

2019 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY
SYMPOSIUM

AUTONOMOUS GROUND SYSTEMS TECHNICAL SESSION
AUGUST 13-15, 2019 - NOVI, MICHIGAN

Cooperative Adaptive Cruise Control (CACC) in Controlled and Real-World Environments: Testing and Results

Jacob Ward¹, Patrick Smith¹, Dan Pierce¹, David Bevly, PhD¹, Paul Richardson, PhD², Sridhar Lakshmanan, PhD², Athanasios Argyris², Brandon Smyth², Cristian Adam², Scott Heim³

¹Mechanical Engineering, Auburn University, Auburn, AL

²Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, MI

³US Army Combat Capabilities Development Command, Ground Vehicles System Center

ABSTRACT

The transportation industry annually travels more than 6 times as many miles as passenger vehicles [1]. The fuel cost associated with this represents 38% of the total marginal operating cost for this industry [8]. As a result, industry's interest in applications of autonomy have grown. One application of this technology is Cooperative Adaptive Cruise Control (CACC) using Dedicated Short-Range Communications (DSRC). Auburn University outfitted four class 8 vehicles, two Peterbilt 579's and two M915's, with a basic hardware suite, and software library to enable level 1 autonomy. These algorithms were tested in controlled environments, such as the American Center for Mobility (ACM), and on public roads, such as highway 280 in Alabama, and Interstates 275/696 in Michigan. This paper reviews the results of these real-world tests and discusses the anomalies and failures that occurred during testing.

Citation: Jacob Ward, Patrick Smith, Dan Pierce, David Bevly, Paul Richardson, Sridhar Lakshmanan, Athanasios Argyris, Brandon Smyth, Cristian Adam, Scott Heim "Cooperative Adaptive Cruise Control (CACC) in Controlled and Real-World Environments: Testing and Results", In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 13-15, 2019.

1. INTRODUCTION

The average class 8 vehicle annually travels 6 times as many miles as the average passenger vehicle [1]. The disparity between class 8 vehicles and passenger cars is far greater in terms of annual

fuel consumption, where the average class 8 vehicle consumes approximately 26 times as much fuel as a passenger vehicle as shown in Figure 1 [2].

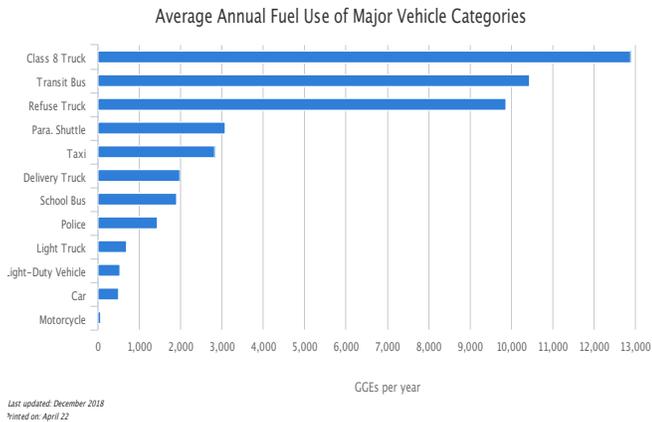


Figure 1. Average Annual Fuel Use of Major Vehicle Types

These statistics reflect the large interest in attempting to reduce the fuel consumption of these vehicles. One of the most promising fuel saving techniques is vehicle platooning, where vehicles travel at close distances to each other resulting in substantial drag reduction. Therefore, a longitudinal controller, which allows control over forward motion of a vehicle, is needed to safely realize these drag savings at close following distances.

Auburn University developed a hardware and software suite to allow autonomous longitudinal control capabilities on two Peterbilt 579’s and two military Freightliner M915’s. The software development was done in both C++ and Python and is implemented with the Robotic Operating Software (ROS). ROS allows various software functions and libraries to exchange variables and data easily. The resulting system allows for multiple trucks, or a platoon, to maintain a constant spacing from each other at speed.

The majority of the development was performed in controlled test environments. This ensured the safety of the vehicle operators, engineers and other motorists when the system was in heavy development. However, in October 2018, a four-truck platoon was taken onto interstates around Detroit and was tested in high traffic environments.

Even in the midst of extremely heavy traffic, where the four-truck platoon was required to change lanes, the system was able to maintain a

constant gap between the vehicles. The longitudinal control system only had to be disengaged for two scenarios: when multiple vehicles would cut-in between the platoon and when cars would merge into the lane. At this point, drivers would take over and catch up with the platoon.

While this testing provided validation for real-world applications of this system, it was not without lessons learned. The rest of this paper aims to provide a high-level introduction to the Auburn University system and address those lessons learned in greater detail.

2. Hardware

To enable automation on the class 8 vehicles, several key pieces of hardware were added to the Peterbilts and M915’s. The main hardware additions were Dedicated Short Range Communication (DSRC) radios, a front facing Delphi radar, a Novatel GPS receiver, and a Nuvo computer. These hardware components enable the longitudinal control of the class 8 vehicles and are explored in more depth below.

2.1. DSRC Communication

Vehicle to Vehicle (V2V) communication is accomplished via a 5.9 GHz DSRC radio network. The specific radios used on the Auburn University vehicles are Codha Wireless MK5 On Board Units (OBU’s). The radio network is used to transmit a custom data packet that contains information necessary for the navigation and control algorithms. Specifically, the message payload is comprised of two different types: the vehicle state and raw GPS data. The vehicle state includes information such as the vehicle’s acceleration, velocity, and brake status. The raw GPS data includes the GPS pseudorange and carrier phase measurement, which are used to produce a range estimate. In addition to sharing information amongst vehicles, V2V communication also addresses stability concerns as platoon size grows [3].

The University of Michigan-Dearborn (UM-D) is a partner university that primarily handles the radio communications and network diagnostics of this platooning work. UM-D developed a software application that runs on the Cohda MK5 OBU and allows for V2V communication.

The V2V communication process starts when control information is passed to the operational radio network through a User Datagram Protocol (UDP) socket. This information is then assembled into a radio packet and transmitted to other radios via DSRC. The receiving radio transforms the data back into a UDP packet and forwards the packet to the Auburn University CACC system. Figure 2 shows an overview of the Cohda radio system and how it transmits data.

Specifically, the vehicle state message is transmitted at 20 Hz and the raw GPS information is updated at 2 Hz. Once a GPS measurement is available, it is appended to a vehicle state message and transmitted as a single packet. As a UDP packet, the vehicle state message is approximately 200 bytes and the raw GPS message is approximately 150 bytes, plus an additional 100 bytes per satellite in view. As a result, the packet size is variable based on the measurement epoch time and the number of satellites observed.

The UM-D software is multi-threaded, having threads for transmitting, receiving, statistics and UDP communication. UDP sockets are incorporated for passing the data to/from the operational radio network, allowing for a flexible interface between host and network. The software also incorporates the ability to monitor real-time radio network health.

On reception of a DSRC message, the UM-D software generates debug information on a separate port. The message latency is calculated by differencing the lead GPS time, which is embedded in each packet header, with the current GPS time of the following vehicle.

The definition of latency used in this paper is the measure of the delay between a message being transmitted by the lead vehicle, and the receive time of the following radio. Latency is an important measure of the network health, and the ability for a truck to CACC platoon. For example, if the latency is too high, the measurement cannot be used because the information is no longer valid.

2.2. RADAR

A Delphi Electronically Scanning RADAR (ESR) provides a high frequency range, range rate and bearing measurement. These measurements are used not only to help detect cut-in vehicles, but also to augment the update rate of the “range”, ie. the longitudinal component of the relative position vector (RPV). Figure 3 shows a visual example of what the RADAR returns while in light highway traffic, with boxes indicating RADAR tracks, or RADAR signatures that are determined to be in the path of the vehicle.

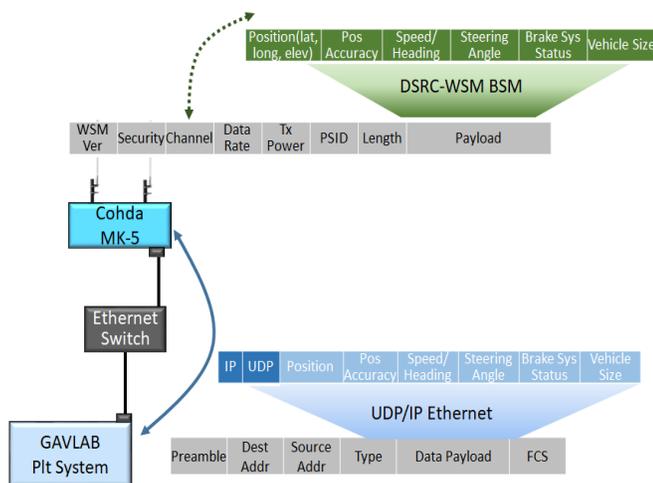


Figure 2- Cohda Radio Interface



Figure 3- Example RADAR Returns

2.3. GPS Receiver

A Novatel FlexPak6D and a dual frequency antenna provides a vehicle position measurement. These measurements are fed directly into the computer for additionally processing in the algorithms described in section 3.

2.4. Computer

A Nuvo-5095GC computer handles all of the data importing, processing, and interfacing. The computer receives data from the GPS, RADAR, DSRC, and vehicle Controller Area Network (CAN) bus. Additionally, the computer generates CAN messages to actuate the low-level vehicle systems for automated control. The computer runs the Linux Ubuntu 16.04 LTS distribution.

2.5. Low-level by-wire system

The low-level by-wire system covers everything from the electric steering motor to the individual Electronic Control Units (ECU), which communicate over the CAN bus. A general overview of these systems and their interactions can be seen in Figure 4. A CAN gateway is used to monitor the J1939 [10] CAN traffic, and allows the user to terminate commanded CAN messages at any point if unsafe conditions occur.

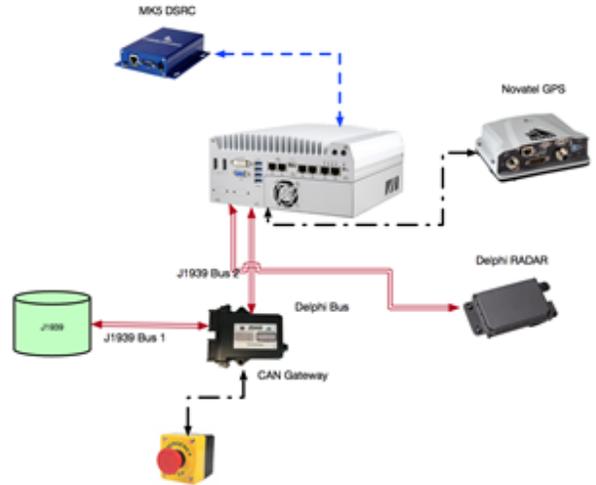


Figure 4- Auburn Hardware and Connectivity Map

3. CACC Algorithms

Auburn University has implemented a number of algorithms to enable the longitudinal control of a CACC platoon. Fundamentally, these algorithms are for range estimation and control. Additional features, such as cut-in detection and ACC control, have been added to the platooning system.

The algorithms are discussed in depth to provide a more holistic view of the CACC system. Furthermore, an understanding of how these algorithms impact the results will provide deeper comprehension of the future work to be accomplished.

3.1. Range-Estimation

The range-estimation algorithm is the backbone of the longitudinal control algorithm and helps alleviate several shortcomings of other sensors. The main function of the range-estimation algorithm is to combine the low frequency GPS position estimate with the higher frequency RADAR measurements. By using both measurements, the range estimator provides a high update-rate range solution with high accuracy.

The GPS measurements from the lead truck are differenced with the GPS measurements of the follower to form centimeter-level accurate relative

position vectors. However, the Dynamic-Base Real Time Kinematic (DRTK) algorithm only provides a solution at ~ 2 Hz, slower than the bandwidth of the system. This makes the GPS measurement alone unsuitable for being the sole range measurement in control applications.

The RADAR, in contrast, provides both range and range-rate measurements at ~ 20 Hz, which is fast enough for real time control. However, the RADAR measurements have a higher variance than the GPS measurements, and there is ambiguity in the RADAR tracks. To mitigate the variance and ambiguity, the RADAR and GPS measurements are combined in a Kalman filter, which provides a low variance, high update rate solution. The Kalman filter is referred to as the range-estimator in the Auburn software library.

The range-estimator provides updates for the state vector $x = [r \ \dot{r} \ \beta]^T$, which contains range r , range-rate \dot{r} and bearing β , respectively. These states are updated at ~ 20 Hz and are used for longitudinal control of the vehicle. Figure 5 provides a diagram showing the states described above.

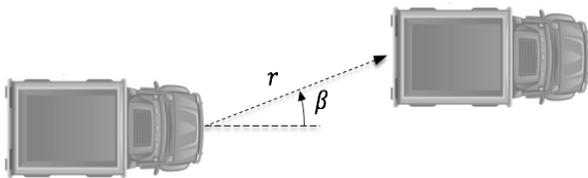


Figure 5- Diagram of Range-Estimator States

Besides estimating critical states for control, the range-estimator also performs a critical safety task. The range-estimator keeps track of RADAR signatures from neighboring vehicles, and will determine if a neighboring vehicle has inserted into the platoon and become a “cut-in” vehicle. If a cut-in is detected, the system switches to an Adaptive Cruise Control (ACC) mode and falls back to a safe following distance until the cut-in vehicle exits the platoon.

The cut-in detection works on the concept of a pure-pursuit model. The pure pursuit model fits a

circle through the leader and followers’ position. The arc that travel through the leader and follower is then used to generate a path, using the width of a typical road lane. The path duplication was heavily influenced by previous work done in generating a path for a vehicle using ACC [5]. Once the path bounds are generated, the range estimator checks its “tracked vehicles” output, which is the estimated range to neighboring vehicles, and looks to see if any of those vehicles fall within the generated bounds. If a neighboring vehicle is within the bounds, and stays in for more than $\sim 1/4$ second, a cut-in is declared and the truck switches from a CACC mode to Adaptive Cruise Control (ACC) mode. Figure 6 shows a generated path between two Peterbilts, and Figure 7 shows the cut-in detection checking for a cut-in vehicle.

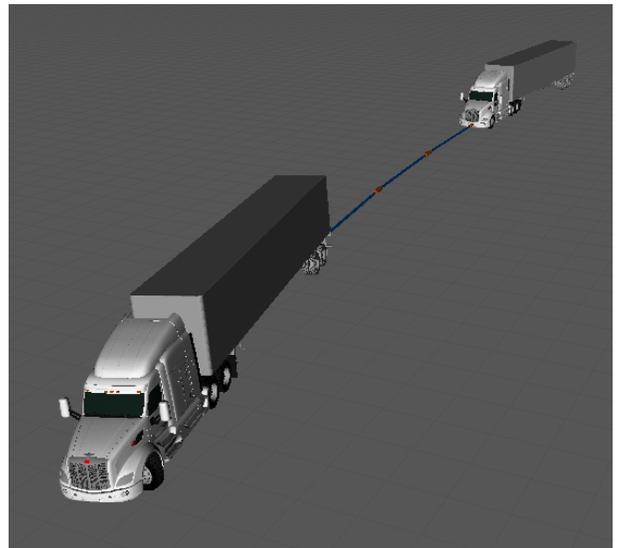


Figure 6- Pure Pursuit Path

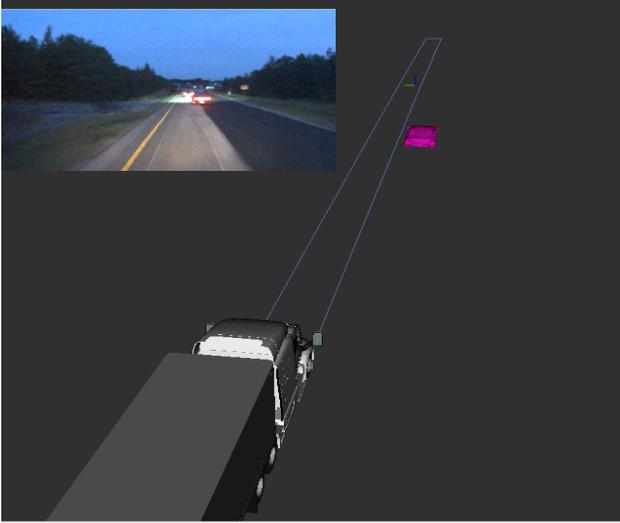


Figure 7- Cut-In Detection with Cut-In Bounds Displayed

3.2. Dynamic-Base Real Time Kinematic

One of the core software libraries in the system is the Dynamic-Base Real Time Kinematic (DRTK) positioning library. DRTK is an extension of the Real-Time Kinematic (RTK) positioning algorithm, with the key difference being the base station is not static in DRTK.

Since DRTK algorithm has been researched, a simplified overview is presented here [4]. For a GPS position solution, a minimum of 4 satellites are required to be in view of a single receiver in order to generate an accurate position solution. Four satellites are needed to calculate the X, Y and Z position of the receiver, as well as the receiver clock bias. However, atmospheric effects can cause unwanted errors of a meter or more. By differencing raw GPS observables, one is able to effectively cancel out atmospheric effects and develop an extremely precise relative position vector between vehicles, rather than an absolute position that RTK algorithm produces. This relative position vector is then used to help generate high-accuracy range measurements for use in the longitudinal control algorithm.

3.3. Longitudinal Control

The longitudinal control algorithm is the centerpiece of the CACC system. This algorithm

takes in the outputs of the range-estimation algorithm, i.e. $[r \dot{r} \beta]^T$, and uses a Proportional-Integral-Derivative (PID) control architecture with feed-forward to maintain a desired range, or headway. Maintaining a constant headway is important because fuel savings are directly related to the following distance of platooning trucks [6].

On system start-up, the controller evaluates a file containing current system parameters, and generates a set of controller gains using Equations (1-3) seen below. In Equations (1-3), τ_1 , τ_2 and τ_3 are time constants defined by the engineer, with m_{eff} , b_{eff} , R_{eff} , n_{trans} and n_{diff} being effective mass, effective damping, effective wheel radius, transmission and differential gear ratios, respectively.

$$k_P = -\frac{m_{eff}R_{eff}}{n_{diff}n_{trans}} \left(\frac{1}{\tau_1\tau_2} + \frac{1}{\tau_1\tau_3} + \frac{1}{\tau_2\tau_3} \right) \quad (1)$$

$$k_I = -\frac{m_{eff}R_{eff}}{n_{diff}n_{trans}} \frac{1}{\tau_1\tau_2\tau_3} \quad (2)$$

$$k_D = -\frac{m_{eff}R_{eff}}{n_{diff}n_{trans}} \left(\frac{1}{\tau_1} + \frac{1}{\tau_2} + \frac{1}{\tau_3} - \frac{b_{eff}}{m_{eff}} \right) \quad (3)$$

The PID controller takes in the headway error, h , and outputs a desired engine torque, T . This can be seen as a block diagram in Figure 8. The T_{FF} term is a feed-forward torque, which is defined in equation 4, where C_d is the trucks coefficient of drag, and T_{rr} is the torque required to overcome rolling resistance. The output torque shown in Figure 8 is commanded to the engine via the CAN network.

$$T_{FF} = m_{eff} * accel_{lead} + b_{eff} * vel_{lead} + C_d * vel^2 + (mass * g * \sin(grade)) + T_{rr} \quad (4)$$

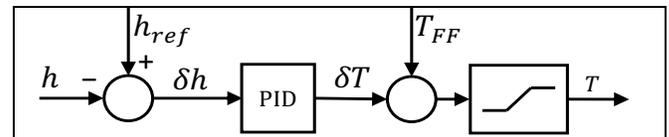


Figure 8- Auburn Controller Block Diagram

4. Results

This section presents the results from extensive testing performed on Interstate 275 and 696 in Michigan and on highway 280 in Alabama. The subsequent sections will focus on the control performance, range-estimator and cut-in performance, and the radio network.

4.1. Control Performance

The longitudinal controller implemented on the Auburn University trucks tested extensively at ACM and surrounding highways. At ACM, the trucks platooned for 614 miles over 5 days of testing. Through the course of these platooning events, several aspects of the controller were noted. The main areas of focus were on the steady-state performance of the system, and the stability of the platoon as the number of trucks was increased.

The steady-state performance of the longitudinal controller is important for several reasons. First, the steady-state performance plays a large role in the expected fuel savings. Additionally, poor steady state performance could result in an unsafe scenario, especially at close following distance. Figure 9 presents the normal steady-state behavior of the Auburn University trucks. These results occur with low variations in grade, and no outside vehicle inserting into the platoon.

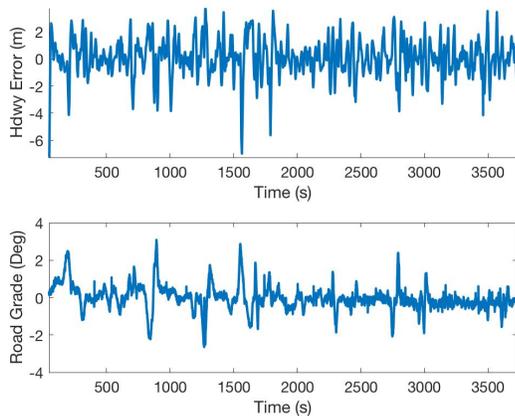


Figure 9- Steady-State Platooning Results in an Optimal Environment

To help achieve the desired steady-state performance, the controller uses a feed-forward torque. However, there were a few issues that presented themselves during testing. The first issue was the sensitivity of the feed-forward torque to noise in the lead acceleration measurement. Due to large m_{eff} for class 8 vehicles, sign changes of the acceleration measurement resulted in a substantial change to the feed-forward torque. This manifested as control chatter, which would attempt to apply the service brakes while also commanding an engine torque to maintain the set gap. Figure 10 presents the output torque requested without the acceleration measurement being filtered. Immediately after the test presented in Figure 10, the lead acceleration was assumed to be zero. Figure 11 shows the desired results of the run immediately following this change.

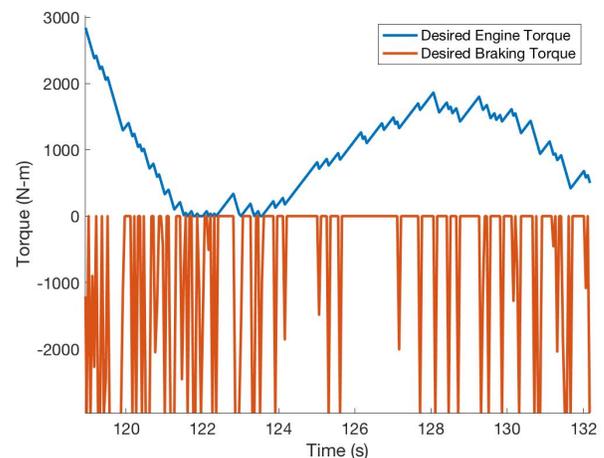


Figure 10- Control Chatter Due to Noisy Measurements

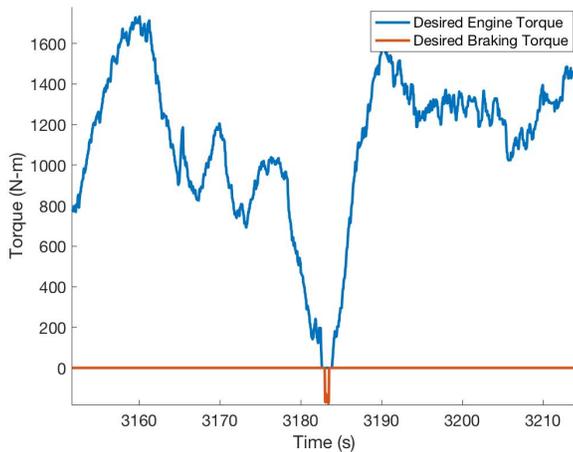


Figure 11- Control Output with Acceleration Removed

Lastly, Auburn University had the chance to test the CACC system on a four-truck platoon at the National Center for Asphalt Technology (NCAT) test facility in Auburn, Alabama. NCAT is a 1.8-mile long test track with flat straight sections and is pictured in Figure 12. This provided a “best-case” environment to test the controller’s ability to scale to larger platoon sizes. The testing was performed with two Peterbilt 579’s and two military M915’s which are pictured in Figure 13.



Figure 12- NCAT Test Track



Figure 13- Auburn University Test Vehicles

When the results from the four-vehicle platooning tests were analyzed, an interesting trend emerged. When the headway errors were aligned in time, using GPS time, the headway error seemed to grow for each subsequent truck in the platoon, as shown in Figure 14. This is an issue known as string stability and can become a safety problem as platoon sizes grow. Although V2V communication is required for the string stability of constant gap platooning, the use of V2V communication does not guarantee string stability [3]. This is an important issue to address in future controller implementation since scalability, or the ability to use this control architecture with larger platoon sizes, will be required to realize many of the benefits of platooning.

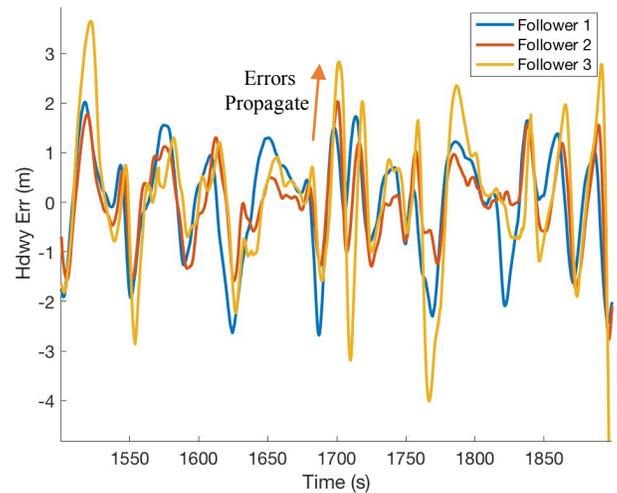


Figure 14- Headway Errors in a Four-Truck Platoon

4.2. Range Estimation/Cut-ins

The range estimator was an additional area of interest for two reasons. The first was that it provides the primary input to the longitudinal controller, and secondly, it performs the safety critical task of cut-in detection.

Figure 15 presents the normal operating procedure for a platoon when a vehicle inserts into the platoon. It can be seen that when the range-estimator detects a vehicle within the generated path bounds, it sets a value called “cut-in detected” equal to 1. When this occurs, the truck switches to

an ACC mode, and fall back to a safe following distance. This is represented by the large spike in headway error, representing the truck following too closely to the cut-in vehicle. When the vehicle leaves, the truck has switched back to the platooning mode, and closes back to the set following gap.

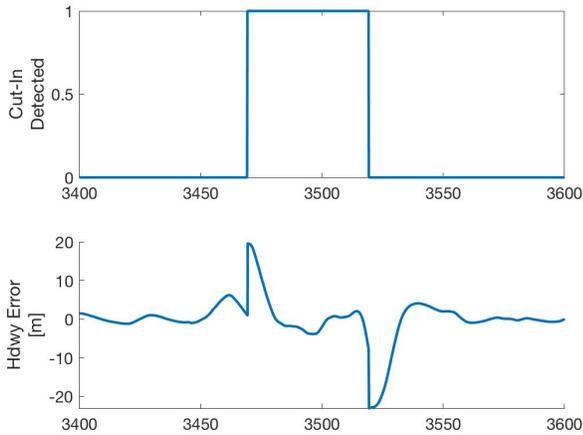


Figure 15- Normal Fallback Procedure for Cut-in Events

While Figure 15 is representative of many results collected in real-world tests, there were scenarios where the pure-pursuit model did not pick up the cut-in vehicle as intended. This stems from the sensitivity of the path bounds to the relative position of the leading and following trucks. If a vehicle is entering the lane and just crosses the path bounds, the system may begin to declare a cut-in. However, if the leading truck then shifts slightly in its lane, the path bounds will move, and the vehicle may no longer be declared a cut-in. Figure 16 presents a few other scenarios in which the pure-pursuit model breaks down, and Figure 17 shows how the false cut-out signals manifest in the headway control.

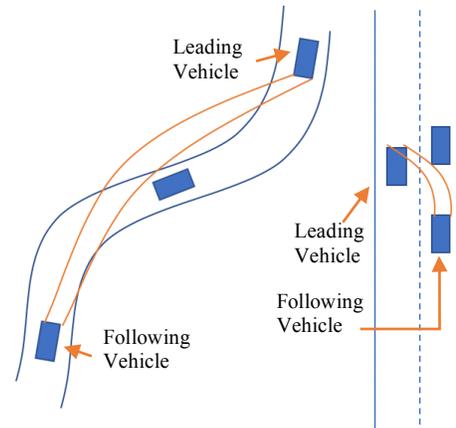


Figure 16- Pure-Pursuit Model Failure Modes

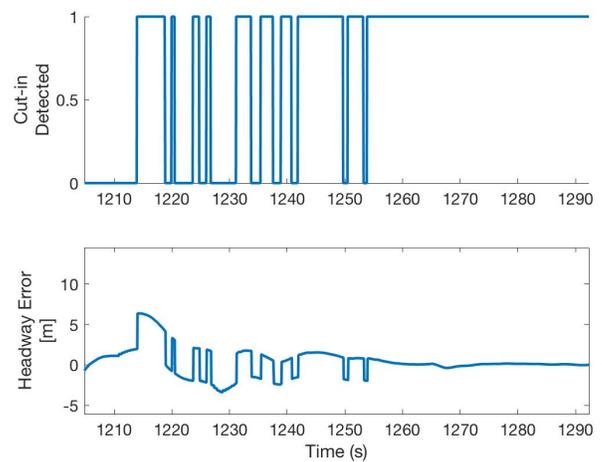


Figure 17- Cut-in w/ Pure Pursuit Failure Scenario

4.3. Radio Network

The radio network was the final key area of interest throughout development and testing. The network health is critical for CACC platooning. The information shared among vehicles is needed for initialization and transient operation of the system. Therefore, packet loss and latency were identified as two key parameters for network health. Throughout testing, the radio network was proven to be reliable with a few exceptions to be discussed below.

As previously discussed, high packet latency is undesirable because information is no longer valid when it is received. A similar problem occurs when the radio network has packet loss, or when a packet

is never received at all. Packet loss was observed several times during testing and was determined to occur when a maximum packet size was exceeded. Experimentally, the maximum packet size was found to be ~1300 bytes, and this issue occurred when a large number of GPS satellite measurements were observed by the GPS receiver. Specifically, the byte limit was exceeded when greater than ten satellites were observed. During this scenario, the radio network failed to transmit the large packet, i.e. the packet was dropped. Although the platooning system can handle small outages, this problem can persistent long enough to prevent the CACC system from operation.

Several solutions were investigated and implemented to resolve this issue. As an immediate fix, the Novatel GPS firmware was changed to limit measurements based on the mask angle, or low elevation relative to the GPS receiver. Furthermore, another solution was to filter out the quality of received GPS measurements, based on the carrier to noise ratio. This solution is implemented in the CACC software and requires no changes to the hardware. Lastly, another solution can be implemented in the DSRC radio software. One method includes truncating a GPS measurement from the UDP packet before transmitting. Overall, a combination of these solutions is needed for a robust method to guarantee the byte limit is below the desired amount.

5. Conclusions/Future Work

This paper has attempted to show the true complexity of the platooning problem. Even with numerous sensors and diagnostic tools available, minute issues resulted in things such as undesirable control output, cut-in vehicles not being detected, and losing V2V communication.

Auburn University has begun an investigation of the effects of feeding forward acceleration, and is weighing various fixes. The two solutions are to implement a low pass filter to attenuate sensor noise, or to remove the acceleration term from the feed-forward torque all together. The solution that

will be implemented will be picked after a more thorough analysis of the data is performed. The analysis will aim to determine the exact impact of feeding forward the lead vehicles acceleration on various types of terrain.

Furthermore, future work is being developed for a more robust cut-in detection system.. One such idea was to consider the yaw rates of both vehicles, and if the leader/follower are yawing in different directions, generate a spline. A basic visualization of this is shown in Figure 18.

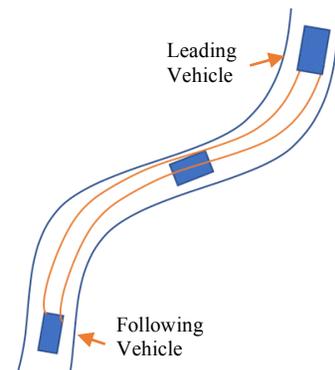


Figure 18- Path Generation using Spline Approach

An alternative solution is adding additional sensors to aid the cut-in detection process. One such solution would utilize a forward-facing camera and perform lane tracking. Andra Petrovat et al. successfully implemented a similar system, where a stereocamera was used for path generation and object detection [7]. With lane tracking and a relative position vector to the lead vehicle, one can check the lane boundaries for objects between the lead and following vehicles. If combined properly with the existing system, the stereocamera would allow for more robust cut-in detection.

Lastly, the radio software is in the process of being updated. Future work includes increasing the byte limit, hopping messages from vehicle to vehicle, and scalability of the network. These updates should allow for a continuous V2V communication network. Overall, this future work is incremental to make the system more robust to

the real world, but provides a system that is adaptable and flexible for platooning testing.

Acknowledgments

The Authors would like to thank the US Army Combat Capabilities Development Command, Ground Vehicles System Center for sponsoring much of this work, and allowing the use their M915's for testing. FPInnovations was also a key partner in allowing the team to test on forest road and other terrains in Canada. Finally, Rob Daily of Integrated Solutions for Systems (IS4S) was a key member of not only developing the longitudinal controller, but also implementing the software library.

1. REFERENCES

- [1] "Average Annual Vehicle Miles Traveled of Major Vehicle Types," Nov. 11, 2018. Accessed on: April 19, 2019. [Online]. Available: <https://afdc.energy.gov/data/10309>
- [2] "Average Annual Fuel Use of Major Vehicle Types," Nov. 11, 2018. Accessed on: April 19, 2019. [Online]. Available: <https://afdc.energy.gov/data/10308>
- [3] Yanakiev, D. and Kanellakopoulos, I., "Nonlinear spacing policies for automated heavy-duty vehicles," *Vehicular Technology, IEEE Transactions on*, vol. 47, no. 4, pp. 1365-1377, 1998.
- [4] Tabb, Thomas et al., "Improved Relative Positioning for Path Following in Autonomous Convoys" In *Proceedings of the Ground Vehicles Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 7-9, 2018.
- [5] Baum, D., Hamann, C.D. and E. Schubert., "High Performance ACC system Based on Sensor Fusion with Distance Sensor, Image Processing Unit, and Navigation System," *Vehicle System Dynamics*, 28:6, 327-338, DOI:10.1080/00423119708969360
- [6] Browand, F. et al., "Fuel Saving Achieved in the Field Test of Two Tandem Trucks." *UC Berkley: California Partners for Advanced Transportation Technology*
- [7] Petrovai, A, et al., "A Stereovision Based Approach for Detecting and Tracking Lane and Forward Obstacles on Mobile Devices"
- [8] Torrey IV, W.F. and Murray D., "An Analysis of the Operational Costs of Trucking: A 2014 Update," 36, <http://www.atri-online.org/wp-content/uploads/2014/09/ATRI-Operational-Costs-of-Trucking-2014-FINAL.pdf>, accessed: April 2019
- [9] SAE V2X Core Technical Committee, "Dedicated Short Range Communications (DSRC) Message Set Dictionary: A March 2016 Update," J2735_201603, https://saemobilus.sae.org/content/j2735_201603, accessed: October 2019
- [10] SAE Truck Bus Control and Communications Network Committee, "Serial Control and Communications Heavy Duty Vehicle Network - Top Level Document" J1939_201808, https://saemobilus.sae.org/content/J1939_201808/, accessed: February 2019