Base Real-Time Kinematic Range

DRTK is an algorithm used to find a relative position vector between two receivers. It is based upon the same principles as Real-Time Kinematic (RTK) GPS positioning. In an RTK positioning system, one receiver is static and has a known position, and is referred to as the "base" receiver. The second receiver is considered a "rover" receiver. The position solution for the rover can be found globally with an accuracy 2-3 cm if the corrections from the base receiver are sent to the rover. Naturally, this positioning scheme requires a communication link between the base and rover. This is typically done with radios, however, as the use of RTK is becoming common many public corrections are being broadcast over the Internet and can be used for a given area up to 20 Km away from the base station receiver. While it may be possible to use these base stations for local corrections it is not practical to use them when traveling along highways, where the base stations would have to change frequently and may not be available in a specific area. It is desirable, therefore, to use the variation of RTK positioning known as DRTK.

As the name suggests the primary difference between the two algorithms is that RTK requires one receiver to be a static "base" receiver with a known global position, and DRTK can have a moving base receiver. The global position is no longer known to a high degree of accuracy but, the relative position accuracy is not diminished. The primary use of DRTK in the case of the convoy is to know the relative position between the two trucks accurately, and

global position is not required. The algorithm for calculating a DRTK solution is somewhat complex and is described in more detail in; [18] however, a brief overview is given here.

DRTK relies on the accuracy with which a receiver can measure the phase of the GPS carrier signal. The carrier signal of GPS measured at the receiver is compared to the carrier signal generated by the receiver. The phase angle between these two signals can be measured accurately. Accurate positioning requires knowledge of the total number of cycles between the receiver and the satellite. This is a whole number and is referred to as the integer ambiguity, N. As in normal code based GPS positioning, the solution is degraded by ionospheric and tropospheric delays, electrical noise and receiver clock bias. Considering all of these, the following equation shows the carrier measurement model(ϕ_A^s) for a receiver A and satellite S.

$$\phi_A^s = \lambda^{-1}[r^s + I_\phi + T_\phi] + \frac{c}{\lambda}(\Delta t_A + \Delta t^s) + N + \epsilon_\phi \tag{3.1}$$

In Equation (3.1) the true range to the satellite is represented by r_s, λ is the carrier signal's wavelength, I_ϕ and T_ϕ represent the ionospheric and tropospheric delays, c represents the speed of light, both of the Δt measurements represent both the receiver and satellite clock bias, finally ϵ_ϕ is the electrical noise. Two more equations are used to calculate the relative position vector between (RPV) between the two receivers they are known as the single difference and double difference. These are shown between two receivers A and B in Equations (3.2 and 3.3) [19].

$$\Delta\phi_{AB}^s = \lambda^{-1}(\Delta r_{AB}^s) + \frac{c}{\lambda} * \Delta t_{AB} + N_{AB} + \epsilon_{\phi_{AB}} \tag{3.2}$$

$$\Delta\phi_{AB}^{s_1 s_2} = \lambda^{-1}\Delta r_{AB}^{s_1 s_2} + N_{AB}^{s_1 s_2} + \epsilon_{\phi_{AB}}^{s_1 s_2} \tag{3.3}$$

A Kalman filter is used to provide an estimate of the single difference carrier ambiguities. These single difference estimates are then transformed into double difference estimates prior

to solving for the integer ambiguity. As can be seen in Equations (3.2 and 3.3) these two steps remove common mode errors such as atmospheric delays in the case of the single difference and receiver clock errors in the double difference equation. The Kalman filter used in the DRTK algorithm finds floating point carrier ambiguity values. However, this ambiguity must be an integer as it represents the whole number of carrier cycles. Simply rounding this floating point appears to be the most straightforward approach to solving for the whole integer number. However, the floating point measurements from the Kalman filter are highly correlated and simply rounding to these values to the nearest whole number does not take full advantage of all of the available information in the measurement covariance. For this reason, an algorithm known as the least-squares ambiguity deceleration adjustment (LAMBDA) is used to fix the integer values required for precise measurements. More detail about the LAMBDA method can be found here [20].

The final result gives a double difference range, relative to the unit vectors to the satellites and the relative potions vector (RPV) between the receivers. This relationship can be seen in Equation (3.4)

$$\Delta r_{AB}^{s_1 s_2} = -(\vec{u}_A^{s_1} - \vec{u}_A^{s_2}) * \vec{b}_{AB} \tag{3.4}$$

Ultimately the measurement desired is the RPV ,$\vec{b}_{AB}$, between the two receivers. This RPV is found in the ECEF coordinate frame and must be transformed into the local frame before use. This is done by using the GPS course measurement as a heading for the vehicle. This is generally a good assumption for vehicles traveling straight on highways as there is little sideslip during strait line driving. The final result is a relative position vector from the rear vehicles receiver to the leader. The norm of this value gives the range to the leader's receiver. Figure 3.2 shows the geometry of the GPS antenna locations and the desired range measurements. The final range calculation from the RPV is shown in Equation (3.5)

$$range = \left\| \vec{b}_{AB} \right\| - L_1 - L_2 \tag{3.5}$$

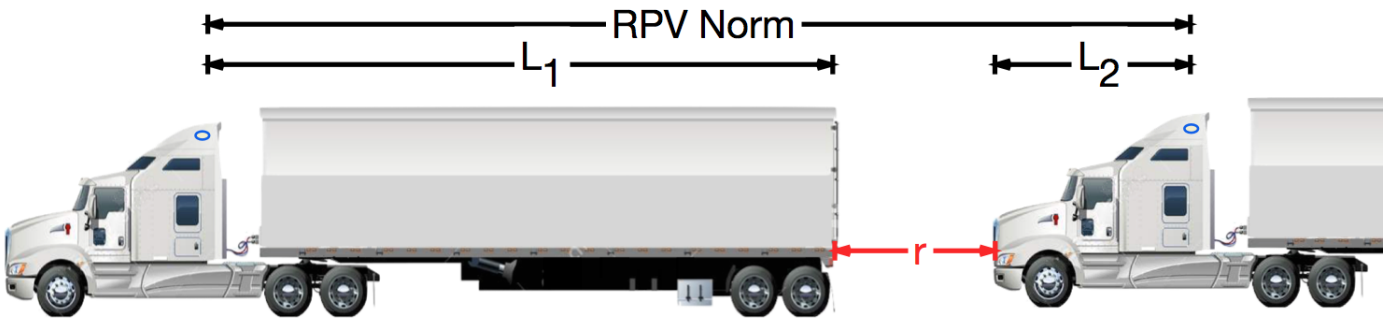


Figure 3.2: RPV Vehicle Geometry

Again, the accuracy of this range measurement is 2-3cm. The range produced by the DRTK us published at 2Hz in this work. This lower rate is used to decrease network traffic between the vehicles and because carrier measurements on a GPS receiver can be degraded as the publishing frequency increases. It is necessary to use an additional measurement with a higher update frequency to provide a robust range estimate for vehicle control. A RADAR was chosen for these purposes.

3.1.2 RADAR Range Solution

A forward-facing RADAR has become commonplace in production passenger vehicles today and can be used for both ACC and collision avoidance systems [5]. These systems are gradually making their way into the heavy freight trucks as well. For this reason a RADAR is selected as a complementary sensor to be fused with the DRTK range estimates. The RADAR provides two primary benefits to the estimation scheme. Firstly, it provides a higher frequency measurements which can be used for the controller thus increasing the total bandwidth of the feedback system. Secondly, the RADAR measurement does not rely on any sort of communication with the lead vehicle and, therefore, increases the reliability of the system as a whole. However, the RADAR is not without drawbacks. Its measurements are very noisy and it can be difficult to select relevant targets tracked.

Before the RADAR measurements can be used for control the primary track, representing the leading vehicle must be selected from a group of 64 tracks. Not all tracks return

relevant measurements at each time-stamp and these bad returns must be filtered out. All of this filtering can be done on the chosen RADAR, the DELPHI 2.5 ESR, but vehicle state information must be fed to the RADAR in order to compute a relative solution. The state information required by the RADAR is as follows:

- Velocity
- Yaw rate
- Steering wheel angle
- Curvature

These allow the radar to group objects into moving and stationary objects primarily by using the range rate measurement found by using the Doppler effect. In addition to the range rate measurement the steer angle and estimated curvature are used to help classify the objects by their azimuth angle. These objects are segmented into ones that are "in-path" and those that are not. Again, all of this filtering happens on the device if properly mounted and supplied with the required measurements. The final result is an object track that is grouped into a single in-path object with associated range, range rate and azimuth measurements.

3.1.3 RADAR and DRTK Fusion Algorithm

As mentioned previously, some manner of fusing both RADAR measurements and DRTK range measurements into a robust solution for both range and range rate is required. A Kalman with a simple constant velocity time update is selected for this thesis. Both the RADAR and GPS measurements are implemented into the filter as measurement updates. Figure 3.3 shows the estimation block diagram. In the figure one can see that the DRTK measurement is strictly used as a measurement update and only when a fixed integer solution is computed. DRTK is also used to initialize the state estimate. This initialization step is required due to the fact that the RADAR measurements must be validated using

the Chi-Squared validation gate as shown in [21], which requires an initial state estimate to create the validation gate. The RADAR range and range rate measurements are only used if they pass the validation process to determine if the RADAR track is indeed tracking the leading truck. This is a necessary step as the radar tracks may move from various objects in their field of view. Not all of these measurements should be used to update the state estimate.

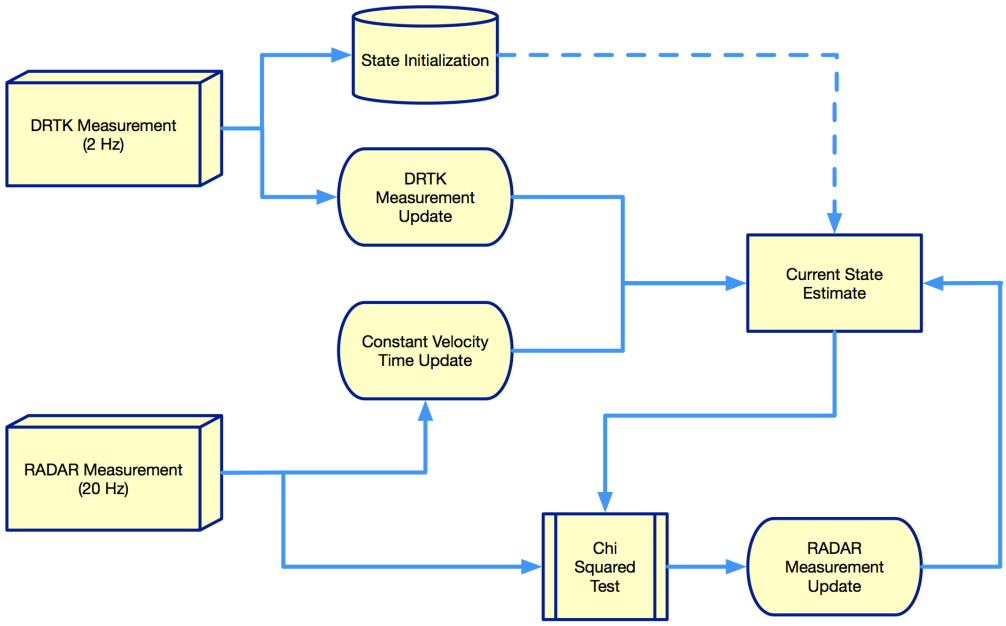


Figure 3.3: Estimation Block Diagram

To better explain the formulation of this filter the desired state and measurements must be defined. They are simply the range and range rate between the preceding and following trucks.

$$x = y = \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \tag{3.6}$$

The time update is performed at the rate of the RADAR's publishing frequency and is a basic model assuming a constant velocity. The discrete state transition matrix is shown in Equation (3.7).

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.7}$$

Again, the DRTK measurement is applied as a standard measurement update if the solution is a fixed integer solution. The RADAR is slightly more complex and must first go through a validation process. This validation is known as the Chi-Squared test. First the innovation covariance is calculated.

$$S = C * P * C^T + R \tag{3.8}$$

Where C is the measurement model , P is the state estimate covariance , and R is the measurement covariance matrix. The measurement model is defined as

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.9}$$

$$R(RADAR) = \begin{bmatrix} 0.019 & 0 \\ 0 & 0.00677 \end{bmatrix} \tag{3.10}$$

The DRTK measurement covariance only contains the range measurement (0.0012 was used in this work) In order to check if a measurement is valid, the measurement innovation $\tilde{y}$must first be calculated as follows

$$\tilde{y} = y - \hat{y} \tag{3.11}$$

This innovation can then be used to check the validity of each the measurements used to calculate the innovation using a threshold (2 is used as the threshold in this thesis). The measurement is valid if it passes the following test.

$$\sqrt{\tilde{y}^T * S^{-1} * \tilde{y}} < Threshold \tag{3.12}$$

The threshold shown here is used as a tuning parameter to determine the number of radar measurements accepted. The validation threshold is shown in the measurement space with example valid and invalid measurements in Figure 3.4. All measurements that are within the validation gate are used as measurement updates.

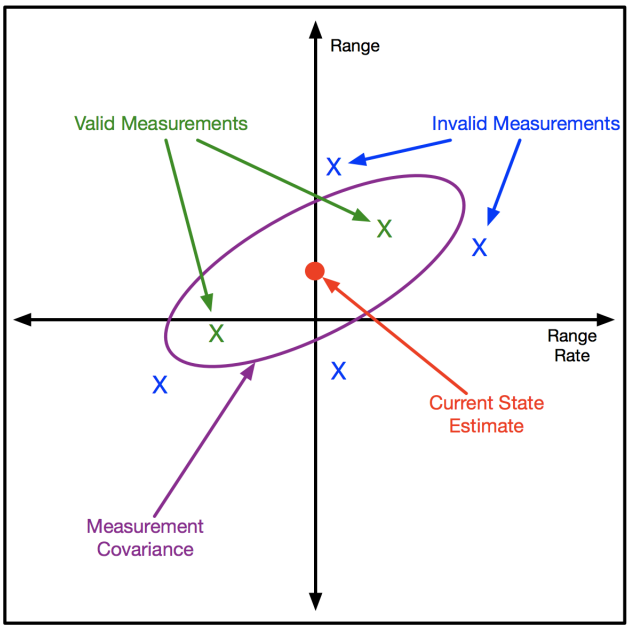


Figure 3.4: Estimation Measurement Validation

It is worth noting that the Chi-Squared validation gate is performed separately for the range rate measurements as well. This is done because, for some measurement epochs, a range measurement may be invalid while the range rate measurement may be valid. This

happens when the radar object tracking location shifts to another part of the vehicle. For instance, the RADAR may be tracking the bumper and may shift to the axle. In this case the range rate measurement would be valid while the range would be invalid. Only the range rate measurement is used as an update in these cases.

Figure 3.5 shows the performance of the estimator in real time. Note the large vertical spikes in the RADAR signal are due to incorrect objects tracked. Note these objects are rejected by the validation gate and would not be used as measurement updated during these periods. It can also be seen in the figure (from 315 seconds to 345 seconds)that there is a significant RADAR outage where the RADAR does not return any "in-path" objects. During these outages, the fused solution still performs well as it is able to run on the GPS aided solution.

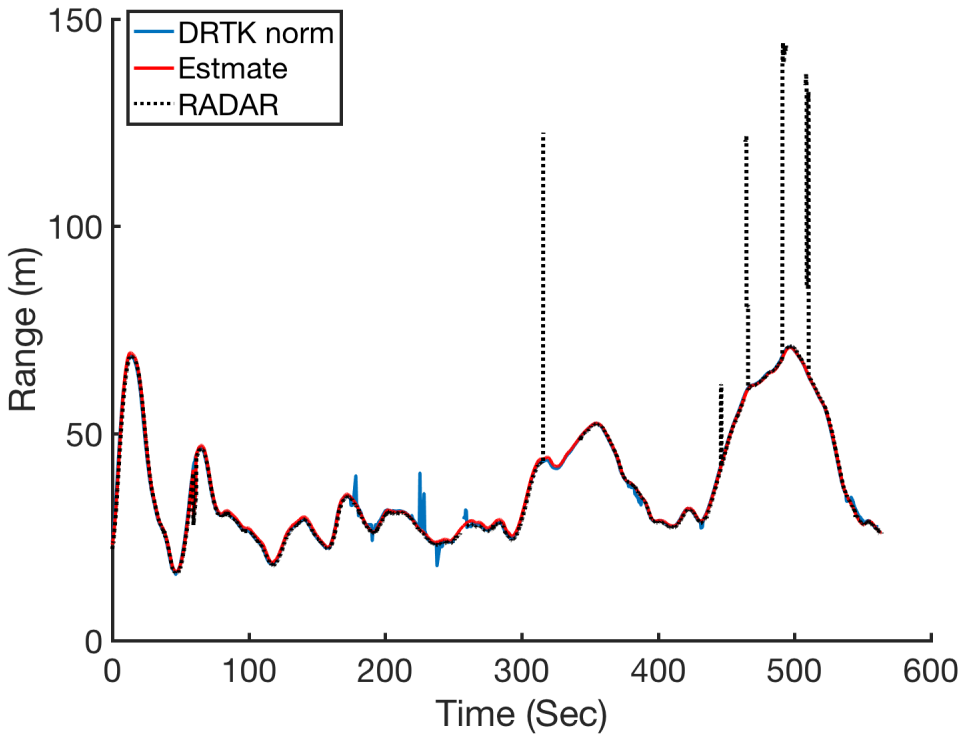


Figure 3.5: Estimation Performance With Chi-Squared Validation

Chapter 4

Control Algorithms

With a good estimate of the inter-vehicle gap, it is now possible to design a headway control scheme. This headway controller must actuate the engine throttle, retarder and foundation brakes to achieve the desired gap. The controller must ensure each truck is asymptotically stable individually as well as string stable as a convoy. These criteria will ensure safe operation of the convoy as a whole while improving fuel economy due to the decreased drag if the inter-vehicle spacing is small.

4.1 Classical Cascaded Controller Design

Using the mechanical modeling described in Chapter 2, a controller can be designed which divides the overall control into two segments. This segmentation is achieved through a cascaded approach where two classical control feedback loops are implemented in series. The innermost loop controls the speed of the following vehicle. This desired speed is set as the lead vehicle's speed with the addition of a supplemental velocity, which is determined by the outer headway control loop. The headway control loop uses the range and range rate to determine the desired supplemental velocity required to close the gap according to the designed specifications. Figure 4.1 below shows the block diagram of this control architecture.

The control specifications determine the aggressiveness of the controller. For heavy vehicle convoying, there is a trade-off between control aggressiveness and drag reduction. To achieve the best fuel results it is important to strike a balance between controller actuation, which causes higher fuel consumption in the transients, and fuel economy savings from drafting. Thus, the gains chosen for fuel savings implement a relaxed controller which

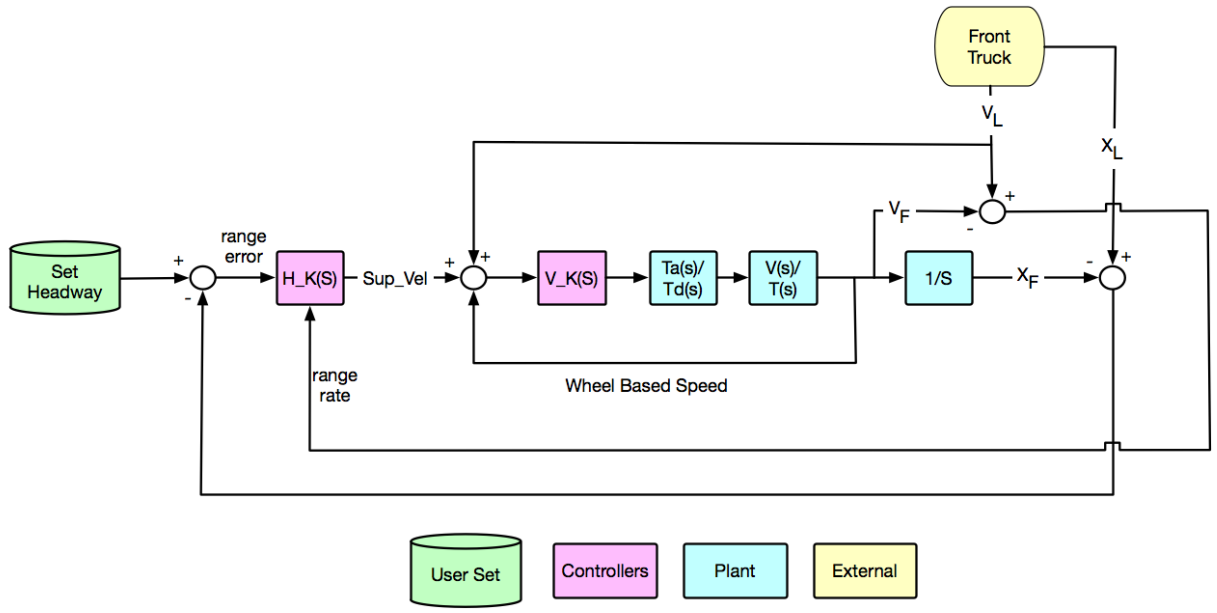


Figure 4.1: Cascaded Control Block Diagram

attempts to close the gap for drafting benefits when the fuel required does not outweigh the potential drafting benefits.

4.2 Feed Forward Control

Much of the control effort required to maintain a constant gap can be predicted by analyzing the steady state outer loop controller where the vehicles are at a constant gap. In this case the goal of the inner-loop controller is to maintain a zero range rate, i.e., the relative velocities of the two vehicles should be maintained at zero.

$$v_{follow} = v_{lead} - \dot{r} \tag{4.1}$$

The brake force, engine torque and retarder torque can be combined into a single total effective torque term, T_{eff}. This term can also be thought of as two components: a torque variation(T) and a steady state term T_{ss}.

$$T_{eff} = \delta T + T_{FF} \tag{4.2}$$

The steady state term can be found by looking at the dynamic model developed in Chapter 2. This analysis can be done by inserting the equation for $\dot{r}$,Equation (4.1, and its time derivative into Equation (2.14). Since the inner loop can be thought of as a simple regulator if the headway is at steady state, this will yield the total torque required to maintain consistent speed between leader and follower.

$$M_{eff} * (a_{lead} - \ddot{r}) + B_{eff} * (v_{lead} - \dot{r}) = \frac{N_{diff} * N_T}{R_{eff}} * \tau_e - F_{brake} - F_{RR} - F_{drag} - M_{total} * g * sin(\phi) \tag{4.3}$$

Breaking the dynamics down like this shows the dynamics of the lead truck and its impact on the follower in terms of total torque required. To do this it is first necessary to move all of the range and its derivatives parameters to one side. It should also be noted that the feed-forward term can be non-linear and is left in this form. This slightly effects Equation (2.14) in that it removes the linearized air drag and replaces it with the non-linear term, shown here as F_{drag} for brevity.

$$M_{eff}\ddot{r} + B_{eff} * \dot{v} = -M_{eff} * a_{lead} - B_{eff} * v_{lead} - \frac{N_{diff} * N_T}{R_{eff}} * \tau_e + F_{brake} + F_{RR} + F_{drag} + M_{total} * g * sin(\phi) \tag{4.4}$$

The right-hand side of this equation can be broken up into two terms, feed forward torque and variation torque, defined below

$$T_{FF} = \frac{R_eff}{N_{diff} * N_T} * (M_{eff} * a_{lead} + B_{eff} * v_{lead} + F_{RR} + F_{drag} + M_{total} * g * sin(\phi)) \tag{4.5}$$

$$\delta T = \tau_e - \frac{R_eff}{N_{diff} * N_T} * F_{brake} \tag{4.6}$$

This gives a new way of formulating the equation of motion for the vehicle shown in Equation (4.7).

$$M_{eff} * (a_{lead}) + B_{eff} * (\dot{r}) = \frac{N_{diff} * N_T}{R_{eff}} * (\delta T + T_{FF}) \tag{4.7}$$

Figure 4.2 shows how this new feed-forward term is applied in the inner-loop controller. In reference to the total torque required by the rear truck, the feed-forward term is dominated by the acceleration of the lead truck and the changes in road grade. Fortunately, it is relatively easy to measure or estimate both of these terms. However, it is recommended that both of these measurements be passed through a low-pass filter to decrease the variations in torque due to sensor noise.

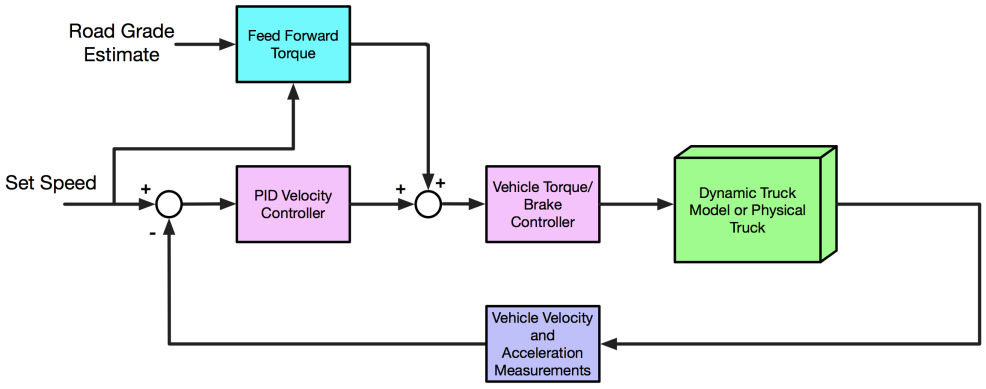


Figure 4.2: Speed Control Feed Forward

4.3 Controller State Machine

The section described above is the primary control mode, which uses the engine torque and retarder torque to maintain the gap. However, in order to achieve the benefits of drafting, heavy vehicles must maintain a gap which is much closer than a human can safely

maintain due to the slow reaction times. In emergencies, the gap must be maintained using the foundation brakes. For driver comfort and fuel efficiency, it is important to maintain a smooth transition between these control modes. To illustrate the different control modes and the transition between these modes a range vs. range-rate plot is shown in Figure 4.3. An example scenario is shown where the system is engaged in the top left part of the figure. The control system moves through Velocity and Headway Control "modes" to reach the desired set point.

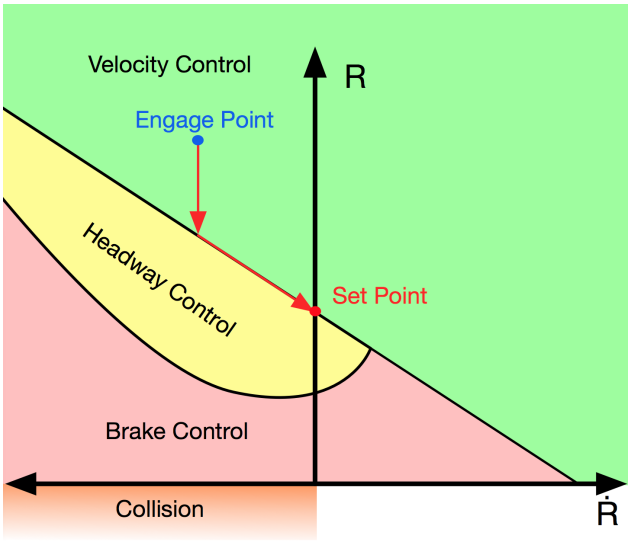


Figure 4.3: Control Modes Phase Plane

At each epoch the controller uses the range and rage-rate measurements to determine which control regime should be utilized. The driver engaged well beyond the set point and controller decides to approach at a constant relative velocity. When the following vehicles get to a set distance away from the desired following gap, it switches to "Headway Control" which smoothly closes the gap. In headway control mode all actuators (throttle, brake and, engine retarder) can be utilized to maintain the gap. The "Brake Control" region is only used for collision avoidance scenarios.

4.4 Control Gain Selection

The state machine outlined above shows the overall goal of the current control regime, but it does not provide the actual control gains selected to achieve the desired response. The inner loop is designed first with a PID controller chosen to allow placement of the closed loop poles that meet the design specifications. An over-dameped response was chosen with a settle time of 5 seconds for this thesis. The integral control is still used in this inner loop in conjunction with the feed-forward terms as there is still some model inaccuracy and uncertainties that can be compensated for with the integral control. The outer loop utilizes a PD controller as all uncertainties are accounted for in the inner loop.

For controller design, root locus techniques were used to determine the correct gains for each control mode. The inner loop was chosen to have an over-damped response. The outer loop is more relaxed, allowing a slightly under damped response to preserve fuel with less aggressive actuation. It is important to note that the control gains are only valid for one specific gear; thus, new gains must be selected for each gear. These are stored in a gain schedule where the appropriate control gains are chosen based on a discrete change in model parameters. This technique ensures correct response for all operating modes.

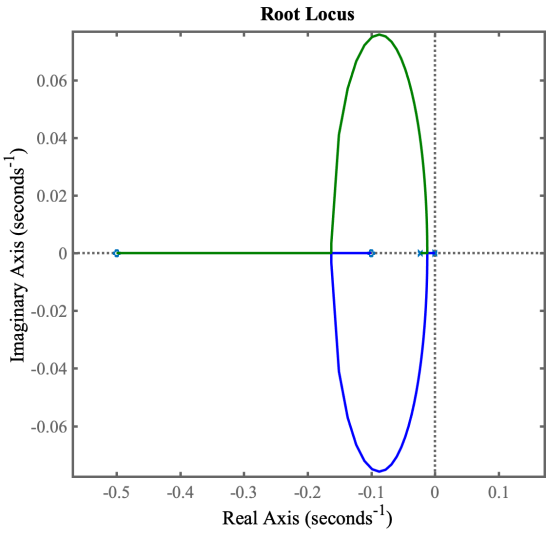


Figure 4.4: Root Locus for Inner Loop Controller

Figure 4.4 shows the root locus for the inner loop. The diagram shows the possible closed loop eigenvalues for the specific gear. The specific gear in this case is 8th gear. It was found that this is the most common gear during operations close to 50 mph. For all gears a specific gain has to be chosen as the root locus will shift as the effective mass and effective damping terms are heavily dependent on gear. To select the correct gains for each gear a gain schedule was implemented for each gear. This helps ensure stability for each possible gear.

4.5 String Stability

When designing a controller for a single vehicle, it is necessary to ensure that the system will be asymptotically stable and that the system will reach a steady state. In the case of a gap controller, the following vehicle will reach a steady state inter-vehicle spacing and should drive the relative velocity between the leading and following vehicle to zero. While this is a sufficient condition for a vehicle convoy consisting of only two vehicles, it becomes necessary to examine the stability of the convoy as a whole as the number of vehicles increases. In the case of larger convoys, the inter-vehicle spacing must be examined for all vehicles relative to that of their leader. It is necessary to ensure that disturbances in the leading vehicles do not get amplified as they propagate towards the back of the platoon. This is called string stability analysis and was first presented in [5]

The issue of string stability is present in all types of control where the individual control decisions of a single entity will have an impact on the reference of another. In fact, one of the primary reasons for developing Cooperative Adaptive Cruise Control (CACC) platoons is to help improve the string stability of longitudinally controlled platoons by introducing a communication network between the vehicles. Before examining the potential benefits of this communication channel, it is helpful to first inspect the Adaptive Cruise Control (ACC) case and discuss the requirements for string stable behavior. For this analysis a simplified model

of a two-vehicle pair is shown in Equation (4.8). This model ignores all of the dynamics discussed in Chapter 2 in order to simplify the discussion.

$$H(s) = \frac{r_i}{r_{i-1}} \tag{4.8}$$

The first requirement is that energy in the spacing error should decrease as it travels back the platoon. This effectively ensures that a disturbance occurring upstream will not get amplified as it propagates along following vehicles. On the contrary, this requirement ensures that the platoon's spacing errors will diminish further back in the convoy. To clarify further, Equation (4.9) essentially stats that the spacing errors of each vehicle will be smaller than their predecessors in the platoon.

$$\left\| H(s) \right\|_\infty \leq 1 \tag{4.9}$$

The second criterion for string stability is that the corresponding impulse response, $h(t)$, should not change sign shown in Equation (4.10). This criterion ensures that, for each vehicle in the platoon, the steady state spacing error will all be positive, which essentially means that all vehicles will not be closer than desired.

$$h(t) > 0 \tag{4.10}$$

To perform the analysis of a constant spacing control scheme on an ACC control it is best to simplify things by assuming a given acceleration can be achieved instantaneously. Effectively all of the dynamics derived in Chapter 2 are assumed to be negligible. While this simplification is not realistic, it simplifies the analysis. It should be stated that though this model is not accurate for controller synthesis any lag caused by these dynamics will amplify the issues revealed in this analysis.

Assume a basic PD control scheme for the headway yields the following dynamic equation for the i-th vehicle.

$$\ddot{x}_i = -K_p * r_i - K_d * \dot{r}_i \tag{4.11}$$

Applying this to two vehicles in a convoy results in the following equation.

$$\ddot{r}_i = \ddot{x}_i - \ddot{x}_{i-1} = -K_p * r_i - K_d * \dot{r}_i + -K_p * r_{i-1} - K_d * \dot{r}_{i-1} \tag{4.12}$$

Now it is possible to find the closed loop error dynamics in the form of a transfer function shown below.

$$H(s) = \frac{r_i(s)}{r_{i-1}(s)} = \frac{K_d * s + K_p}{s^2 + K_d * s + K_p} \tag{4.13}$$

To ensure that the gain of the bode plot for the above transfer function is not above unity the damping ratio (ζ) must be greater than or equal to 0.707. The damping ratio can be written it terms of the control gains K_p and K_d as shown here.

$$\zeta = \frac{K_d}{2 * \sqrt{K_p}} \geq 0.707 \tag{4.14}$$

Therefore, the first criterion for a string stable controller is for the control gains K_p and K_d to satisfy the following condition.

$$K_d = \geq 2 * 0.707 * 2 * \sqrt{K_p} \tag{4.15}$$

The second requirement for the design of the control gains is derived by examining the zeros in the transfer function described by Equation (4.13). Again values of K_p and K_d must be found so that the resultant gain of the controllers error dynamics must be less than or equal to one for all frequencies. In this case it sets the location of the zero of the transfer function. The location of the zero cannot be less than the resonant frequency of the system,

in this example $\sqrt{K_p}$. Thus the control gains must satisfy the following conditions.

$$\frac{K_p}{K_d} > *\sqrt{K_p} \tag{4.16}$$

It is not possible to find values for K_p and K_d that satisfies both Equations (4.15 and 4.16) and therefore, a string stable controller cannot be designed for a constant spacing policy controller. ACC systems get around this by defining the controller as instead. This alters the gap required based on the speed of the vehicles. A constant time-gap policy sufficiently alters the error dynamics to allow for a string stable platoon control scheme. A derivation of one such control algorithm and the proof of its string stability can be found in [17]. The control scheme described there sets the bandwidth of the controller by the lag introduced in the lower-level actuation. The lag of the system dynamics sets the time gap for the ACC controller. For typical ACC systems on trucks, the time gap is set to 2-2.5 seconds. At highway speeds this leads to a gap of greater than 60 meters or approximately 200 feet. At which point drafting benefits are greatly reduced.

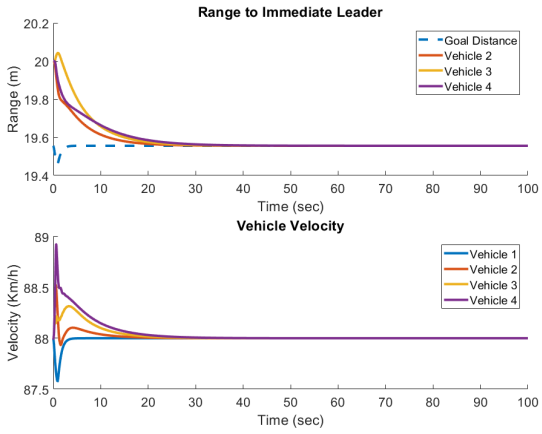
With the introduction of the inter-vehicle communication network, sufficient information, specifically the lead trucks acceleration, can be passed between the vehicles and a string stable controller with a constant spacing can be derived. Slotine and Li proved the feasibility of such a controller in [22]. As mentioned previously the lead vehicle's acceleration is utilized in the cascaded controller design used in this thesis, resulting in a string stable controller.

4.6 Simulation Evaluation

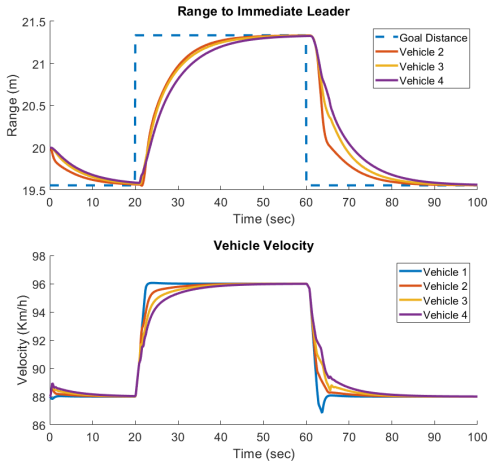
Before testing the convoy controller on the actual vehicle, it is important to test the control algorithms in simulation. Chapter 2 described the simulation environment developed in TruckSim and SIMULINK. This environment was used for all of the initial validation of the convoy controller, performed on platoons of four vehicles. This allows the validation of

many maneuvers that may not be easily tested in the real world (such as cut-ins) and the evaluation of string stability.

Figures 4.5a and 4.5b show the performance of the controller to both a small acquisition and step changes in desired spacing,respectively. This shows good performance and matches the design specification of an over-damped response. There are some slight variations in the control that are not smooth, these are due to gear shifts through the changing speeds.



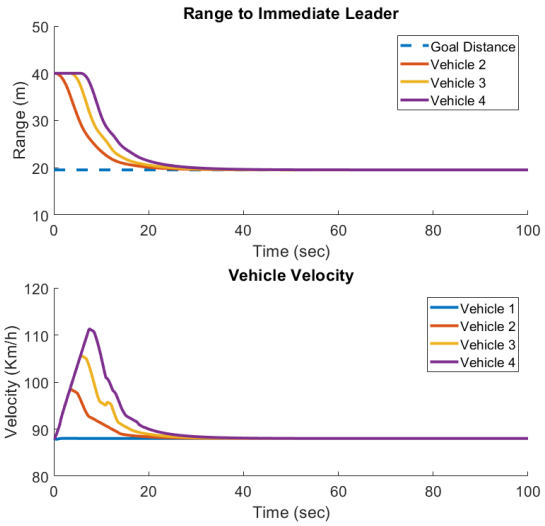
(a) 4 Vehicle Convoy Formation



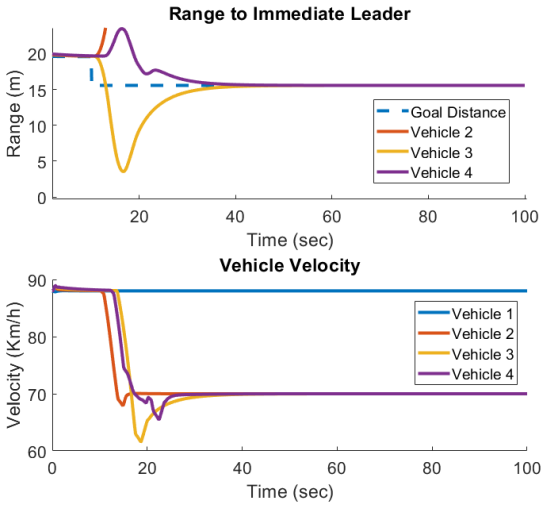
(b) Vehicle Convoy Step Changes

Figure 4.5: Acquisition Validation

The small gap convoy formation shown in the previous plots are unlikely real-world scenarios as it requires a human driver to close the gap to unsafe distances. Normally the controller would be engaged at much larger distances and close the gap in an automated state. This presents a potential issue with some control schemes in that the large control gap can cause integrator windup causing overshoot. It is important for the controller to exhibit an over-damped response as overshoot can bring the vehicle to an unsafe control distance.To prevent this the integral control scheme implements and anti-windup integration saturation that limits the total torque that can be applied due to integral errors. Figure 4.6a shows the formation of a convoy where each vehicle starts at a 40-meter gap and closes to the 20 meters desired spacing. The performance of each controller is similar to the results shown in the previous figures.



(a) Formation with Large Initial Gap



(b) Second Vehicle Brake Event

Figure 4.6: Controller Validation

It is also important to test the controller to sharp brake events. It can give both validation to the safety of the controller and the string stability of the convoy as a whole. The controller will exhibit string stability if the disturbance caused by a brake event diminishes as it travels back through the platoon. It can be seen from Figure 4.6b that the response from the 4th vehicle shows a better response to the brake event then vehicle 3. Thus the error dynamics diminish as the disturbance moves along the platoon.

The final test performed in simulation is to a changing reference, shown in Figure 4.7 . The controller still exhibits good performance with the whole convoy capable of tracking the reference. There are some minor disturbances in tracking due to lower-level brake performance. The cascaded controller handles these minor disturbances without significant deviation from the desired reference. There is some lag in the controller due to the limitations of the bandwidth of the lower-level controllers.

The controller performance is acceptable for the purposes of the work in this thesis. It shows string stable behavior and good tracking performance with no steady state error. To further validate the feasibility of CACC control algorithms real world testing and validation of the whole system is shown in Chapter 6. There is still optimization that can be performed

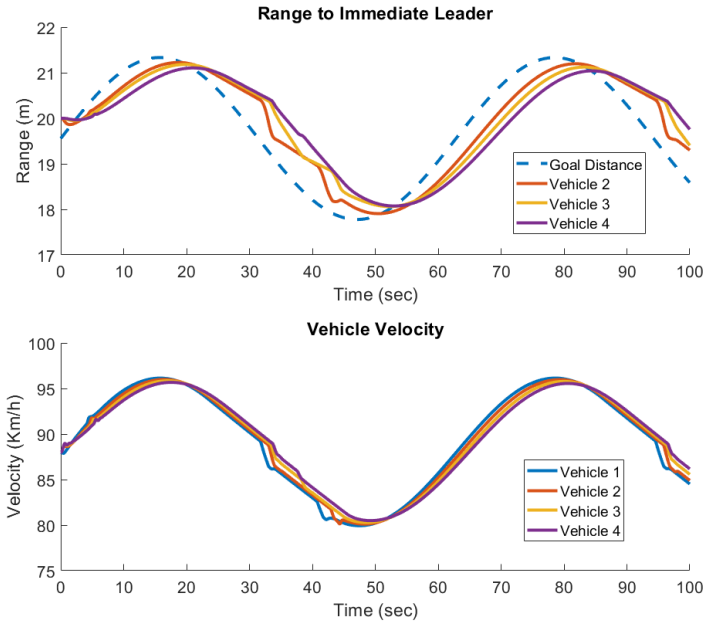


Figure 4.7: Reference Tracking

on the control algorithm to achieve better performance. However, that work will be left for future researchers. The primary contribution of this thesis is in the development and validation of the testing platform.

Chapter 5

Hardware and Software Design and Implementation

This chapter describes the system architecture designed to test and experimentally validate the algorithms previously discussed in this thesis. An in-depth look at J1939 CAN (Controller Area Network) standards and the Robot Operating System (ROS) software implementation required for this system is presented.

In order to validate the estimation and control schemes presented in Chapters 3 and 4 respectively and to demonstrate the practical feasibility, the CACC system has been implemented on two Peterbilt 579 heavy freight vehicles. These vehicles were selected because they are already equipped with ACC systems. The actuation components required for adaptive cruise control provide the foundation for the CACC system. Two primary components were added: DSRC radios for inter-vehicle communication and a GPS receiver. Vehicles equipped with ACC systems are becoming increasingly common and the addition of a GPS receiver and DSRC system is not cost prohibitive. ACC systems give access to lower-level control systems required to modulate the gap. This is because by-wire actuation of engine throttle, engine retarder torque, and foundation brakes is required for both ACC and CACC. Alternatively, by-wire capability could be added but would require additional hardware components designed to actuate these systems. In fact, many systems currently in use for vehicle automation validation attempt to act as "hands" and "feet. These systems seek to not make any changes to the CAN bus, but rely on other physical signals such as physically moving pedals designed to be used by a driver or intercepting and altering position sensor signals. These systems have their merits as the vehicle cannot tell the difference between a human driver and automated actuation, allowing all of the vehicles safety systems to behave exactly as designed. The work presented in this thesis focused strictly on using the ACC by-wire

actuation through the J1939 CAN bus. This method was chosen for two primary reasons: First, the physical modification of the vehicle is greatly reduced by using existing hardware, allowing the system to quickly be installed or removed. The second reason for relying on existing by-wire actuation is the fact that ultimately production-level systems will use this method of implementation. Thus, the system designed doesn't have to be concerned with unrealistic latency and complexities introduced by lower-level actuation. The difficulty in the by-wire actuation method is that digital CAN signals must be intercepted, recreated and rebroadcast. This can be challenging as many of these signals are proprietary and some level of vehicle "hacking" is required to spoof these signals.

Apart from the primary actuation systems, many software and hardware components were devolved to provide higher-level control signals to the lower-level systems. Each of these components will be discussed from both a hardware and software perspective.

Figure 5.1 shows a schematic representing the components and signals of communication between each of the elements used in the vehicle instrumentation process. The figure shows both the leading and following vehicles. While this system only shows the schematic relationship between two vehicles, it can easily be extended to much larger convoys.

Moving from simulations to real time implementation is always a significant step in the design process. It requires complete code to either be rewritten in an efficient runtime language or trans-coded using software tools such as MATLAB coder. While such a trans-coder would work for higher-level software packages such as controllers or estimators, it does not work for the lower-level software that must be written specifically for a given set of hardware. For this reason, all software was written from the ground up using an open-source middleware which has become very common on robotics platforms called Robot Operating System (ROS) . ROS allows individual code modules to easily communicate with each other which maximizes code re-usability. This also helps reduce the time required to change vehicle types and implement new controllers.

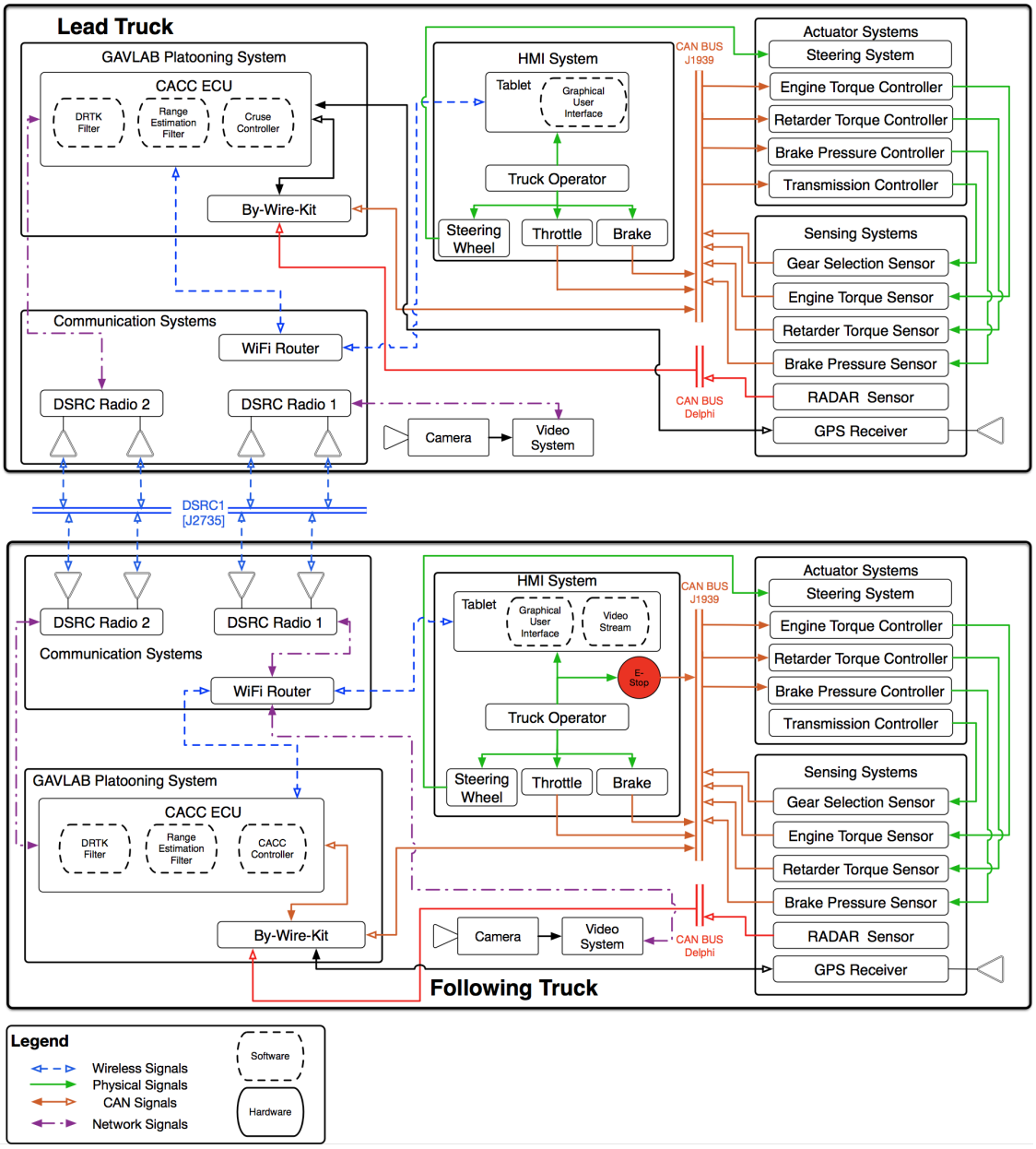


Figure 5.1: Schematic of Peterbilt Instrumentation

5.1 Hardware Implementation

As described above, the vehicle hardware additions required for convoying are minimal for a production vehicle; with the primary addition being the vehicle-to-vehicle communication(V2V),in this case DSRC, communication hardware. For the testing performed to date,

additional hardware was added to ensure better measurement accuracy and CPU bandwidth. A complete list of total hardware add-ons follows.

- Convoy PC (data logging and controller development)
- Delphi Radar (low-level range and range rate measurements)
- Novatel GPS Receiver (raw GPS carrier measurements for precise range measurements)
- Cohda Wireless DSRC (V2V communication)
- Hardware Enable/Disable Switch
- CAN disconnect system
- Emergency Stop Switch

5.1.1 Chassis Wiring

While the use of the existing ACC by-wire actuation limited the number of total modifications required on the vehicle, there were some additional wiring that had to be added to ensure proper routing of all of the required signals. Figure 5.2 shows a basic wiring diagram that was used to route required signals and power to the additional components listed above. Note this diagram does not show the connection of other components such as the DSRC and GPS.

5.1.2 CAN overview

Before discussing many specifics about the J1939 implementation, it is first necessary to give some background on CAN and the J1939 standards. CAN was invented by BOSCH in 1986 as a fault-tolerant communication architecture for connecting various systems on vehicles. The CAN bus provides a robust way for sensors, actuators and control systems to communicate with each other. One of the inherent safety features of CAN is the fact

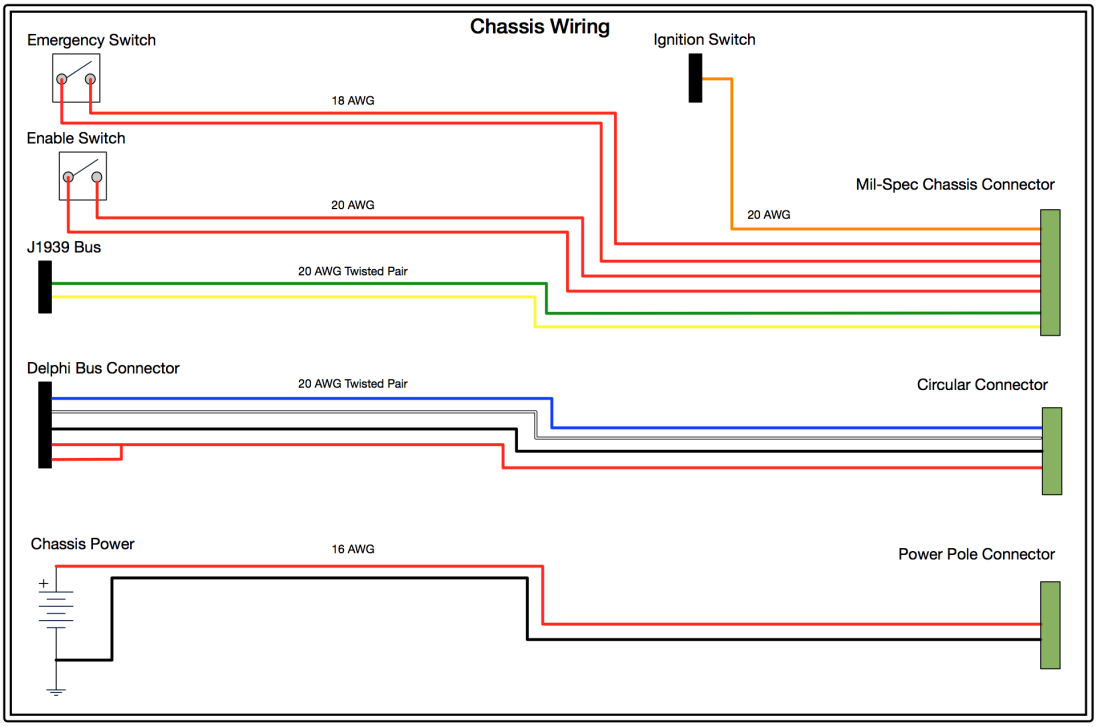


Figure 5.2: Schematic of Peterbilt Instrumentation

that no module sends out messages that "latch" , they are continuously streamed so that a break in communication or a crash in software cannot lock the system down to a specific stale command. The CAN implementation is simplified and broken into three parts (shown in Figure 5.3).

- The **Application Layer** is the component that interacts with the operating system and application of the CAN device.
- The **Data Link Layer** connects the actual data bytes to the protocol in terms of validating data, error checking, sending and receiving.
- The **Physical Layer** is the actual hardware along which the CAN signals propagate. i.e., physical electrical signals and wires connecting all of the devices on the bus.

Figure 5.3 presents these different layers and how they relate to each other. This is a somewhat simplified representation, as there are many other components to the protocol such as the network and transport layers. However, these are outside of the scope of this

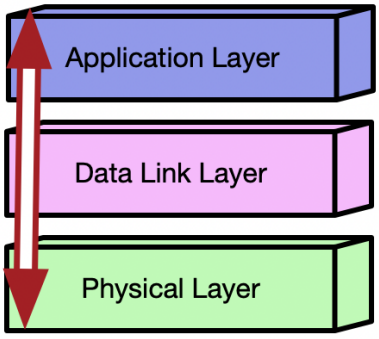


Figure 5.3: CAN Protocol Stack

discussion and played a minor role in the implementation. To simplify things even further, in most applications the data link layer and the physical layer can be ignored as they are already integrated into readily available silicon. Before moving on to the specifics of the integration at the application layer, it is worth noting some important aspects about the physical layer. The physical connection between all of the devices on the CAN bus is a simple twisted pair of wires carrying a differential signal. Secondly, these two wires extend between the beginning of the chain of connected devices and the end. This twisted pair is terminated at both ends by 120 Ω resistors. Note that each device along the chain pulls off a connection from these two wires and that it is recommended that these "legs" not be longer than six inches. This ensures the proper impedance on the line and reduce the number of repeated messages on the bus to ensure robust communication between devices.

When communicating with standard CAN devices for sending message packets on the bus such as the PCAN devices used in this implementation, the primary information required to send or receive messages on the bus is the message ID and the message data packet. Standard CAN uses and 11-bit ID; however, the J1939 standard uses a 29-bit ID. This ID is unique not only on the type of message sent but also on the sender's ID. This is one of the primary differentiators of J1939 from standard CAN which does not contain this information. This sender's ID is of singular importance when trying to replicate the communication process between devices. For instance, when trying to mimic the ACC system control signals sent

the first step was to determine the message IDs present on the bus with and without the ACC system.

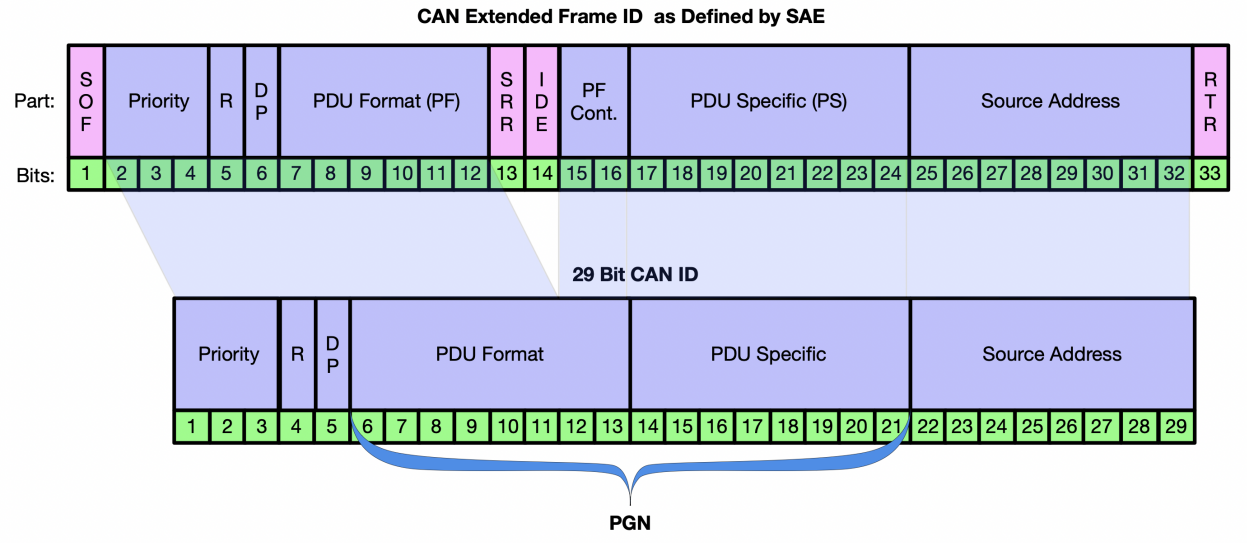


Figure 5.4: CAN ID Structure

Figure 5.4 shows the 33-bit definition of the 29-bit ID message as identified in the SAE standard as well as the abstracted version used when relying on lower-level silicon to handle the physical and data link layers. This is not specifically specified in the SAE standard as defined in SAE J1939-71 and can be difficult to understand. The SAE J1939-71 document does specify many measurements and commands that are used by most hardware components on heavy trucks. These messages are identified by two different IDs, the Parameter Group Number (PGN) and the Suspect Parameter Number (SPN). The PGN can be thought of as the actual message ID and is unique for every message type on the bus. The PGN, as shown in Figure 5.4, is composed of both the Protocol Data Unit (PDU) Format and the PDU Specific 8-bit IDs. The PDU Format and Specific will be discussed more in future. The SPN is a definition of a specific measurement type within the message packet for a given PGN. A PGN can be thought of as the message ID that describes a specific type of data packet, while this data packet may be broken up into many SPN types. As an example, PGN 61444 is a message describing some aspects (there are many messages published by the

engine ECU) of the electronic engine controller, and is often described by the abbreviation EEC1. The EEC1 message has a data packet associated with it that is eight bytes in length and is comprised of seven SPNs. The PGN describes the bit layout within the data packet. Each SPN is a specific measurement like Engine Speed. To determine how to parse the engine speed measurement it is necessary to use the SPN, 190 in this case, to determine the specific scaling and offset to take the two bytes and convert them into an engine speed. Table 5.1 shows an example PGN. Note that the PGN number 61444 is made by simply taking the PDU Format left shifting it by 8 bits and then performing a bitwise OR with the PDU Specific.

Table 5.1: PGN Example

PGN 61444 - Electronic Engine Controller 1 (EEC1)

Data Length	8 bytes	
PDU Format	240	
PDU Specific	4	
PGN	61444(0xF004)	
Data Length	SPN Description	SPN
4 bits	Engine Torque Mode	899
1 byte	Driver's Driver's Demand Engine - Percent Torque	512
1 byte	Driver's Actual Engine - Percent Torque	513
2 bytes	Driver's Engine Speed	190
1 byte	Driver's Source Address of Controlling Device for Engine Control	1483
4 bits	Driver's Engine Starter Mode	1675
1 byte	Driver's Engine Demand - Percent Torque	2432

Taking the above example further, the specific parsing required to read the engine speed (SPN 190) from the CAN bus can be seen in the following definition in Table 5.2.

To compete the example by making the 29-bit ID as described in Figure 5.4 it is first necessary to select a priority. Note the priority is simply a parameter that informs the data link layer as to the message's importance. Priority 0 is the maximum and 6 is the minimum. In practice, messages that are related to control are designated as priority 3 and most other measurements are designated priority 6. It should also be noted that if the CAN bus is not

Table 5.2: SPN Example

SPN 190 - Engine Speed

Data Length	2 bytes
Resolution	0.125 rpm/bit
Offset	0
Data Range	0 to 8,031.875 rpm
Type	Measured

near saturation the priority will have little effect on the message throughput. For the EEC1 message, the priority is normally 3. Next, the source address must be selected. Typically the EEC1 message is sent out by the engine as the name might suggest. The engine has a source address of 0. To combine the components into the message J1939 ID Equation (5.1) is used.

$$ID = (((Priority)\, OR\, <<\, 18)\, OR\, PGN)\, <<\, 8)\, OR\, SourceAddress \tag{5.1}$$

Equation (5.2) shows the solution from this example with logical ORs and bit shifts.

$$ID = (((0x3)\, OR\, <<\, 18)\, OR\, 0xF004)\, <<\, 8)\, OR\, 0x0 \tag{5.2}$$

Note the R and DP fields shown in 5.4 represent a reserved bit and a data page bit respectively. At the time of writing of this thesis, the Data Page (DP) data bit is zero as the total number of message types allowed in the current data page have not been exceeded. The reserved data bit is also zero. This is accounted for in the 18-bit left shift of the 3 priority bits.

The PDU Specific and PDU Format parts were of little use in this thesis except for in the startup segment of the system. To discuss this further it is important to note that there are two different types of PDU formats. Type 1 is used for special purposes such as a broadcast message which can send messages with large data packets over the bus. PDU Format type 2

is used for all other message types such as vehicle measurements. At vehicle startup there are several "call and response" messages sent out to all of the systems on the bus to determine the capabilities of the vehicle. During this time, "request" messages are sent out by the chassis controller. These requests must be responded to within a short time frame. Request and response messages are of a specific PDU format (234 and 232) and are described in detail in the network layer protocol description in J1939-21. In this section of the standard, the PDU Format type one is described in detail. When the PDU Format is type 1, the PDU Specific byte is no longer the group extension as with standard messages, but rather it is the destination address of a specific message. This allows two devices to send direct messages. In addition this protocol, the J1939-21 describes a portion of the protocol which allows for messages larger than eight bytes to be sent over the network by rapidly broadcasting multiple messages with the same PGN. This functionality is specifically covered by the Transport Protocol messages (PDF format 236). This message type can be quite complex and is not covered in this thesis other than to say that it is of specific importance when sending the startup messages required for the vehicle to accept both engine and brake commands.

5.1.3 Vehicle CAN Architecture

With the background of J1939 CAN discussed in the previous section, the specifics of the vehicle CAN architecture can be discussed in slightly more detail. Specifically, this section describes the process followed to mimic the ACC system messages. Experiments were designed to help determine the method of controlling both engine and braking systems used by the installed ACC system. Specific tests were performed that attempted to isolate various components to determine the state of the ACC controllers during engine control, retarder control, and foundation brake control. All of the raw CAN signals were recorded during these tests to find changing messages. Again, the specific architecture of J1939 makes the process easier as the source address field of each message allows sorting of specific messages by source. In the specific case of the two Peterbilt 579s source addresses, 42 and 160 were

noticed to only be present when the ACC system was installed. During the experiments mentioned above, the primary messages analyzed were from these source addresses.

Figure 5.5 shows the PGN 1035 message during an ACC braking experiment. In the image each of the plots shows one of the eight bytes contained in the message structure. It can be seen from the image that at time of 18 seconds multiple bytes begin to change. This coincides with the initiation of the brake event. Each byte was analyzed and it was determined that the signal structure matches that of the External Brake Rate (XBR) message defined in the SAE standard. The first and second bytes in this case are clearly the least significant and most significant bytes of the brake-rate signal. The specific settings of the XBR message were found from this plot.

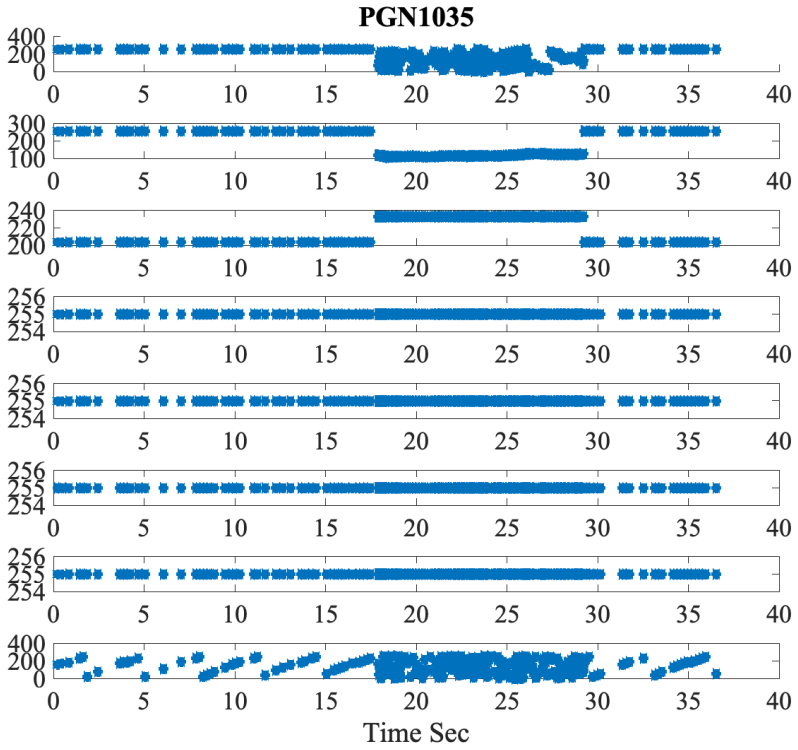


Figure 5.5: PGN 1035 During Brake Experiment

This process was followed for many messages including the engine torque request message and the retarder torque requests. In addition to these messages, multiple status messages are required to "spoof" the vehicle into thinking that the ACC system is operational. In the case of the ACC system, there is no harm in pretending to be this system to the vehicle as a whole. The ACC system is separate from other control functions and using the commands used by this system allows all other safety systems to behave normally. The full list of PGNs that is broadcast to spoof the vehicle is listed below. Their full structure and definitions can be found the source code for the J1939 ROS node.

- PGN 1035 (brake signal)
- PGN 0 (engine signal)
- PGN 15 (retarder signal)
- PGN 2560 (status signal)
- PGN 65226 (dashboard fault status signal)
- PGN 59904 (status signal)
- PGN 59647 (startup signal)
- PGN 61444 (EEC1 message indicating pedal position)
- PGN 61183 (address claim startup signal)
- PGN 61487 (advanced emergency braking system signal)
- PGN 60671 (beginning of broadcast message)
- PGN 60415 (data transmit of broadcast messages)
- PGN 65135 (ACC status message)
- PGN 65327 (status message)

These signals are sent out at specific rates to ensure the CACC controller maintains control of the truck. These signals do not override any signals on the CAN bus with the exception of PGN 61444. This signal is used by the transmission to determine when a shift should be performed. In order to get the driveline to engage, a pedal position must be written to the bus within that PGN. It is normally sent out by the engine control module. In this case the J1939 code writes to the bus at 100Hz to overwrite that signal. This allows the transmission to take control of the engine during shift operations. It should be noted that this is not an ideal solution as it requires abuse of the CAN bus to operate. It is, however, a non-invasive solution. In the future,the signal sent to the transmission should be separated from the main vehicle bus allowing the system to intercept these signals from the engine and populate them as required. Another solution would be to do a more thorough investigation of how shifting is performed during normal cruise control operation. These alternative methods could then be used to allow for smoother shifting operations.

5.1.4 RADAR

All ACC systems require a sensor for gathering the inter-vehicle spacing measurement. Typically these sensors are RADARs as they are the cheapest to produce. However, some systems do use LiDARs to measure the inter-vehicle spacing. The ACC systems installed on the two Peterbilt trucks use millimeter wave RADARs to provide measurements for range and range rate. For the purposes of this work the "stock" RADARs were removed as they have a proprietary CAN message structure that is not available to non-OEM parties. For this reason,the preinstalled RADARS on the vehicles were replaced with Delphi ESR RADARs. These RADARs are still production grade and do not have any specific features that make them superior, Delphi is simply willing to sell these units to 3rd parties and share their CAN message set.

The Delphi ESR is also a millimeter-wave RADAR that has two distinct channels, the first is a mid-range channel that has a wide-angle field of view of 45 degrees with a max

range of 60 meters. There is also a second long-range channel that has a 10 degree and a maximum range of 174 meters. Figure 5.6 shows a better view of the radiation pattern. These channels provide range measurements with an accuracy of 0.5 meters and a range rate measurement accuracy of 0.12 m/s. The RADAR returns 64 scan measurements , often referred to as object tracks. These measurements may merge or change tracking location from one scan to the next, making distinguishing relevant objects difficult. However, the ESR also provides functionality to feed vehicle state information such as speed, acceleration, steer angle and curvature measurements to the device. If these are properly formed the RADAR will then group objects into moving and stationary groups. In addition to this,the RADAR will provide one scan track that is determined to be "in-path. Allowing the RADAR to do this operation greatly reduces the complexity of resolving the relevant scan measurements. With the RADAR measurements reduced to a single relevant scan, the range validation and filtering becomes much easier to perform.

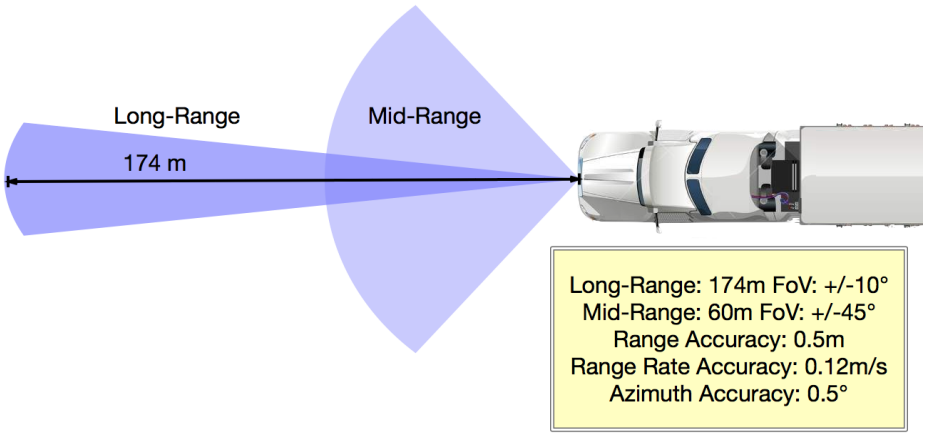


Figure 5.6: RADAR Radiation Pattern

5.1.5 DSRC

As mentioned in Chapter 4, one of the key components required for reducing inter-vehicle spacing is a reliable communication system. The FCC has set aside a band centered at 5.9GHz dedicated to communication networks for vehicles. These communications may

include vehicle to vehicle communication (V2V) as well as vehicle to infrastructure (V2I). CACC only requires a V2V communication network though it could potentially benefit from information from infrastructure such as road grade and speed limit information. For the work presented in this thesis, only a V2V communication was utilized.

The SAE has developed a standard for this communication protocol that is based on the IEEE:1609 wireless communication stack. Essentially this type of communication can be thought of as a Wi-Fi network between vehicles. In fact, the physical layer is merely a restructuring of the 802.11 standard on which Wi-Fi is based. This standard is called Dedicated Short-Range Communication (DSRC) and it has not yet reached high market penetration. SAE is still actively developing this standard and has defined a standard application layer with a specific message set used for V2V communication. These messages are defined in the SAE standard J2735. One message defined in this standard is the Basic Safty Message (BSM) that contains most of the vehicles state measurements required for platooning. It should be noted that the work in this thesis did not use the standard messages as defined in this standard primarily because the standard did not allow enough latitude to develop unique control algorithms which utilize additional information passed between vehicles. The standard was implemented in the early stages of this work to prove the feasibility of using such a standard, but after initial proof of concept testing the standard message set was put aside.

The final implementation is still the same at the physical layer which, again, is defined by the IEEE:1609 standard. The communication protocol is, therefore, subject to the same latency and packet-loss statistics that the SAE standard would experience. The implementation utilized in this work is a basic broadcast message and does not rebroadcast any lost packets. This is primarily done because if a message is dropped, by the time the message is rebroadcast and parsed the data contained is no longer considered "real time. Thus, it can be assumed that no stale messages are received. The messages do have a sequential counter and, therefore, if a message is received that is out of synchronization of the current sequential

count the message is removed and the count is reset. The communication is effectively a User Datagram Protocol (UDP) broadcasting scheme.

In this work two different radios were used to show the ability to pass the required communication packets between vehicles with various hardware sets. The two radios used are a Denso WSU 5001 and the Cohda Wireless MK5. They both showed similar characteristics while the Cohda Software Development Kit (SDK) proved to be slightly easier to use.

In addition to the radios themselves, there are also the DSRC antennas. The placement of these antennas proved to be of significant importance with respect to communication reliability. It was found that the best location to place the antennas is on both sides of the vehicle [23]. Mounts were designed for both sides of the vehicle as shown in Figure 5.7. The mount was devised to accommodate multiple antennas for future research purposes, though only one set of antennas on each side of the truck was used in this work.

Figure 5.7: Antenna Mount

5.1.6 GPS

The GPS system used is a dual frequency receiver that provide measurements on the L1 and L2 bands. The L1 band is used for both code based position tracking and carrier measurements. The L2 band is the Military band which is encrypted and cannot be used for anything other than carrier measurements in this application. This L2 band, however, is

critical for the DRTK algorithm outlined in Chapter 3 as it doubles the number of observations that can be used in the Kalman filter. Having both L1 and L2 carrier measurements make the resulting solution more robust. Novatel ProPAKs and PowerPAKs were used to supply these measurements. These receivers are somewhat expensive and are not commonly used on freight trucks today, however, lower-cost GPS sensors could be used without any significant degradation of system performance.

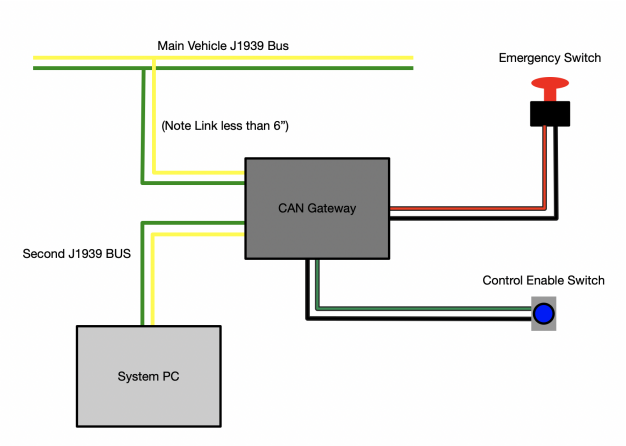
5.1.7 Safety Components

As any automation development requires many hours of testing, it is essential that safety be a primary consideration when designing a system for validation. The need for safety can be of even greater importance when testing vehicles with large gross vehicle weights as these vehicles are close to saturating their actuation ability without any emergency scenarios. For this reason, the system was designed with a two-layer approach to safety. There is a hardware-specific layer and a software layer. The hardware level safety systems are discussed in this section, while an overall system safety architecture is discussed later in the software safety section.

The primary hardware components for safety are an emergency switch and the CAN Gateway. The CAN gateway utilizes an MCP2562 CAN chip that allows the system to maintain two separate CAN buses. On one side is the vehicle's CAN bus and on the other is the CAN bus that connects to the CACC system PC. The MCP2562 chip allows the two buses to be bridged. During normal system operation, the CAN bus has the bridge mode enabled, this ensures that all messages are repeated on each bus. However when the E-stop switch is depressed, the CAN bus bridge pin is pulled to ground. When the bus bridge pin is grounded, it effectively detaches the system PC from the vehicle's bus allowing the vehicle to operate as if no modifications were made. No commands generated by the system PC can possibly reach the vehicle CAN bus during E-stop activation. This is done in hardware, and as such greatly reduces the chance of failure or system bugs. One additional function that

provides safety features to the system is the fact that when the buses are disconnected, a microcontroller is used to read from the vehicle bus and passes through these measurements to the system PC. This is done so that even with the E-stop button pressed, the System PC can still read the state of the vehicle and broadcast this state to other vehicles in the convoy, ensuring safety of the whole convoy. The hardware safety components are shown in Figure 5.8. When the E-stop button is depressed the vehicle is effectively back to its factory state. This allows completely normal operation of the vehicle when desired.

As mentioned previously, there are other hardware safety components to the system but they all rely on software at some level and as such are described in the software safety section 5.1.9 . The hardware components are considered a final layer of safety and are not expected to be used in normal operation. Normal activation and deactivation are handled in the software layer.



(a) Can Gateway Wiring

(b) Emergency switch placement

Figure 5.8: Hardware Safety Components

5.1.8 Software Architecture

The backbone of the system design is the sensor-driving software integration that ties all of the components together. To ensure the software can be used in all future implementations on the trucks, the software applications were written in a modular fashion. In order

to do facilitate the software modularity, a message passing structure is required to make sure separate algorithms are capable of passing variables to each other. To pass these messages between modules a middle ware software package called ROS (Robot Operating System) is used. In this framework, applications (or nodes as they are called in ROS) exist separately and run in their own thread. Nodes can start and stop independent of any other application. These nodes then pass pre-defined messages between themselves in an internal communication layer that mimics a client-server architecture. These messages are called topics and can be published in a synchronous or asynchronous manner, depending on the application.

In the case of this system, the software components can be broken down into three separate categories.

- Drivers
- Estimators
- Controllers

Drivers are software applications that communicate with hardware. These usually use an additional hardware communication layer such as serial, CAN, TCP/IP, or UDP to send and receive messages to the hardware components. Examples of this on the trucks are the RADAR, J1939 CAN and DSRC nodes. The communication application software type must be very robust as it is the foundation for the whole system. These drivers must be able to deliver consistent publishing rates and good error handling. Higher-level applications must be able to determine when one of these applications is not performing properly. ROS provides a framework for handling errors and ensuring consistent publishing rates. However, ROS cannot be run on a microcontroller and it is expected that many of the applications will eventually be implemented on much lower-level processors. For this reason, all of the software modules were written in an "agnostic" fashion, meaning that all of the functionality of the application was written as a stand-alone library independent of ROS. This library was then "wrapped" in ROS to ensure that the software developed can be used an almost any

application. Essentially this means that the software was designed, from the beginning, to make to code as portable and extensible as possible.

The estimation software detailed in Chapter 3, modules take measurements from various sensors and fuse them into more robust values that can be used for control. These estimators also filter measurements to remove noise form sensors before they can impact the control algorithm. The final task performed by the estimator software modules is the approximation of parameters that are not directly measurable. The grade estimator is an example of this form of approximation where the vehicles global velocity measurements are measured by the GPS system and the road grade can be approximated by taking the arc tangent of the vertical and horizontal velocities. It can be seen that these modules rely heavily on the software modules to deliver measurements. However, these nodes must also determine if the estimates provided are valid before passing them on to the controller.

The control nodes are the final major pieces of software that are used by the system. The current implementation only has two modules: the convoy control node and the convoy manager. The convoy control node is the real-time implementation of the controller designed in Chapter 4. This node takes measurements from the vehicles CAN bus, grade estimator, range estimator and DSRC communication node to calculate the required effective torque required to maintain a proper gap. This effective torque is then translated into engine torque, retarder torque and brake acceleration rates by the controller. To apply the correct actuation the control mode must be selected by the state machine mentioned in Chapter 4. In addition to the state measurement decision, the total controllers output effective torque is used to select the proper control mode. This ensures smoother transitions between control regimes. The convoy manager node's primary function is to determine the goal following distance and to decide if convoy control should be enabled. More about this process will be described in the software safety section below (Section 5.1.9).

The complete software architecture for the two-vehicle convoy is shown in Figure 5.9. The topics passing between the nodes represent the measurement, status or control messages

passed between the nodes. Some of the topics were combined or omitted for simplicity as there are close to 50 messages passed within the framework in total.

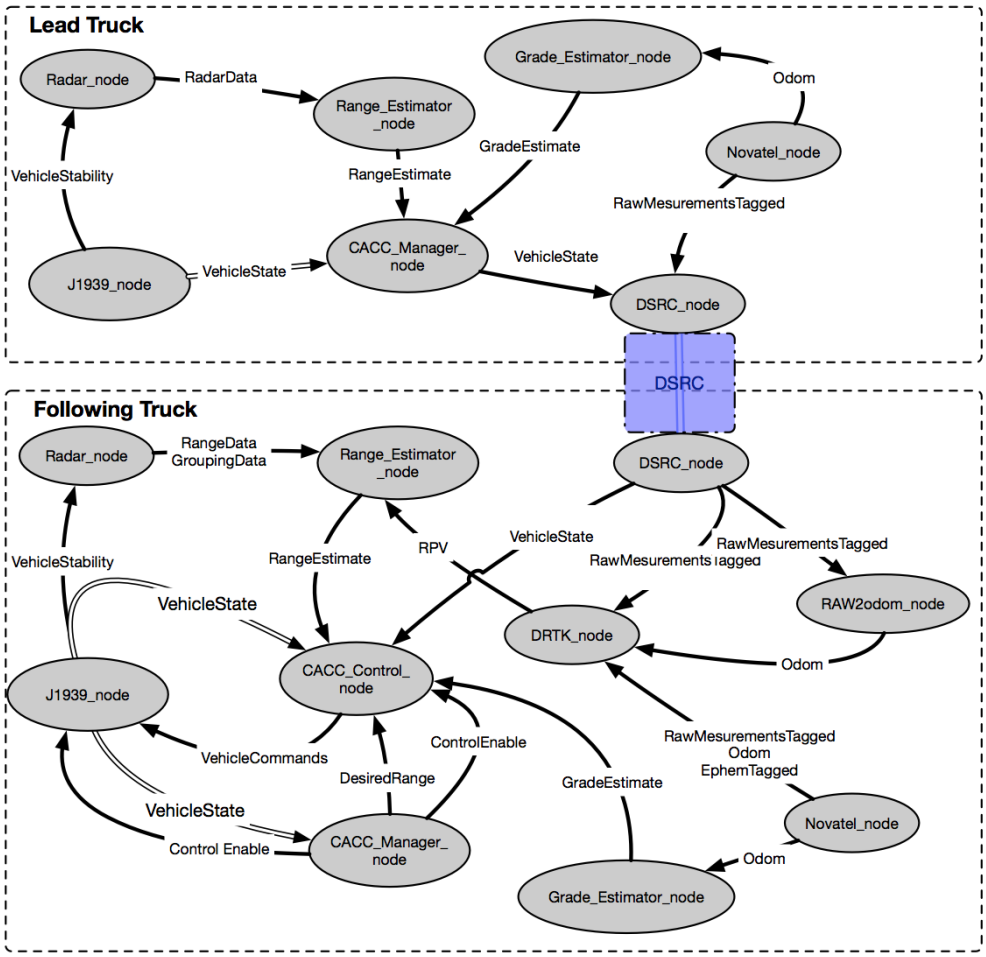


Figure 5.9: ROS Software Architecture

One final aspect of the system software architecture that is critical is the communication network between vehicles. The physical layer and application layer of the communication were described previously. However, there is an additional aspect that should be discussed, which is the method used to translate ROS messages into the UDP packets. The method implemented is a simple serialization of the ROS messages. The message is packed into a compact byte structure that can be passed over the network and translated on the other side. The benefit of this method is that there are no custom message definitions required. This also allows a lot of flexibility as there can be additional messages passed as control and estimation

algorithms change. These changes would not require and change in the communication code base. Figure 5.10 shows the DSRC communication messages used in the current system.

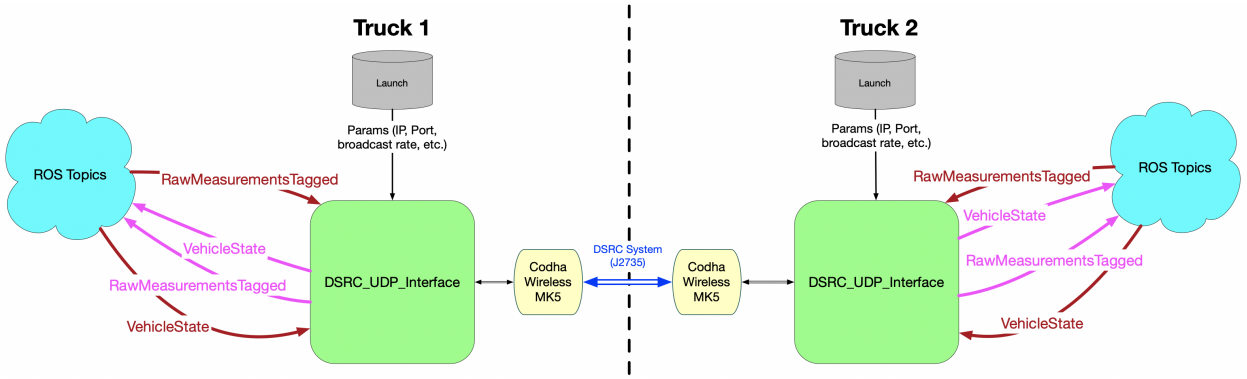


Figure 5.10: DSRC UDP Interface

5.1.9 Software Safety Consideration

As mentioned previously, in addition to the safety layer present in the hardware there is a higher level of safety that allows the driver or system to deactivate the controller within software. The methods of deactivation are similar to those of a standard cruise controller. Figure 5.11 shows the process for activating and deactivating the controller. There are several checks in the algorithm that ensure the vehicle is in a state that should allow control.

The control enable state is managed by the convoy manager node. The convoy manager determines if the required measurements are present, the vehicle communication network is linked to the leading vehicle, and that the vehicle is at a safe distance that allows for safe engagement. If these requirements are met, the convoy manager publishes a message enabling control. The J1939 CAN node alone is capable of activating the controller. The CAN node checks if cruise control button is enabled and "set" button is being depressed. If this is the case, the controller will activate.

To deactivate the controller, the driver can simply press the brake pedal or toggle the cruise enable button on the dash. Both of these steps will deactivate the controller. They are primarily hardware disable switches; however, their state is measured on the CAN

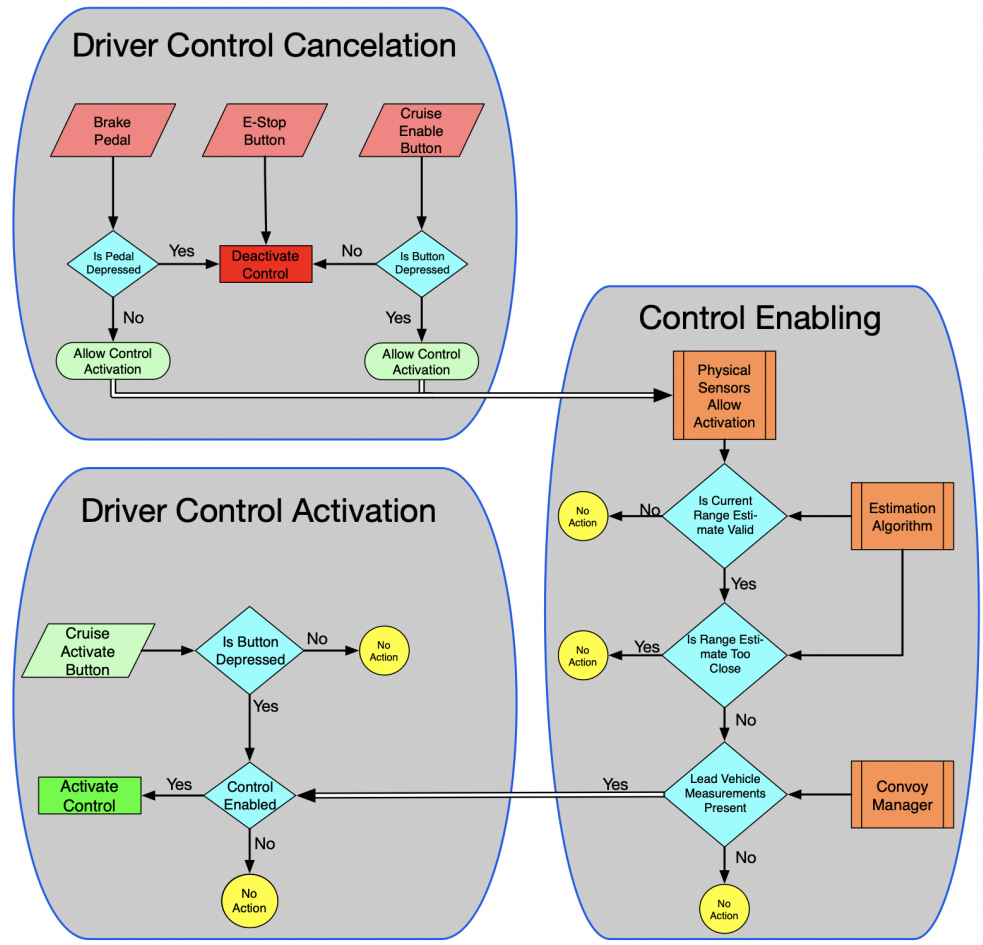


Figure 5.11: Software Activation and Deactivation

bus and, therefore, is considered a software safety layer. These actions are natural for a driver and can deactivate the controller within 100ms. There is a second level of CACC deactivation that is essentially a degradation into an ACC system. This occurs when the vehicle state measurements or GPS measurements are not present for the lead vehicle (three missed packets will engage this functionality). Effectively, this means there is a problem with the communication layer or the measurement process on the leading truck in either case CACC following distances are no longer safe. To prevent abrupt disengagement of the system when following at safe distances, the system simply slowly degrades to an ACC controller by gradually changing the desired gap to 200ft. This is implemented as a first order response with a large time constant that allows the vehicle to simply coast down to

a safe distance in most instances. The controller will not re-enable the CACC control gap until the measurements are again present and the driver reactivates the controller.

It should be noted that during the extensive testing of the controller, the software layer of activation and deactivation has always been sufficient. The E-stop switch has never been used. This is a good indication the stability and reliability of CAN communication.

Chapter 6

System Testing and Evaluation

The hardware and software described in the previous chapter required significant testing during the implementation process. Due to the complex interaction between all of the different hardware and software elements, each node had to go through extensive testing both in isolation and working in concert with other elements.

Much of this testing for control and estimation modules can be done by simply collecting recordings from sensor data and feeding them through the algorithms in post process. ROS has a tool that allows the collection of these recordings as well as allowing accurately timed playback, which allows for rapid code debugging and testing. However, most of the performance of the system has to be tested with the entire system running on the vehicles. The primary testing of the system components by themselves was performed at the National Center for Asphalt Technology (NCAT) test track skidpad. The entire system was mainly tested in three different locations. Initial testing was performed at Auburn Universities NCAT test track, the system was then tested in Michigan on interstate I-69 and finally on US Highway 280.

6.1 NCAT Testing

The NCAT test facility has a 1.7-mile oval test track. It is primarily flat and provides a good testing ground for steady state convoy maneuvers. Before the convoy was tested, a velocity control test was performed to ensure the inner loop can maintain a consistent velocity. Figure 6.1 shows a step response for a 1 m/s change in desired velocity. Two things are noticed from this plot that will influence the outer loop controller. First, there is significant noise in the velocity measurement coming from the wheel speed sensors on the

CAN bus. This can cause significant errors in the control scheme since a PID controller is used, which can cause the noise to be amplified by the derivative control gain. This may lead to excessive control modulation which could increase fuel consumption. A low-pass filter should have been used to filter the signal before its use in the controller. The second thing that is seen is the low bandwidth of the initial control gains. The inner loop is over-damped as expected, but the time constant is around 8 seconds. The bandwidth needed to be increased before effective gap control could be performed.

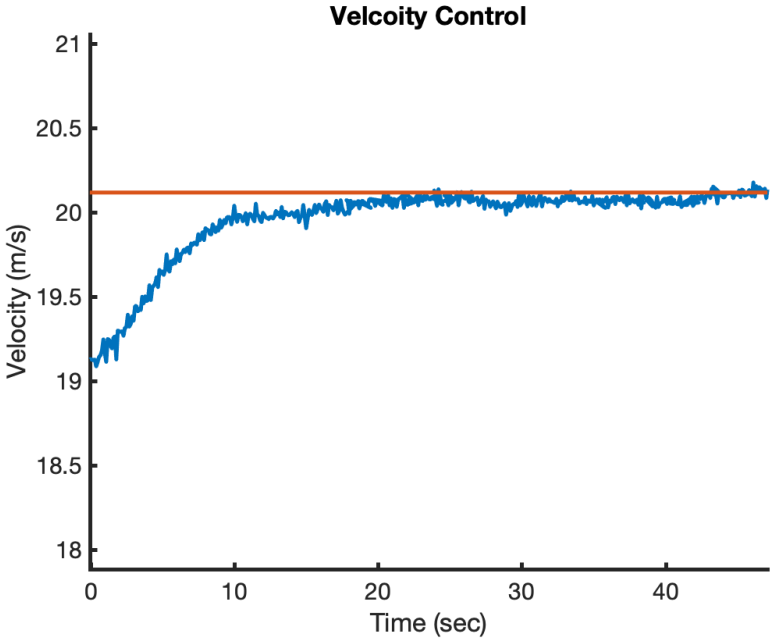


Figure 6.1: Velocity Control With Minor Step-Change

After slight alterations to the inner-loop gains, the convoy control was tested. Figure 6.2 shows the step response of the controller and the tracking performance of the convoy during speed changes. These velocity modulations were performed slowly over two laps around the track. It can be seen the headway error closes the gap well but sees significant oscillations about the set point. To examine this further, Figure 6.3 shows the velocity changes of the convoy. On a large scale, it appears as though the controller performs well and follows the leaders velocity. However, the lower plot shows the velocity error during the convoy. Again, the noise in the signal is particularly noticeable. In addition to this noise, it can be seen that

the controller maintains some velocity error which can be as much as 2 meters per second at times. Further control tuning is clearly required to ensure accurate tracking during changing velocities.

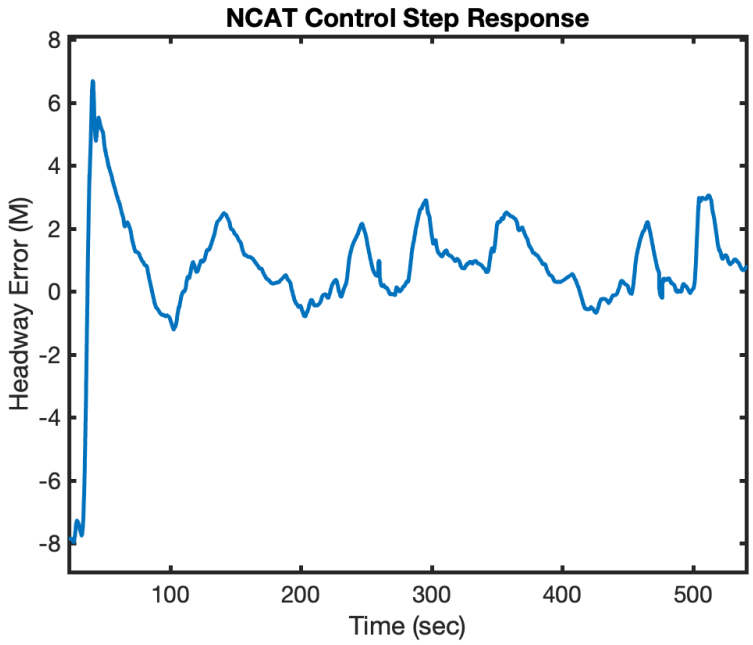


Figure 6.2: Following Vehicles Headway Error During Control (75 ft Gap)

During the tests at NCAT the DSRC latency was also assessed. By checking difference in the time-stamps of messages transmitted and received, the communication latency was obtained. These time-stamps were converted to GPS time as the CPU time would have an inherent bias between the two computers. The latency was found to be 10-18 milliseconds depending on the range between vehicles and the obstructions to line of sight. The largest latency was found in the turns at large gaps of 200 feet. The communication link was very stable and less than two percent of packets were lost during the tests. Finally, the total range where communication was possible was found to be slightly less than 600 meters. At this distance, however ,line of sight is critical. When the leading vehicle was not in view, the signal DSRC was lost.

While the tracking results do not show optimal performance of the convoy, they were deemed acceptable for the purposes of this work. Further tuning was performed during the

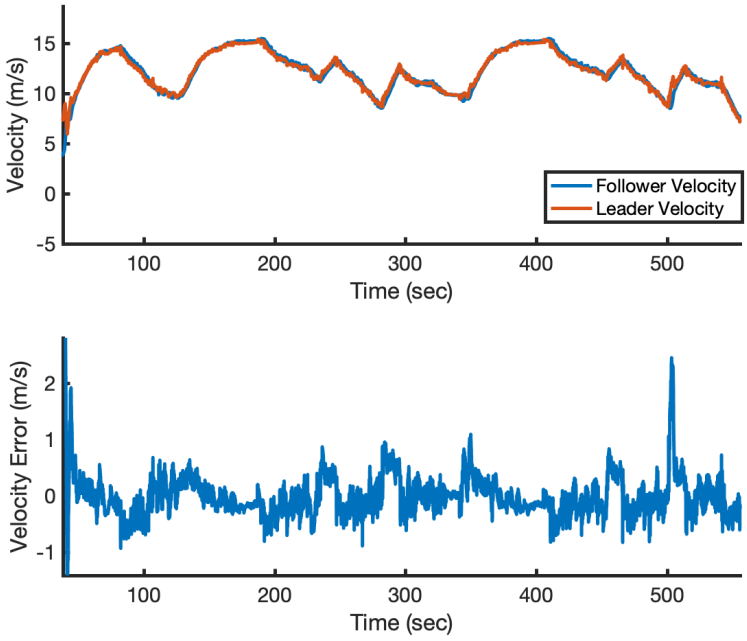


Figure 6.3: Speed Control During Convoy

other testing segments. The basic prototype of the system works and the safety, communication and sensor components performed well.

6.2 I-69 Testing

After initial testing at NCAT, the algorithms described in the previous sections, were run on public highways. The first set of public highway testing was conducted on I-69 in Michigan. The convoy was composed of four vehicles and two Peterbilt 579s tractors (A1 and A2) and two other freight liner vehicles(T13 and T14) with various payloads. The gross combined vehicle weights varied from 24000lbs to 54000lbs. Convoy spacing of 200 ft,100 ft, 75 ft and 50ft were tested. In total the controller was active for more than 170 miles over 4 days.

Initial fuel results show approximately 2-3% fuel savings for each truck on average. Some segments showed much better performance particularly flat segments which exhibited a 56% fuel savings. Figure shows a step response of a single truck being controlled to a 50ft

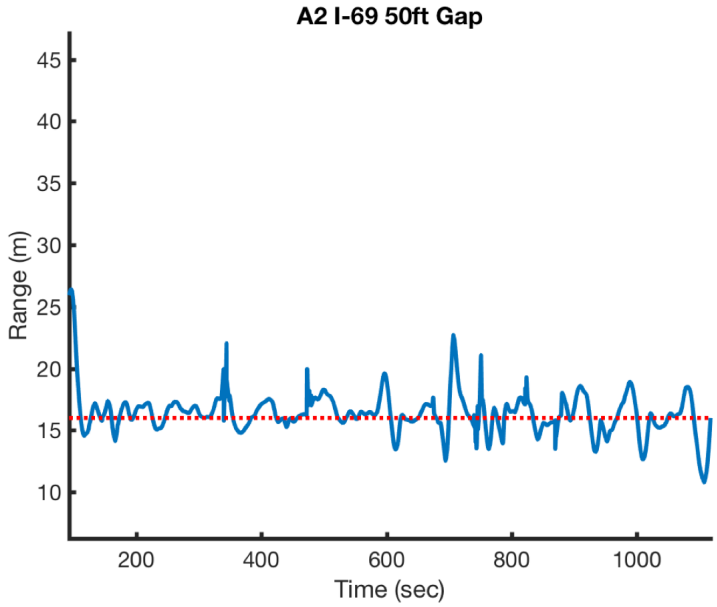


Figure 6.4: Following Vehicles Range to Leader During Control

gap. There is still a small amount of steady state error and 34 meters of control oscillation. This is due to estimation errors from bad GPS fixes. Figure 6.5 shows large changes in the estimated range in very short intervals. It was noted during testing that these changes normally occurred at an overpass where there was no longer clear sky visibility.

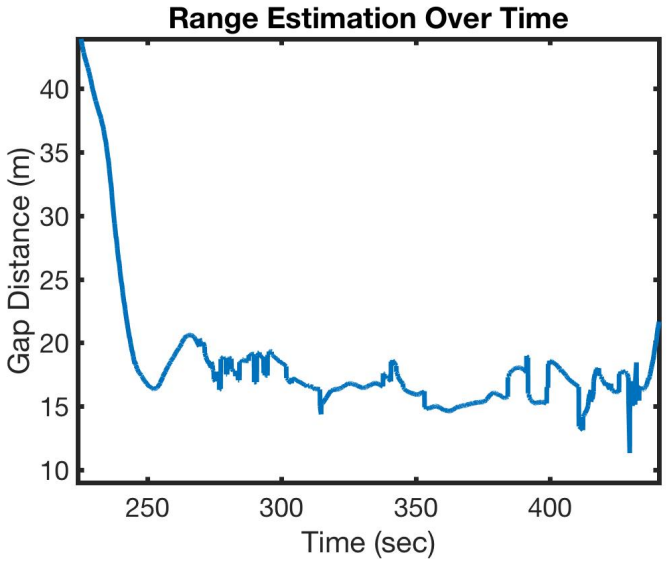


Figure 6.5: Large Changes in Range Estimate Due to GPS Outage

Figure 6.6 shows the fuel flow rate read from the vehicle J1939 CAN. The higher fuel consumption for T13 and T14 is due to the higher weight of the vehicle and the lower efficiency engine. It is important to note that the all trucks showed a minor improvement in fuel consumption relative to their own baseline but not relative to each other. This is simply due to the physical differences in the vehicles that make up the convoy. It is easy to see that the A2 truck is averaging slightly less fuel flow than the leading truck A1. A1 and A2 are very similar vehicles and it can be inferred that the reduction in fuel consumption is due to drafting effects.

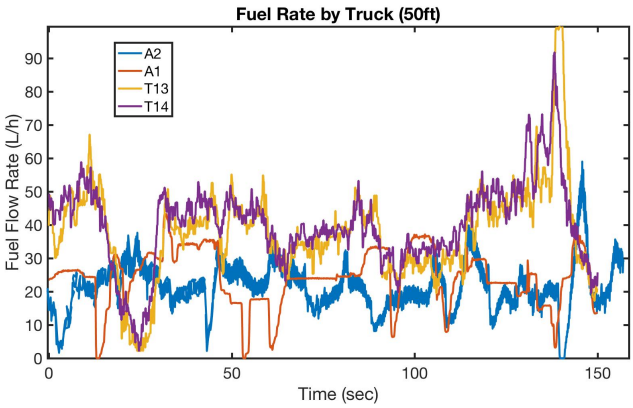


Figure 6.6: CAN Bus Fuel Rate for Each Vehicle During 4 Truck Convoy on I-69

6.3 Highway 280 Testing

After the I-69 tests in Michigan, the system was analyzed and a few modifications were made. The degraded performance on hills suggested modifications to the controller which were detailed in the feed-forward section in Chapter 4. The inclusion of a grade estimation scheme and the modifications to the controller helped the system maintain a more consistent gap during hilly sections. Additional modifications were also implemented in the estimation scheme to help the system be less reliant on the DRTK solution. The vehicle state information passed to the RADAR and the filtered solution returned from the unit allowed more reliable use of the RADAR ranging information. These alterations were then tested on US Highway 280. The corridor used for testing extends between Opelika, Alabama and

Columbus, Georgia. The East bound lanes provide good testing of hill tracking performance while the West bound lanes provide good flat terrain testing. Most testing was performed at 100 ft gap spacing as this distance allowed safe operation without convoy escorts.

Figure 6.7 shows the tracking performance of the controller on the East-bound lanes with hills. Figure 6.8 shows the tracking performance of the controller on the West-bound lanes with relatively flat terrain. Compared to the hilly terrain testing on interstate I-69, the controller oscillations and steady state errors show a much better performance. The controller oscillations on the hills were much improved, seeing on average .92 meters of gap error. The steady state error averaged 0.17 meters, in most cases this was actually a negative error, meaning that the vehicles were closer together than desired. Further investigation of the system is required to understand what alterations need to be made to eliminate this behavior.

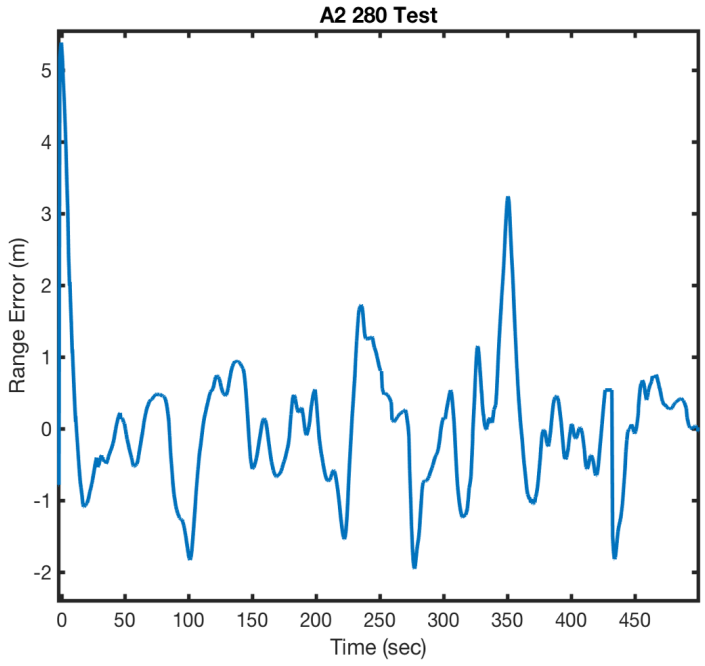


Figure 6.7: Following Vehicle Response and Tracking on Hills

In addition to control modifications, further testing of the RADAR tracking integration was performed on Highway 280. These tests were performed to validate the modifications

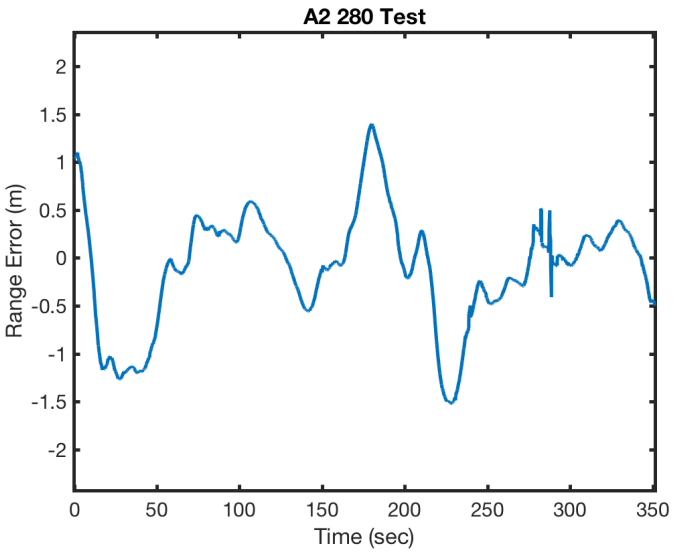


Figure 6.8: Following Vehicle Response and Tracking on Flats

to the RADAR sensor software which allowed the device to output its filtered solution as mentioned in Chapter 3. These software modifications showed a much better tracking rate with the leading vehicle. The vehicle maintained a lock on the "in-path" track for 86 percent of the time. It was noted that the tracking occasionally seemed to pick up passing vehicles in adjacent lanes. It is anticipated that better alignment of the RADAR could help with this. To achieve better RADAR measurements a more precise body alignment angle should be found. This can be difficult to find without a well-defined reference frame fixed to the truck body.

All of the testing to date has shown that the control and estimation schemes used in this thesis perform adequately. However, it is anticipated that with further testing and refinement of both control and estimation, better performance can be achieved. Further discussion of future work suggestions is presented in the conclusions chapter.

Chapter 7

Conclusions

The work outlined in this thesis is primarily a starting point for future research. The system developed proved the feasibility of a convoy control scheme for drag reduction and increased traffic throughput. However, it should be noted that further controller development and tuning is expected to yield better results. The system developed here provides a good foundation for this research.

7.1 System Performance

A basic fuel consumption analysis was performed on the data collected during testing. This analysis is somewhat limited and can only give a general overview of the current fuel consumption. The vehicle CAN fuel rate measurement is used to determine the average fuel savings for the following vehicle. This is a lower accuracy method of testing and should be validated using full SAE type-2 fuel testing in a controlled environment. On the other hand, though the measurement used for fuel consumption estimation is noisy and slightly biased it does give a better view of the types of real world numbers to be expected when including real-world disturbances such as traffic and changes in road grade.

The fuel results from testing showed a slight improvement in fuel consumption during drafting. As has been seen from other fuel testing at Auburn University, much higher fuel savings can be achieved [1]. Further development in control and estimation techniques should yield much better results. In addition to these alternate algorithms fuel testing should be performed to validate results in a formal manner. SAE type-2 fuel testing would provide a consistent basis on which the performance of the system could be analyzed.

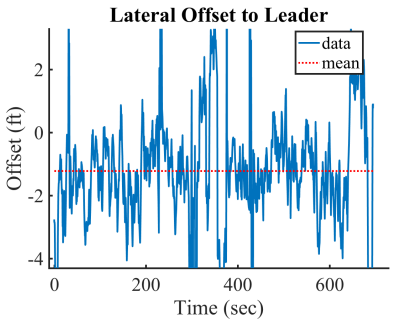
The work presented in this thesis has shown the development of a string stable convoy controller that is capable of regulating the gap with sub-meter accuracy. A fusion algorithm for combining RADAR and DRTK range measurements was also presented that provides robust measurements capable of handling GPS and RADAR outages. Again, the primary contribution presented in this work is the hardware and software system that can be used on any vehicle type as a simple bolt-on addition. This system has safety features integrated in from the highest to the lowest level.The foundational work for a fully automated platoon with both longitudinal and lateral control is in place. The system has been used for hundreds of hours of convoy operation over hundreds of miles of public highways without incident.

7.2 Future Work

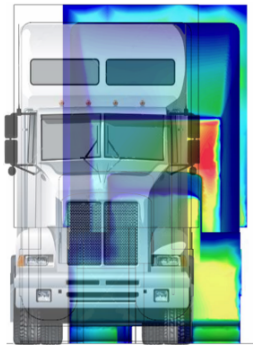
Before concluding this thesis, it is valuable to explain the future steps that should be taken to further the development of this system. Some of the lessons learned while developing the platform can help future researchers improve the systems reliability and performance.

- **Lateral Control** - During most of the testing it was easy to notice drivers of following trucks tended to slightly offset their vehicles in order to gain better visibility. This offset is marked and can be easily measured by using the DRTK RPV. Figure 7.1a shows a fairly consistent lateral offset of close to two feet. This offset causes the following vehicle to get out of the leading vehicle's wake which may diminish the overall fuel savings. Figure 7.1b shows a CFD analysis of what this exposure to the free stream air velocity could do to a following truck in terms of an increased pressure on the front face of the following vehicle.

  It is proposed that a lateral control system could be used in conjunction with the longitudinal controller to ensure following vehicles maintain good positioning within the lead vehicle's wake. Steering motors for this automation are already installed and integrated into the CAN control bus and should be ready for an upper-level controller



(a) Driver Lateral Offset



(b) CFD Analysis of Offset

Figure 7.1: Lateral Offset

wrapped around them. This addition could lead to better fuel consumption and lower driver fatigue.

- **Cut-in Detection** - One additional safety feature that could improve the systems performance is the addition of a cut-in detection scheme which will allow the convoy controller to react to traffic inserting itself into the platoon. In this case, the CACC convoy controller could gracefully degrade into a simple ACC controller, relying solely on the RADAR to estimate the range to the preceding cut-in vehicle. Again, once the cut-in estimation is complete the control system should be able to quickly accommodate a change in convoy spacing. A version of this changing convoy spacing functionality is already integrated in the system for use when vehicle communication is not available.

- **Integrated lower-level control board** - The next iteration on the hardware setup that should be implemented is a move from a higher-level control scheme in ROS to a lower-level implementation on microcontrollers. The primary functionality of the system has been proven and some of the simpler functionality like the J1939 integration, RADAR, and GPS/ DRTK measurements can be moved to a more hardened integration. This can decrease the total complexity of the ROS system and allow certain segments of the code base to be solidified. Effectively all of the vehicle specific components could be abstracted from the system and packaged in a solution that is

unique to each vehicle, allowing all higher lever estimation and control algorithms to be more confident about lower-level control and measurement components.

Bibliography

[1] H. I. Humphreys, "A Computational Fluid Dynamics Analysis of a Driver-Assistive Truck Platooning System with Lateral Offset," Ph.D. dissertation, Auburn Univeristy, 2017.

[2] F. H. Administration, "Freight Analsysis Framework," 2017.

[3] I. O. Standardization, "Intelligent transport systems – Adaptive cruise control systems – Performance requirements and test procedures," 2018.

[4] A. P. Ioannou and C. C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 657–672, 1988.

[5] D. Swaroop, "String Stability of Interconnected Systems - Automatic Control, IEEE Transactions on," vol. 41, no. 3, 1996.

[6] C.-Y. Liang and H. Peng, "Optimal Adaptive Cruise Control with Guaranteed String Stability," *Vehicle System Dynamics*, vol. 32, no. 4-5, pp. 313–330, 2003.

[7] C. Thorpe, T. Jochem, and D. Pomerleau, "The 1997 Automated Highway Free Agent Demonstration," 1998.

[8] F. Li and Y. Wang, "Cooperative Adaptive Cruise Control for String Stable Mixed Traffic: Benchmark and Human-Centered Design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3473–3485, 2017.

[9] J. Ploeg, B. T. M. Scheepers, E. V. Nunen, N. V. D. Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," *Conference on Intelligent Transportation Systems (ITSC)*, pp. 260–265, 2011.

[10] S. M. Martin, "Closely Coupled GPS/INS Relative Positioning For Automated Vehicle Convoys," Ph.D. dissertation, Auburn University, 2011.

[11] T. Sherer, "Radar Probabilistic Data Association Filter with GPS Aiding for Target Selection and Relative Position Determination," Ph.D. dissertation, Auburn Univeristy, 2017.

[12] Xiao-Yun Lu and J. Hedrick, "Longitudinal control design and experiment for heavy-duty trucks," 2003.

[13] X. Y. Lu and J. K. Hedrick, "Heavy-duty vehicle modelling and longitudinal control," *Vehicle System Dynamics*, 2005.

[14] X. Y. Lu, S. Shladover, and J. K. Hedrick, "Heavy-duty truck control: Short inter-vehicle distance following," in *Proceedings of the American Control Conference*, 2004.

[15] F. H. M. Da Rocha, V. Grassi, V. C. Guizilini, and F. Ramos, "Model Predictive Control of a Heavy-Duty Truck Based on Gaussian Process," *Proceedings - 13th Latin American Robotics Symposium and 4th Brazilian Symposium on Robotics, LARS/SBR 2016*, pp. 97–102, 2016.

[16] T. D. Gillespie, *Fundamentals of Vehicle Dynamics*. SAE International, 1992.

[17] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer, 2006.

[18] W. Travis, S. M. Martin, D. W. Hodo, and D. M. Bevly, "Non-Line-of-Sight Automated Vehicle Following Using a Dynamic Base RTK System," *NAVIGATION, Journal of the Institute of Navigation,*, vol. 58, no. 3, pp. 241 – 255, 2011.

[19] P. Misra and P. Enge, *GLOBAL POSITIONING SYSTEM Signals, Measure- ments, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.

[20] P. J. Teunissen, "The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation," *Journal of Geodesy*, vol. 70, no. 1-2, pp. 65–82, 1995.

[21] B. Ristic, S. Arumlampalam, and N. Gordon, *Beyond The Kalman Filter: Partical Filters for Trackin Applicaions*. Artech House, 2004.

[22] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, 1991.

[23] S. Gao, "Improving Vehicular Networking Reliability and Efficiency in the Context of Platooning Applications," Ph.D. dissertation, Auburn University, 2016.

Appendix

A.1 Heavy Vehicle Modeling

A.1.1 Newton Euler Equations of Motion

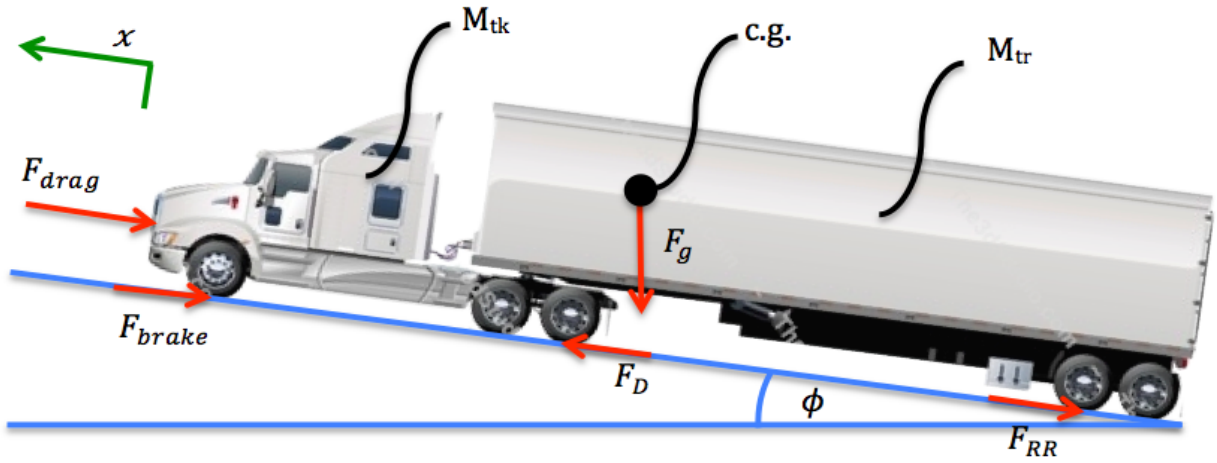


Figure A.2: Tractor Trailer Free Body Diagram

EOM 1:

$$\nwarrow \sum F_x = M_{total} * \ddot{x} = F_D - F_{brake} - F_{RR} - F_{drag} - M_{total} * g * sin(\phi) \tag{A.1}$$

$$M_{total} = M_{tk} + M_{tr} \tag{A.2}$$

Find drive force as a function of engine torque

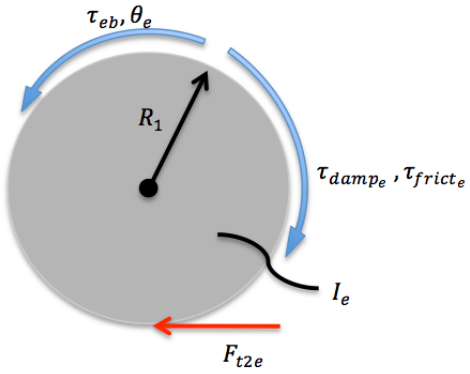


Figure A.3: Engine Flywheel Free Body Diagram

τ_{eb} Is the brake engine power produced. To be determined as a function of throttle position by using engine map.

EOM 2:

$$\circlearrowleft \sum M = I_e * \ddot{\theta}_e = \tau_{eb} - \cancel{\tau_{fricte}} - \tau_{damp_e} - F_{t2e} * R_1 \tag{A.3}$$

Note: engine friction is canceled here because it is assumed it is part of the mechanical efficiency of the engine.

Rewriting eq. A.3

$$\circlearrowleft \sum M = I_e * \ddot{\theta}_e = \tau_{eb} - b_e * \dot{\theta}_e - F_{t2e} * R_1 \tag{A.4}$$

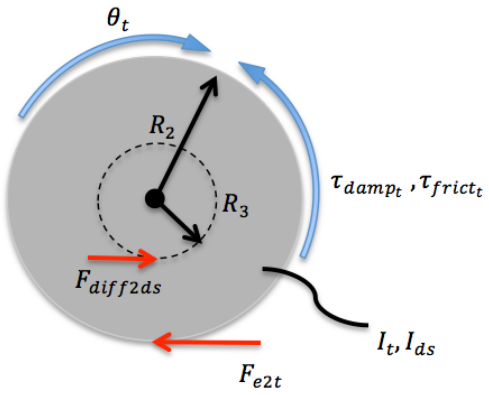


Figure A.4: Transmission Free Body Diagram

EOM 3:

$$\circlearrowleft \sum M = (I_t + I_{ds}) * \ddot{\theta}_t = -\tau_{frict_t} - \tau_{damp_t} + F_{e2t} * R_2 - F_{diff2ds} * R_3 \tag{A.5}$$

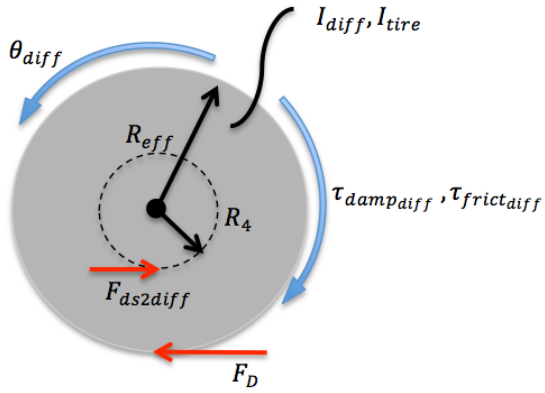


Figure A.5: Differential Free Body Diagram

EOM 4:

$$\circlearrowleft \sum M = (I_{diff} + I_{tire}) * \ddot{\theta}_{diff} = -\tau_{frict_{diff}} - \tau_{damp_{diff}} + F_{ds2diff} * R_4 - F_D * R_{eff} \quad \text{(A.6)}$$

A.1.2 Relating Equations of Motion

Fig. A.6 shows the linkages between the free body diagrams shown above. The driveshaft and drive axle are assumed to have no compliance i.e. $\theta_t = \theta_{ds}$ and $\theta_{diff} = \theta_{tire}$

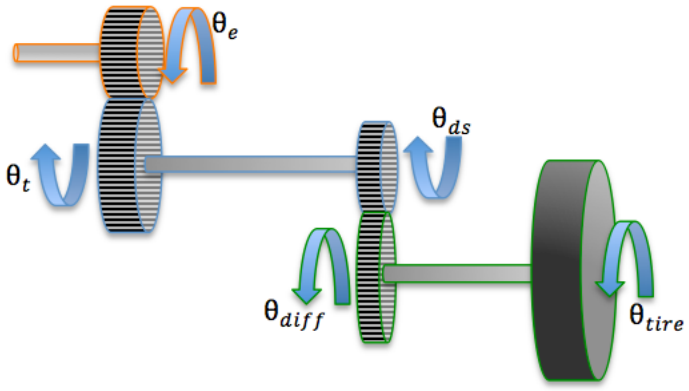


Figure A.6: Spin-Rate Diagram

$$N_T = \frac{R_2}{R_1} \quad \text{(A.7)}$$

$$N_{diff} = \frac{R_4}{R_3} \quad \text{(A.8)}$$

Assuming no tire slip the following relationships can be found between the spin rates of the drivetrain and the longitudinal speeds of the tractor trailer.

$$\theta_{diff} = \frac{x}{R_{eff}} \quad \text{(A.9)} \qquad \dot{\theta}_{diff} = \frac{\dot{x}}{R_{eff}} \quad \text{(A.10)} \qquad \ddot{\theta}_{diff} = \frac{\ddot{x}}{R_{eff}} \quad \text{(A.11)}$$

$$\theta_t = \frac{x * N_{diff}}{R_{eff}} \quad \text{(A.12)} \qquad \dot{\theta}_t = \frac{\dot{x} * N_{diff}}{R_{eff}} \quad \text{(A.13)} \qquad \ddot{\theta}_t = \frac{\ddot{x} * N_{diff}}{R_{eff}} \quad \text{(A.14)}$$

$$\theta_e = \frac{x * N_{diff} * N_T}{R_{eff}} \quad \text{(A.15)} \qquad \dot{\theta}_e = \frac{\dot{x} * N_{diff} * N_T}{R_{eff}} \quad \text{(A.16)} \qquad \ddot{\theta}_e = \frac{\ddot{x} * N_{diff} * N_T}{R_{eff}} \quad \text{(A.17)}$$

A.1.3 Solving for the Relationship between $\ddot{x}$ and τ_{eb}

Rearranging Eq. A.3 and substituting in Eqs. A.15 and A.16 to solve for F_{t2e}. Note all frictions in EOMs 2-4 are assumed to be accumulated in the F_{RR} term in EOM 1.

$$I_e * \frac{\ddot{x} * N_{diff} * N_T}{R_{eff}} = \tau_{eb} - b_e * \frac{\dot{x} * N_{diff} * N_T}{R_{eff}} - F_{t2e} * R_1$$

$$F_{t2e} = \frac{\tau_{eb} - I_e * \frac{\ddot{x} * N_{diff} * N_T}{R_{eff}} - b_e * \frac{\dot{x} * N_{diff} * N_T}{R_{eff}}}{R_1} \tag{A.18}$$

Rearrange Eq. A.5 and substitute in Eqs. A.12 and A.13 to solve for $F_{diff2ds}$.

$$(I_t + I_{ds}) * \frac{\ddot{x} * N_{diff}}{R_{eff}} = -b_t * \frac{\dot{x} * N_{diff}}{R_{eff}} + F_{e2t} * R_2 - F_{diff2ds} * R_3$$

$$F_{diff2ds} = \frac{-(I_t + I_{ds}) * \frac{\ddot{x} * N_{diff}}{R_{eff}} - b_t * \frac{\dot{x} * N_{diff}}{R_{eff}} + F_{e2t} * R_2}{R_3}$$

Now plug in Eqs. A.18 and A.7

$$F_{diff2ds} = \frac{-(I_t + I_{ds}) * \frac{\ddot{x} * N_{diff}}{R_{eff}} - b_t * \frac{\dot{x} * N_{diff}}{R_{eff}} + (N_T * \tau_{eb} - I_e * \frac{\ddot{x} * N_{diff} * N_T^2}{R_{eff}} - b_e * \frac{\dot{x} * N_{diff} * N_T^2}{R_{eff}})}{R_3}$$

Combine terms

$$F_{diff2ds} = \frac{-(I_t + I_{ds} + I_e * {N_T}^2) * \frac{\ddot{x} * N_{diff}}{R_{eff}} - (b_t + b_e * {N_T}^2) * \frac{\dot{x} * N_{diff}}{R_{eff}} + N_T * \tau_{eb}}{R_3} \tag{A.19}$$

Rearrange Eq. A.6 and substitute in Eqs. A.9 and A.10 to solve for F_D.

$$(I_{diff} + I_{tire}) * \frac{\ddot{x}}{R_{eff}} = -b_{diff} * \frac{\dot{x}}{R_{eff}} + F_{ds2diff} * R_4 - F_D * R_{eff}$$

$$F_D * R_{eff} = -(I_{diff} + I_{tire}) * \frac{\ddot{x}}{R_{eff}} - b_{diff} * \frac{\dot{x}}{R_{eff}} + F_{ds2diff} * R_4$$

Plug in Eqs. A.19 and A.8

$$F_D = \frac{-(I_{diff} + I_{tire}) * \frac{\ddot{x}}{R_{eff}} - b_{diff} * \frac{\dot{x}}{R_{eff}} + N_{diff} * (-(I_t + I_{ds} + I_e * {N_T}^2) * \frac{\ddot{x} * N_{diff}}{R_{eff}} - (b_t + b_e * {N_T}^2) * \frac{\dot{x} * N_{diff}}{R_{eff}} + N_T * \tau_{eb})}{R_{eff}}$$

Combine terms

$$F_D = -((I_t + I_{ds} + I_e * {N_T}^2) * {N_{diff}}^2 + (I_{diff} + I_{tire})) * \frac{\ddot{x}}{{R_{eff}}^2} - ((b_t + b_e * {N_T}^2) * {N_{diff}}^2 + b_{diff}) * \frac{\dot{x}}{{R_{eff}}^2} + \frac{N_{diff} N_T * \tau_{eb}}{R_{eff}} \quad \text{(A.20)}$$

Rearrange Eq. A.1 and plug in Eq. A.20

$$[M_{total} + \frac{((I_t + I_{ds} + I_e * {N_T}^2) * {N_{diff}}^2 + (I_{diff} + I_{tire}))}{{R_{eff}}^2}] * \ddot{x} + (\frac{(b_t + b_e * {N_T}^2) * {N_{diff}}^2 + b_{diff}}{{R_{eff}}^2} + C_D * \rho * A * \dot{x}) * \dot{x} = \frac{N_{diff} N_T * \tau_{eb}}{R_{eff}} - F_{brake} - F_{RR} - M_{total} * g * sin(\phi) \quad \text{(A.21)}$$

Linearize about set velocity V_0

$$[M_{total} + \frac{((I_t + I_{ds} + I_e * {N_T}^2) * {N_{diff}}^2 + (I_{diff} + I_{tire}))}{{R_{eff}}^2}] * \ddot{x} + (\frac{(b_t + b_e * {N_T}^2) * {N_{diff}}^2 + b_{diff}}{{R_{eff}}^2} + C_D * \rho * A * V_0) * \dot{x} = \frac{N_{diff} N_T * \tau_{eb}}{R_{eff}} - F_{brake} - F_{RR} + \frac{C_D * \rho * A * {V_0}^2}{2} - M_{total} * g * sin(\phi) \quad \text{(A.22)}$$

A.1.4 Find Transfer Function $\frac{x(s)}{\tau(s)}$

Removing all other inputs from Eq. A.22

$$[M_{total}+\frac{((I_t+I_{ds}+I_e*{N_T}^2)*{N_{diff}}^2+(I_{diff}+I_{tire}))}{{R_{eff}}^2}]*\ddot{x}+(\frac{(b_t+b_e*{N_T}^2)*{N_{diff}}^2+b_{diff}}{{R_{eff}}^2}+C_D*\rho*A*V_0)*\dot{x}=\frac{N_{diff}N_T*\tau_{eb}}{R_{eff}}$$

Taking the Laplace of both sides

$$\left[[M_{total}+\frac{((I_t+I_{ds}+I_e*{N_T}^2)*{N_{diff}}^2+(I_{diff}+I_{tire}))}{{R_{eff}}^2}]*s^2+(\frac{(b_t+b_e*{N_T}^2)*{N_{diff}}^2+b_{diff}}{{R_{eff}}^2}+C_D*\rho*A*V_0)*s\right]*x(s)=\frac{N_{diff}N_T*\tau_{eb}}{R_{eff}}$$

$$\frac{x(s)}{\tau(s)}=\frac{\frac{N_{diff}N_T}{R_{eff}}}{[M_{total}+\frac{((I_t+I_{ds}+I_e*{N_T}^2)*{N_{diff}}^2+(I_{diff}+I_{tire}))}{{R_{eff}}^2}]*s^2+(\frac{(b_t+b_e*{N_T}^2)*{N_{diff}}^2+b_{diff}}{{R_{eff}}^2}+C_D*\rho*A*V_0)*s} \tag{A.23}$$