EMPIRICAL ANALYSIS OF A FREEWAY BUNDLED CONNECTED AUTOMATED VEHICLE APPLICATION USING EXPERIMENTAL DATA

Jiaqi Ma[a]*; Edward Leslie[b]; Amir Ghiasi[c]; Zhitong Huang[d]; Yi Guo[e]

a. Assistant Professor, Department of Civil and Architectural Engineering and Construction Management, University of Cincinnati, 765 Baldwin Hall, Cincinnati OH, jiaqi.ma@uc.edu (Corresponding author)

b. Senior Electrical Engineer, Leidos, Inc., 11251 Roger Bacon Drive, Reston, VA 20190, edward.m.leslie@leidos.com

c. Transportation Engineer, Leidos, Inc., 11251 Roger Bacon Drive, Reston, VA 20190, amir.ghiasi@leidos.com

d. Senior Transportation Engineer, Leidos, Inc., 11251 Roger Bacon Drive, Reston, VA 20190, zhitong.huang@leidos.com

e. Graduate Research Assistant, Department of Civil and Architectural Engineering and Construction Management, University of Cincinnati, 765 Baldwin Hall, Cincinnati OH, guo2yi@mail.uc.edu

ABSTRACT: Connected and automated vehicles (CAV) hold the potential for substantial improvements to traffic safety, travel time reliability, driver comfort, roadway capacity, environmental impacts, and users' overall travel experience. Numerous modeling and simulation studies have been conducted to evaluate these impacts. However, model accuracy and simulation assumptions limit the validity of evaluation results. These factors have resulted in the wide range of differences in effectiveness among studies examining the same CAV applications available in the literature. In this study, we propose a bundled CAV application that involves platoons of equipped vehicles governed by an integrated set of cooperative adaptive cruise control (CACC), cooperative merge, and speed harmonization applications. We implemented the bundled application in a fleet of five vehicles at the Saxton Transportation Operations Lab of the Federal Highway Administration. Experiments were conducted to collect and compare data on CAV and human-driven behavior. Based on the real experimental data, our results show that the performance of the CAV operations, including platooning and cooperative merging under varying Infrastructure-to-vehicle speed commands, demonstrate string stability. The results also present key behavioral parameters of the vehicles and strings. This will eventually help the research community, particularly the modelers, to come up with models with realistic performance to further understand the CAV impacts on traffic. The results can also serve as references for transportation agencies to make informed decisions on infrastructure and traffic management decisions.

KEY WORDS: connected automated vehicle (CAV); Cooperative Adaptive Cruise Control (CACC); Cooperative Merge; Speed Harmonization; Field Experiment; Vehicle Behavior

INTRODUCTION

Connected and automated vehicle (CAV) technologies offer potentially transformative traffic impacts, including significant mobility, safety, and environmental benefits. Numerous modeling studies have been conducted to evaluate these impacts. However, model accuracy and simulation assumptions limit the validity of evaluation results; additionally, the lack of field data exacerbates this problem of inaccuracy, leading to improper model calibration.

In this study, we are interested in a bundled CAV application that involves platoons of equipped vehicles governed by an integrated set of cooperative adaptive cruise control (CACC), cooperative merge, and speed harmonization applications. We implemented the bundled application in a fleet of five vehicles at the Saxton Transportation Operations Lab of the Federal Highway Administration (FHWA). These three applications are selected because their effectiveness has been demonstrated in simulation and small-scale tests in the literature (*1-11*) as the most promising application for improving both efficiency, capacity, and sustainability.

Cooperative Adaptive Cruise Control (CACC)

The class of CACC systems utilizing V2V communication could potentially allow the mean following time gap to be reduced from about 1.4 seconds when driving manually to approximately 0.6 seconds when using CACC (*1*), resulting in an increase in highway lane capacity. Several highway traffic simulations conducted by the California Partners for Advanced Transportation Technology (*2, 3*) showed that autonomous ACC alone, even at high market penetration rates, had little effect on lane capacity, and recent on-the-road experiments (*4*) have shown that a stream of autonomous ACC vehicles is string unstable, resulting in a negative impact on lane capacity and safety. However, with the shorter following gaps enabled by CACC systems, lane capacity could potentially be increased from the typical 2,200 vehicles per hour to almost 4,000 vehicles per hour at 100 percent market penetration.

Cooperative Merge

The concept of cooperative merging leverages V2V and V2I communications to enable CAVs to signal other vehicles (e.g., via dedicated short-range communication, or DSRC) of their intention to merge into traffic streams. Using this information, merging vehicles may identify upcoming acceptable gaps on the mainline and make lane changes when possible. In addition, upstream managed lane vehicles may cooperate by adjusting their speeds to create a gap for the requesting vehicle. The trajectories of merging vehicles are then optimized. The merge movement can then occur safely and with minimal impact on the platoon’s mobility.

A recent Federal Highway Administration (FHWA) study (*6*) describes an effort in developing an innovative vehicle control platform to successfully conduct a proof-of-concept field experiment of a cooperative lane change maneuver driven by a simple algorithm. This demonstration was executed using automated speed control, vehicle-to-vehicle (V2V) communications, and vehicle-based radar systems. Experimental results show the effectiveness of the proposed platform and the successful proof of the concept of cooperative lane change. Chou et al. (*7*) tested in simulation two cooperative automated merging strategies for highway entry, one using I2V communication and the other using V2V communication in microscopic simulation. The results show that I2V reduced travel time in the merging section when the traffic flow was high, and the V2V case supports a significant increase in traffic flow without increasing travel times. The results indicate the potential advantages of using cooperative automation to relieve the bottleneck in the merging section. Scarinci et al. (*17*) presents a novel merging assistant strategy that exploits the communication capabilities of intelligent vehicles. The proposed control requires the cooperation of equipped vehicles on the main carriageway in order to create merging gaps for on-ramp vehicles released by a traffic light. The study develops a macroscopic-level control and simulation testing results shows great potential of such strategies in improving throughput and reduce delay.

Speed Harmonization

Generally speaking, speed harmonization involves gradually lowering speeds upstream of a heavily congested area in order to reduce the stop-and-go traffic that contributes to frustration and crashes. To date, a related strategy known as variable speed limits (VSL) has been applied at several locations in Europe and a few locations in America (*8*). Although VSL systems may achieve speed harmonization when successful, driver response to suggested speed targets has not been consistent. Dynamic speed limit adjustments are less efficient than dynamic adjustments of recommended and/or actual speeds communicated directly into connected and automated vehicles as the speeds are adjusted automatically unless drivers intervene. Compared to the segment-based speed harmonization (which provides the same speed recommendation or command for all traffic on a freeway segment) (*9*), trajectory-based speed harmonization is a category of more advanced approaches that control and coordinate an individual vehicle's trajectories. Individual vehicle speeds can be controlled in real time depending on each vehicle's location on the roadway segment to enable them to smoothly pass downstream bottlenecks. Recent simulation studies (e.g., *10*) and field experiments (*11*) suggest the potential of such an approach in enhancing traffic smoothness and therefore improving efficiency and safety. In particular, trajectory control can facilitate freeway merge. In this scenario, a central controller (e.g., traffic management center) coordinates the trajectories of upstream managed lane vehicles and merging vehicles such that smooth and efficient merging and minimum impact on mainline traffic can be guaranteed.

Research Goal

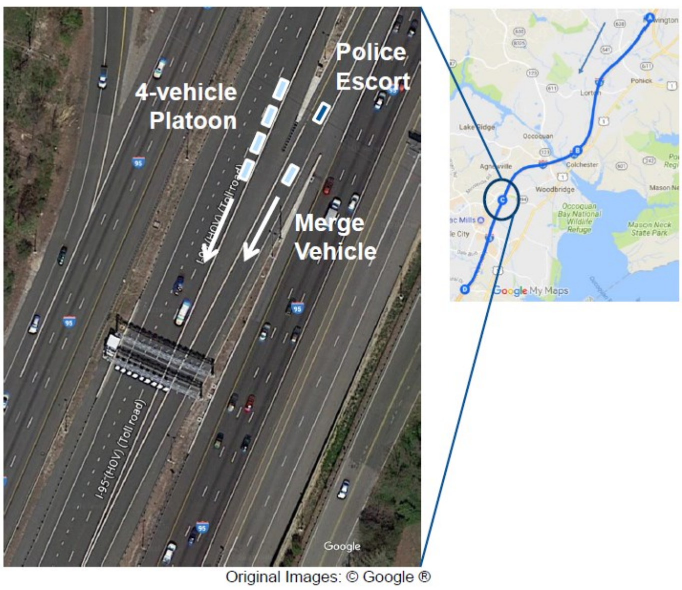
The goal of the work is to accomplish the following two using real vehicle experimental data: 1) developing and testing the new vehicle platform that is implemented with the bundled application; and 2) evaluating the performance and behavior of the CAV vehicles and strings to confirm the effectiveness of the embedded algorithms. This will eventually help the research community, particularly the modelers, to come up with models with realistic performance to further understand the CAV impacts on traffic. The results can also serve as references for transportation agencies to make informed decisions on infrastructure and traffic management decisions.

In this study, after reviewing these applications, we discuss the experimental design, hardware and software implementation. Then we conduct an empirical analysis of the experimental data, particularly during platooning and merging processes. Last, the CAV vehicle and string longitudinal behavior is analyzed to derive key parameters and managerial insights.

FIELD EXPERIMENT DESIGN

Experiment Site and Process

This section documents field experiments that were conducted on the I-95 Express Lanes facility, owned by Virginia DOT and managed by Transurban. The selected location for the merge area is just south of exit 158 on the I-95 Express Lanes, near Potomac Mills.



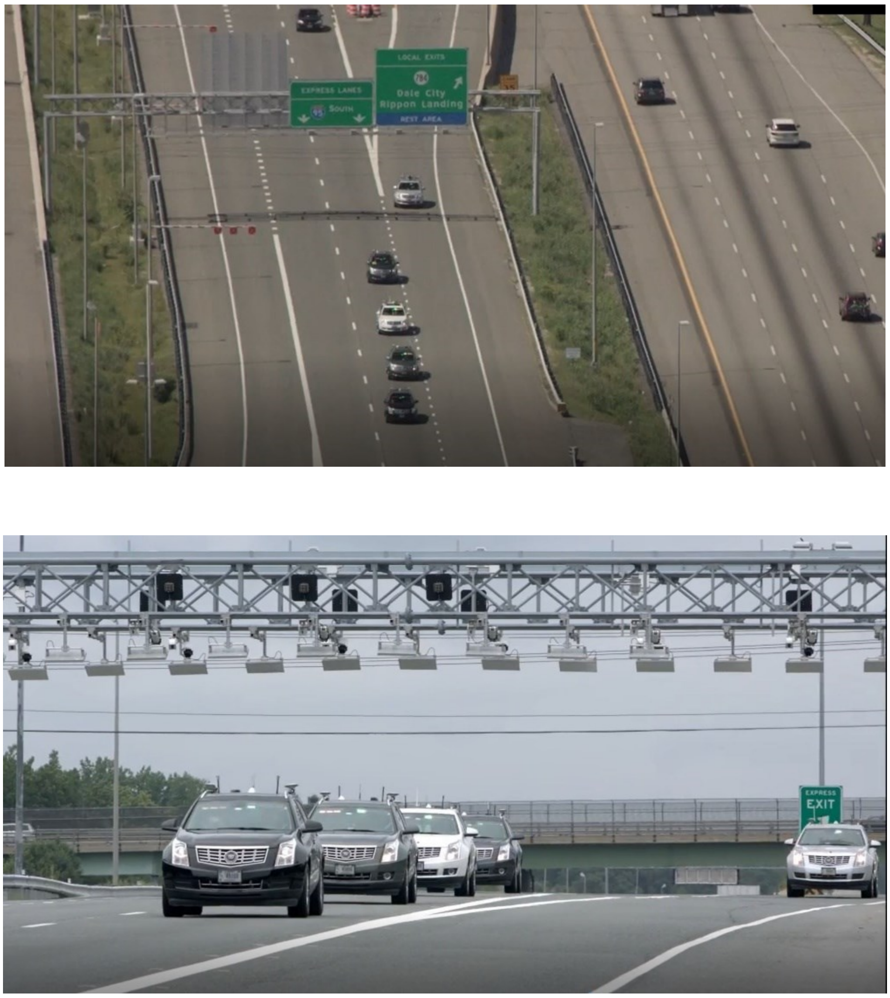


Figure 1. Southbound Test Routes, Geometry and Site Scene

The experiment was designed to be carried out with five vehicles: four vehicles traveling down the leftmost managed lane of the facility and one vehicle merging into the managed lane. The experiment was performed with multiple scenarios to evaluate the effectiveness of the connected vehicle applications on traffic flow. In one case, the platooning vehicles were driven manually and the merging vehicle was released in a fashion similar to a metering light and manually driven to the merge point. In the other case, the vehicles executed the platooning maneuver cooperatively and the merging vehicle was released automatically to join the platoon. Data such as vehicle speed, following distance, and time to merge were recorded and compared between the two cases.

Vehicle and Infrastructure Platform

The team used the FHWA Saxton Lab's fleet of Cooperative Automation Research for Mobility Applications (CARMA) vehicles, as shown in **Figure 2**(a), and they use a robot operating system (ROS) framework for testing and evaluating connected automated vehicle (CAV) applications. The platform provides for the implementation and execution of custom algorithms in the form of plugins to the vehicle software's guidance module. These plugins are loaded onto the Linux PC installed in the vehicle and then selected by the user at software startup. Once selected by the user, the plugins are allowed to participate in a cooperative motion planning process for the vehicle, working in conjunction with all other enabled plugins to generate a speed profile for the vehicle to execute. Once a plan has been generated by the plugins and accepted by the vehicle's validation system (to ensure that the trajectory passes basic sanity checks, obeys the local speed limits, does not conflict with known current or future locations of other vehicles, etc.) these maneuvers are then executed.

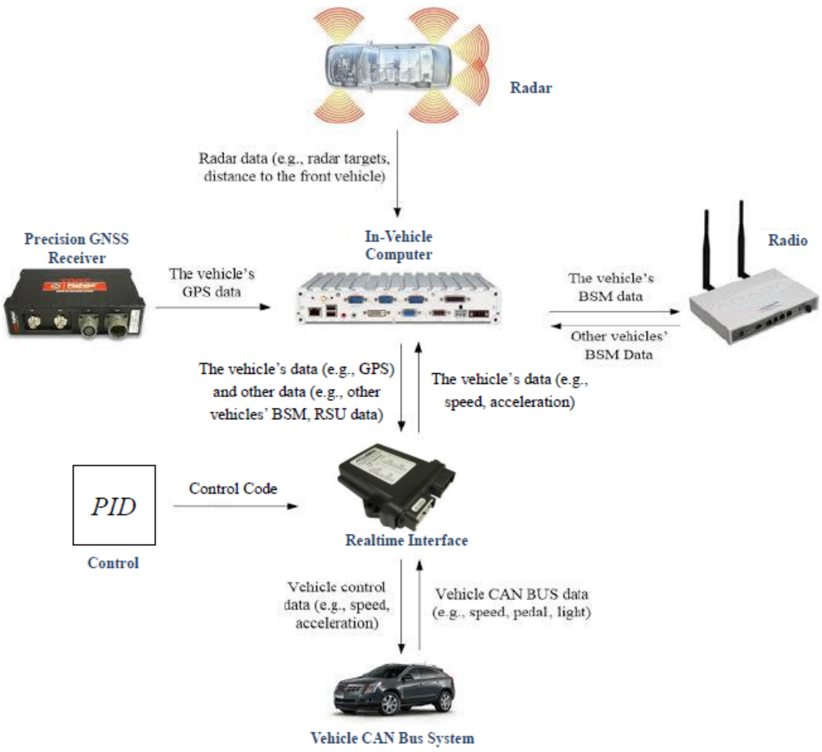
The CARMA vehicle's hardware platform is built on top of a 2013 Cadillac SRX. The vehicles are outfitted with a drive-by-wire system that is capable of overriding the stock ACC system to directly command throttle and brakes over the vehicle CAN bus. Speed commands are generated by the CARMA platform on the in-vehicle Linux PC and are sent to a real-time control module with a proportional-integral-derivative (PID)-based speed control loop.

For localization, the CARMA platform relies on a Pinpoint device which uses a combination of dual GPS and an inertial measurement unit to produce an accurate reading on the vehicle location and altitude which remains available in the face of momentary GPS satellite drop out due to occlusion or other issues. The CARMA platform uses the stock forward-facing radar sensor via a tap into the vehicle's forward object CAN bus and the data are provided to the Linux PC and its SRX objects driver and sensor fusion system. Other data elements, such as vehicle speed and ACC status, are also available from the high-speed CAN bus. The CARMA system includes networking hardware that provides a cellular access point to enable communication with the standard internet while the vehicle is in motion. The vehicle's DSRC communications are powered by a Cohda OBU forwarding messages to and from the Linux PC for processing. At the merge area, one mobile RSU as shown in

Figure 2(b) is deployed for vehicle-to-infrastructure communication. The vehicle platform structure is shown in **Figure 2**(c),

(a) Testing Vehicle

(b) Mobile Roadside Unit



(c) Vehicle Platform Structure

Figure 2 Experimental Hardware

Platooning

The platooning plugin, which is included in the vehicle, uses an algorithm developed by (*15*). This algorithm provides means for leader selection when two or more vehicles begin to communicate with each other. Once a leader is selected, the following vehicle plans a motion profile which will bring it in close proximity to the lead vehicle or the rearmost vehicle in the lead vehicle's platoon. Joining a platoon is only possible from the rear, and no allowances are made for cut-in joins). Once the following vehicle is in position, it switches to controlling its target gap to the lead vehicle (based on position reported via DSRC) and its gap to the preceding vehicle (as measured by radar) by means of a standard PID controller and smoothing filters. At all points in time during this platooning operation, the CARMA platform's basic radar-based ACC system, not the Cadillac built-in ACC, is enabled with a significantly reduced headway setting to act as a safety measure should platooning fail for any reason. But note that the radar-based ACC system may not be a perfect safety backup due to the potential response delay. Safety backup is out of the scope of this paper and we will consider additional technologies for this purpose in future system enhancements.

Speed Harmonization

The speed harmonization plugin included with the CARMA platform enables remote server control of the vehicle's speed. The speed harmonization server itself is deployed remotely in the cloud and communicates with the plugin running in the vehicle via HTTP/REST through a cellular modem installed on the vehicle's network. The plugin on the vehicle is responsible for sending the remote server periodic status updates with regard to the vehicle's automation status, location, and speed, which will be used by the algorithms configured to run on the server itself. Once enabled by the user and after it has established communications with the remote server, the speed harmonization plugin will insert speed commands into the vehicle's motion profile that will allow it to execute the speed commands the server sends it. As the server is able to send speed commands at a variable rate the vehicle will maintain the last received speed command if one was sent or it will simply maintain the current speed of the vehicle until the first speed command is received. Since it is not possible to create a traffic condition in our experiment, we use microscopic traffic simulation to simulate a congested traffic condition.

In this work, the speed harmonization is based on the offline generated simulation traffic. We first model the experimental segment and simulate the traffic. The simulation data were then stored in a local database. We assume the CACC vehicles will enter the segment at certain simulation time and this one platoon will not impact overall traffic performance. When the platoon slows down, it will not affect the downstream traffic at all, but only upstream vehicles in the vicinity. Therefore, we will feed the speed harmonization algorithm with traffic data at the downstream bottleneck location based on the stored simulation data. We use a simple speed-based speed harmonization algorithm for this experiment and therefore the input to the speed harmonization algorithm is bottleneck area speed and location of the platoon. This is the algorithm that has been used in many variable speed limit deployments and an earlier CAV speed harmonization algorithm (*18, 19*). Technically, this algorithm can be replaced by any other algorithms. The focus of this paper is not to show the effectiveness of speed harmonization, but rather the feasibility of bundling different applications.

Cooperative Merge

A cooperative merge plugin for the vehicle was developed under an earlier experiment (*6*). This plugin enabled the vehicle to coordinate with a roadside unit (RSU) deployed at the experimental merge area, as shown in **Figure 2**(b), and with the experimental platoon vehicles. The Cooperative Merge plugin listens for the periodic broadcasts from the RSU, which announce its presence and availability, and upon approaching the ramp-metering point (where a traffic signal would normally be) it places itself under the direct control of the RSU. This direct RSU control is accomplished via 10-hertz messages from the RSU containing speed commands, which are used by the cooperative merge to immediately control the vehicle. The RSU will command the vehicle to a stop at the merge point and hold it there until it determines that the rearmost vehicle of the approaching platoon is in such a position that the merge would be acceptable. At this point, the RSU commands the vehicle to accelerate to the speed of the platoon. Due to the way the vehicle's original equipment manufacturer (OEM) adaptive cruise control system functions when coming to a stop, it is necessary for the driver to apply a minimal amount of throttle and re-engage the ACC system to re-authorize automated control. When the merge is complete, the vehicle is released from RSU control and begins execution of the platooning plugin logic to join the platoon.

In our experiment, the trajectory of the merge vehicle is controlled throughout the process. The merge vehicle will receive information from the RSU about the real-time location of the platoon. Then, the merge vehicle on-board computer will calculate the best trajectory to merge into the end of the mainline platoon. The merge vehicle first stops at an upstream location of the on-ramp because of the experimental setup of assuming a ramp metering available at the stop point. This is to consider the scenario of creating high-performance traffic streams on the mainline (e.g., CAV managed lanes) and only allow an on-ramp merge vehicle when a gap is detected or created. This is an infrastructure-based control scenario. However, it is also possible to eliminate the ramp metering for general speed or trajectory control of merge vehicles. Our experiment also tests that case in which the merge vehicle speed control algorithm (or any trajectory control) considers an initial speed of zero.

RSU Metering Application

The RSU metering application was purpose-built in this study to enable the merge vehicle to closely synchronize its merge with the passing platoon. This application continuously broadcasts requests to begin metering while waiting for a response from the merge vehicle. Upon the merge vehicle's response (acknowledging intent to allow RSU control) the RSU metering application sends commands to the merge vehicle to stop and hold at the configured metering point, a digital analogue for where the traffic light might be in a more traditional ramp metering scenario. Once the vehicle is brought to a stop, the RSU metering application begins listening for status mobility messages from an approaching platoon. Once the platoon is discovered, the RSU metering application computes the location of the rearmost vehicle of that platoon and begins to continuously calculate how long it would take the merge vehicle to reach the end of the merge area compared to how long it would take the rearmost vehicle of the platoon to reach that location. When the timing for both vehicles would allow the merge vehicle to join at a safe distance behind the rearmost platoon vehicle, the RSU metering application commands the merge vehicle to accelerate to the platoon speed. The command is continuously updated until the merge vehicle reaches a predefined merge point. Once the merge vehicle completes its acceleration and reaches the merge point, it is released from RSU metering control and allowed to engage in platooning operations.

DATA ANALYSIS AND RESULTS

Data were recorded from the vehicles and from infrastructure for use in data analysis. Figure 3 shows the vehicle speed during a typical automated run. The vehicles are listed in the legend according to their order in the platoon and the color of the vehicle. The vehicles leave the staging area together under manual control. "Cruising Control Begins" indicates the point at which the automated system takes over (~150 sec). During that time, the vehicles follow the speed limit associated with that point on the route. As the vehicles cross the virtual entrance to the managed lane (~215 sec), the platoon begins to form. "4 Vehicles Form Platoon" indicates the time at which each vehicle crosses this virtual entrance and begins the search for a platoon to join. At this point, there is a spike in vehicle speeds as the following vehicles close the gap with the lead vehicle. The vehicle speeds eventually stabilize as they approach the programmed headway, though there are still some variations in the speed, primarily due to the terrain. Prior to the merge point (~275 sec), the merge vehicle activates and it approaches the holding point under RSU control. As the four-vehicle platoon approaches the merge area, speed harmonization slows the speed of the platoon. "Reduced Speed Zone" indicates the time at which the reduced speeds go into effect. The Roadside Unit detects the approaching platoon and sends a command to the merge vehicle to release it from its holding point. "Merge Vehicle Released" indicates the time at which the merge vehicle begins to accelerate toward the mainline. In this case, the speed overshoots slightly as the merge vehicle aligns itself with the rear of the platoon. With the merge vehicle in place and at speed, a five-vehicle platoon is formed. The platoon travels an additional distance until it is dissolved, beginning with the rear vehicle, and the experiment ends.

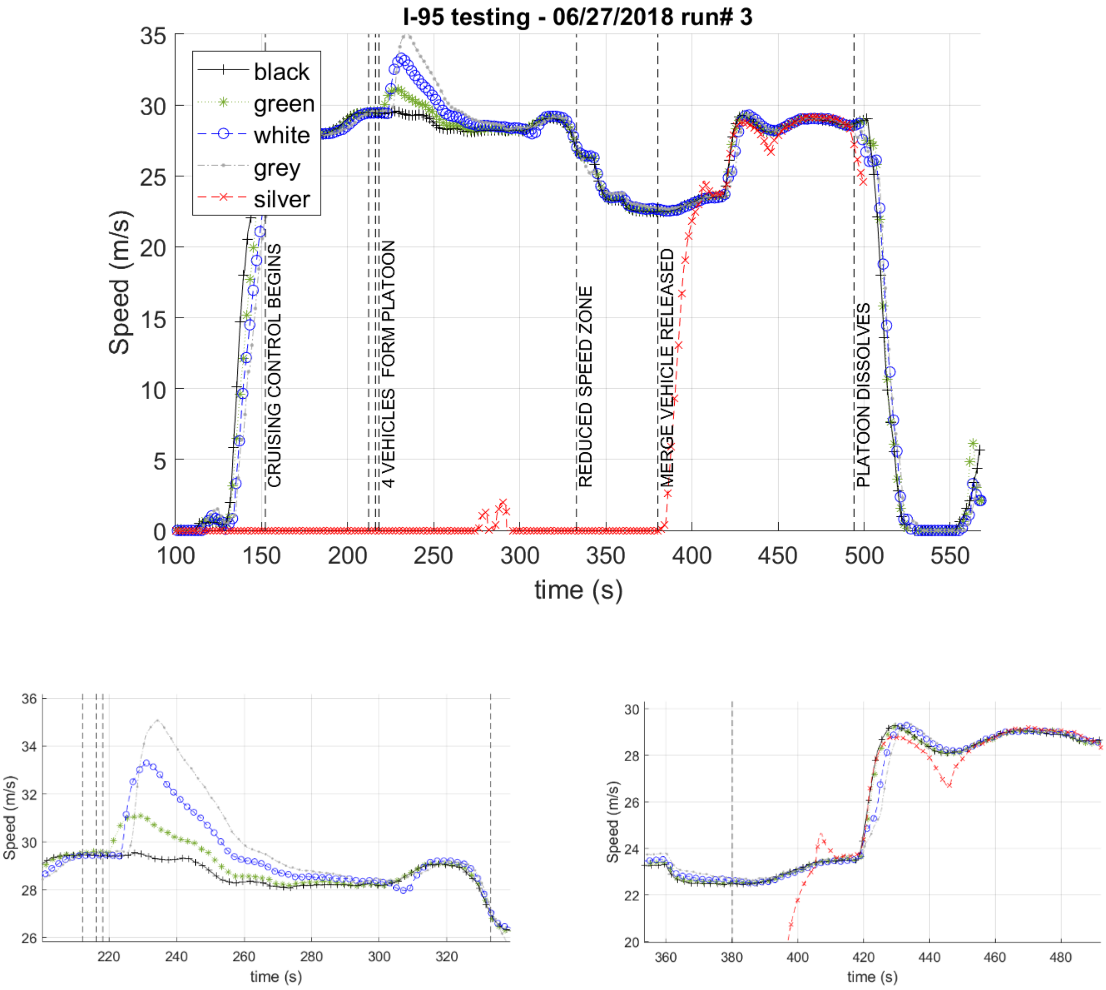
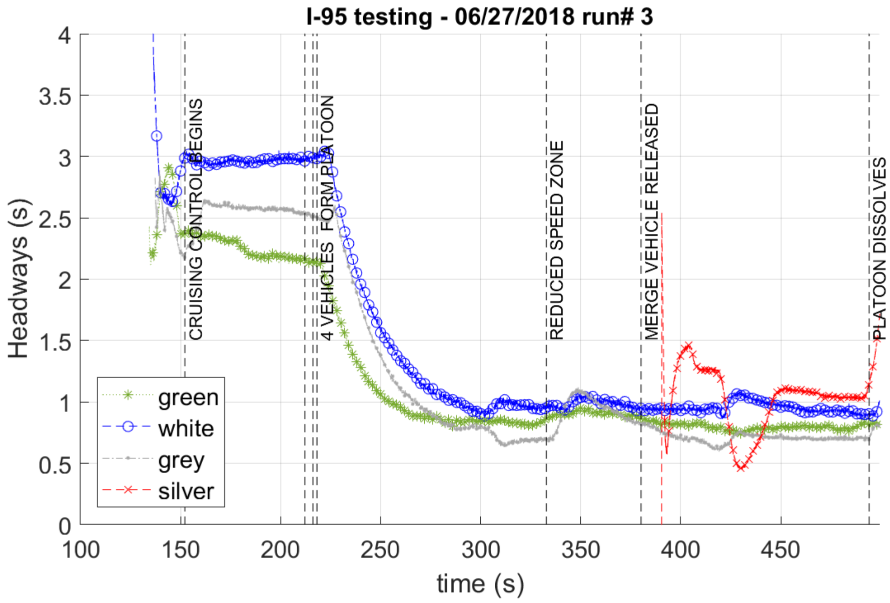
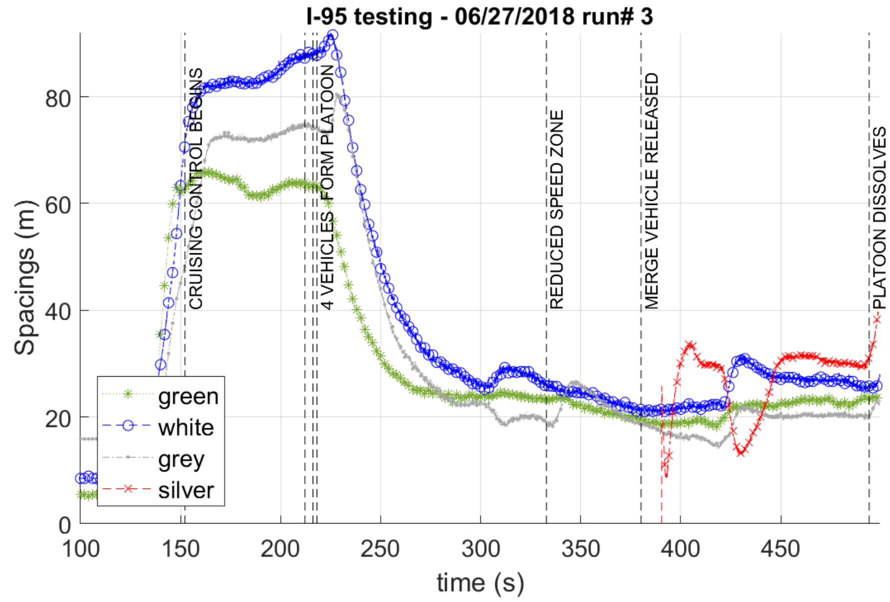


Figure 3. Vehicle speeds during a typical automated experiment.

Note that Figure 4 shows oscillations of the merge vehicle behavior, implying that part of the merge vehicle algorithm that controls the vehicle to catch up with the platoon is yet to be improved. The current algorithm represents simple adaptive cruise control approach that can be better tuned to reduce the oscillatory behavior, and we plan to complete it before the future experiments.



(a)



(b)

Figure 4. Vehicle headway (a) and spacing (b) during the initial platoon formation.

In the manual case, the lead vehicle follows a predetermined trajectory and each following driver regulates their own speed to maintain a specified gap. It is difficult for a driver to follow an exact time gap without overapplying brake and throttle, which can lead to unnatural driving behavior. To make the driving behavior more natural, drivers were told either be “aggressive,” “normal,” and “conservative.”, depending on the actual driving habit of the drivers. The drivers are also told to driver as natural as possible.

The merge vehicle was released manually based on the position of the platoon. When the platoon passed a predetermined point ahead of the merge area, the driver of the lead vehicle informed the merge vehicle by radio and the merge vehicle started accelerating down the ramp. This is meant to approximate a more traditional ramp meter. Just as in a normal non-automated driving environment, acceleration is not constant, and the alignment of the vehicles is not perfect.

Figure 5 shows the vehicle speeds during a manual run where the lead vehicle slows down in the merge area. In this case, the lead vehicle follows a pre-generated set of speed commands based on the output of the traffic simulation using ACC speed regulation. It can be seen the manually-driven vehicles have rather large fluctuations in their speed throughout the experiment. There is obviously amplification of oscillations, as seen from the speed profiles and grey and silver vehicles. It is noted that it is necessary to collect a large number of runs of data for human-driven trajectories because of the randomness the human behavior. However, due to the limited project resource, we were not able to conduct more runs, though the runs we conducted show the similar trends. Also, note that the oscillatory human-driven behavior may also be attributed to other reasons, such as how different drivers understand the experiment after they are instructed the same way and the driving environment (i.e., mostly late night in this experiment).

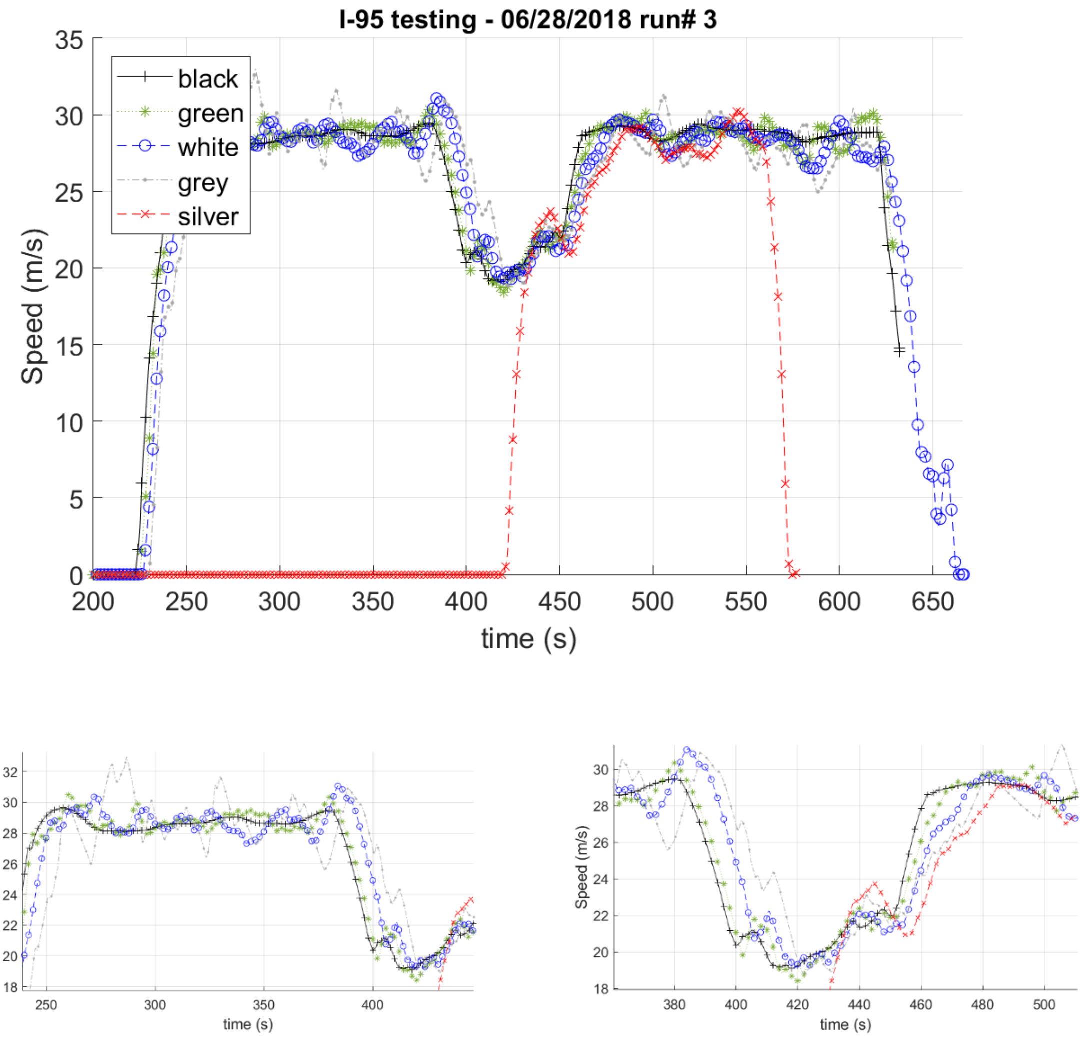
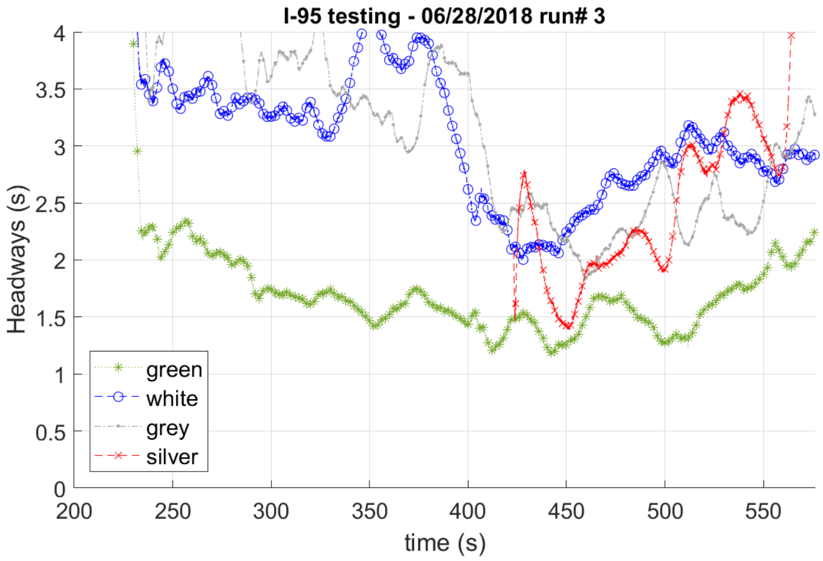
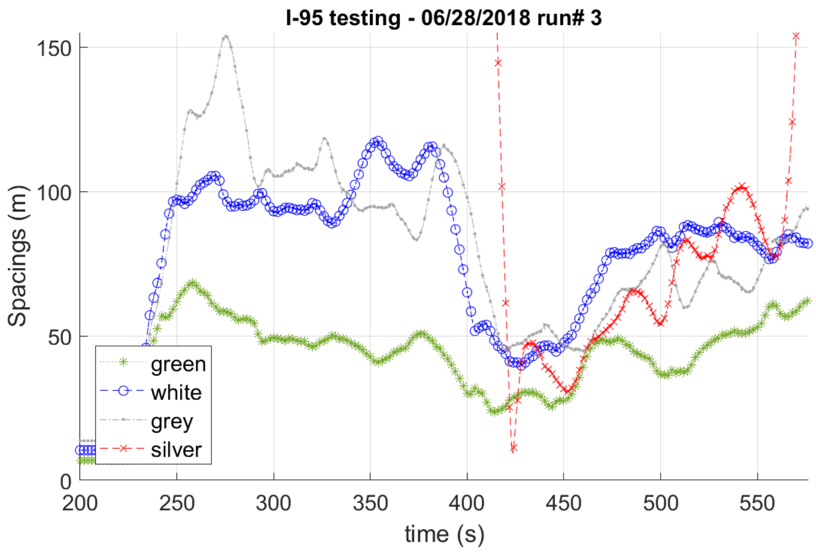


Figure 5. Vehicle speeds during a typical manual experiment in closed conditions.

Figure 6. shows the headways and spacings for the manual run. It can be seen around the merge point (~420 sec) as the vehicles slow down that the vehicle spacing reduces slightly. But as speeds increase, the spacing also increases. This is expected because drivers normally tend toward a constant time gap criterion in vehicle following.



(a)



(b)

Figure 6. Vehicle headway (a) and spacing (b) during a typical manual run.

Comparison and Analysis

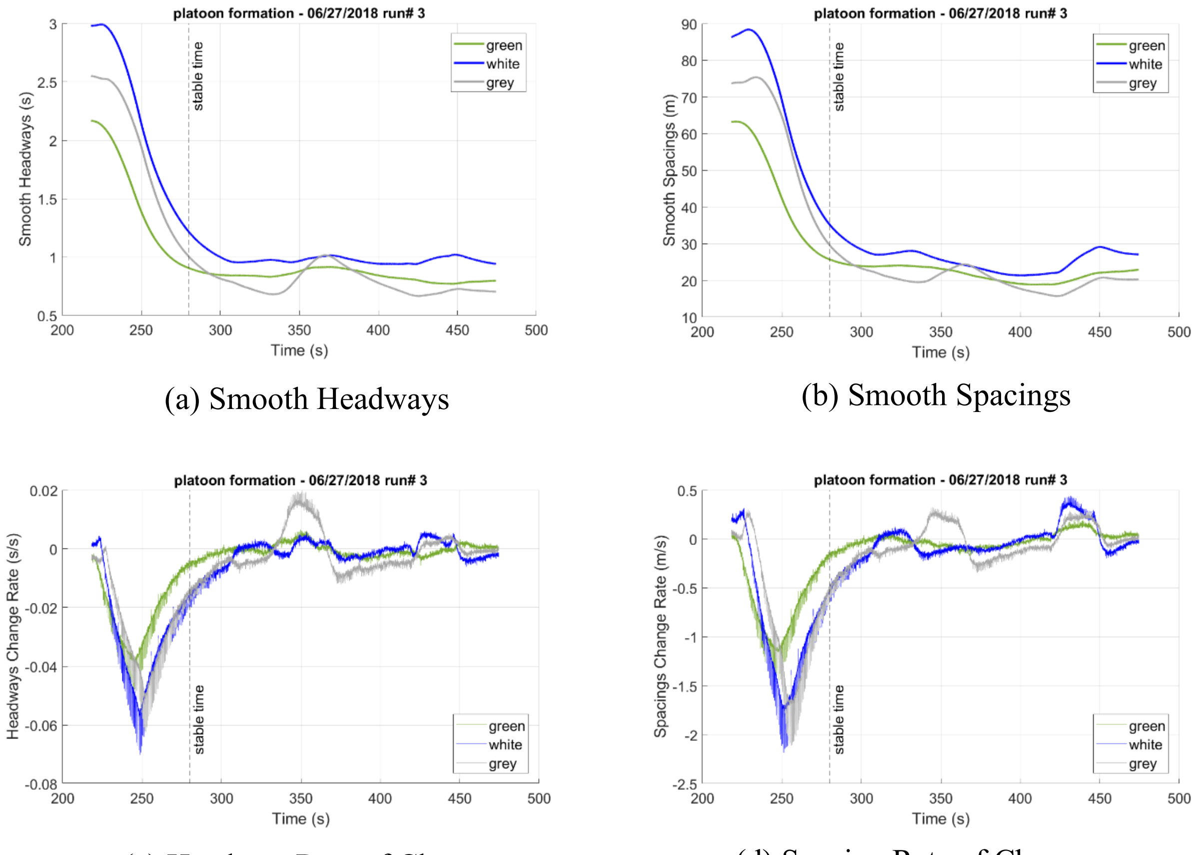
Platooning Analysis

According to the experimental observations, when a group of vehicles attempts to form into a platoon, a safe spacing level to the preceding vehicles must be reached and as a result, each vehicle will have to adjust its speed and spacing in coordination with other vehicles in the group. These adjustments impose some fluctuations in vehicle speed, headway, and spacing profiles until they reach the target spacing or headway with relatively small fluctuations. In this section, the period of time that is required for the vehicles to adjust their spacings and headways, denoted by T^{p}, is analyzed. For this, a set of criteria is defined to quantify the “stable” platoon and measure the adjustment period. These criteria basically investigate two factors: 1) analyzing the vehicle gaps to see if the vehicles are sufficiently close to each other, and 2) investigating the stability of their maneuvers.

Two criteria are considered in defining a stable platoon. First, in order to consider vehicles in a platoon, we require the vehicles to get sufficiently close to each other. Therefore, the first defined criterion states that the maximum absolute difference between the spacing and headway values should be less than the corresponding predetermined threshold parameters, denoted by $\Phi^{\mathrm{s}} = 10$ meters and $\Phi^{\mathrm{h}} = 0.4$ seconds, respectively. The second criterion requires the oscillations in spacing and headway profiles to be relatively dampened. For this, the criterion focuses on the rates of change of spacing and headway values and states that the maximum absolute spacing and headway rate of change values should be less than the corresponding predetermined threshold parameters, denoted by $\varphi^{\mathrm{s}} = 0.5$ meters per second and $\varphi^{\mathrm{h}} = 0.015$ seconds per second, respectively. With that, “stable time” is defined as the time at which all these requirements are met. Then, T^{p} is determined by calculating the difference between “stable time” and the start time of platoon formation (that is obtained from the vehicles’ log files).

For this analysis, four experiment runs are identified that contain relatively reliable data. With the assumed parameters, the results show an average value of 64 seconds for T^{p} in these experiments. Note that this outcome is sensitive to the assumed parameters, i.e., the Φ^{s}, Φ^{h}, φ^{s}, and φ^{h} parameters. Figure 7 shows the platooning analysis results for two experiment runs with the obtained “stable time” values. In these plots, the vehicle orders are the same as shown in the legends.

Note that in an ideal world, the spacings or headways should be approaching the preset values. However, in the experiment, many other factors are affecting the behavior of the vehicles. For example, the data, such as radar output or basic safety messages, that are fed into the CACC algorithm may have been compromised by various external factors (e.g., roadway curvature, communication quality). Also, the algorithm implemented in the vehicle (*15*) adds an extra buffer to vehicle gap regulation behavior and does not require the vehicles to exactly follow the set gap immediately, if a deviation occurs, to avoid fluctuation if some vehicles start to accelerate or decelerate. From this perspective, the data in Figure 7 shows the platoon headways and spaces are all approaching relatively stable states toward the second half of the graph.



(a) Smooth Headways (b) Smooth Spacings

(c) Headway Rate of Change (d) Spacing Rate of Change

Figure 7. Four figures showing headway and spacing characteristics during the initial platoon formation.

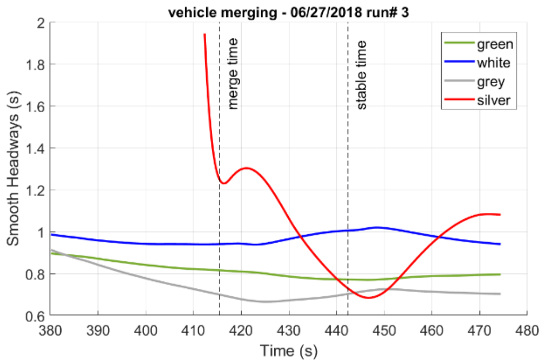
Merging Analysis

This maneuver consists of three periods of time. In the first period, the merging vehicle accelerates to catch up to the tail of the platoon. When the merging vehicle reaches the end of the platoon (the beginning of the second period), the platoon size increases by one, and the last vehicle in the platoon starts adjusting its speed and spacing to balance them according to the target values. In this period, the spacing and headway of the last vehicle may fluctuate until they reach to the stable levels. Finally, in the last period, the platoon proceeds in a stable state until the end of the experiment. In this analysis, the start time of the second and the third periods are estimated using the following analysis procedures.

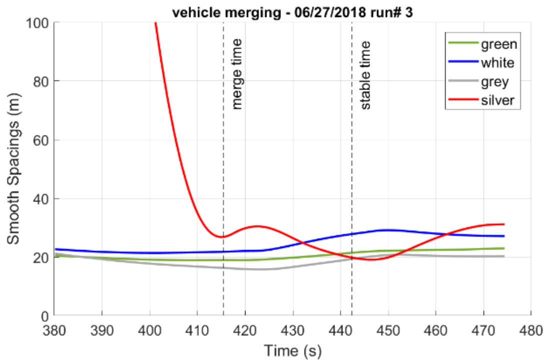
To estimate the arrival time of the merging vehicle to the tail of the platoon, it is assumed that the merging vehicle reaches a speed that is equal to or greater than that of the platoon. With this assumption, the spacing and headway of the merging vehicle both decrease during the first defined step. When the merging vehicle becomes sufficiently close to the platoon tail, the spacing and headway adjustments (the second step) begin and as a result, these values start fluctuating rather than strictly decreasing. With that being said, “merge time” is defined as the maximum time corresponding to first local minima of the smooth spacing and headway profiles of the merging vehicle (or when their rates of change reach positive values). Then, the time needed for the vehicle to merge into the platoon, denoted by T^{m}, is determined by calculating the difference between “merge time” and the release time of the vehicle, which is obtained from the vehicle’s log file.

To estimate the start of the third step, a process similar to that described in the platooning analysis section is implemented to identify the time when the merging vehicle headway becomes stable. In this analysis, we use the same criteria defined in the platooning analysis section with the same parameter set, i.e., Φ^{s}, Φ^{h}, φ^{s}, and φ^{h}. Again, all four requirements (i.e., the first criterion on spacing, the first criterion on headway, the second criterion on spacing, and the second criterion on headway) should be met.

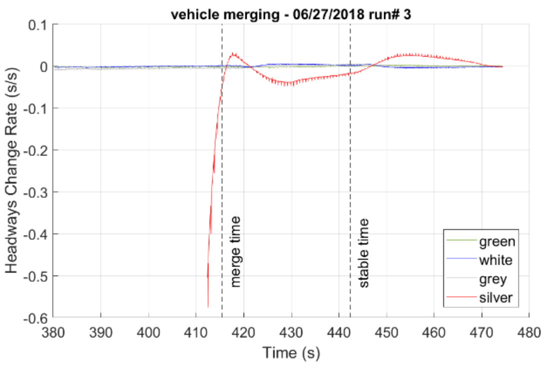
The same four experiment runs as the platooning analysis section are used for this analysis. With the assumed parameters, the results show average values of 35 and 64 seconds for T^{m} and T^{p}, respectively. The average T^{p} value obtained in this analysis is perfectly consistent with the platooning analysis results. Again, note that these outcomes are sensitive to the experimental settings. More specifically, T^{m} depends on the on-ramp length, the location of the merging vehicle staging area, and the speed of platoon at the merging area. Figure 8 show the merging analysis results for two experiment runs with the obtained "merge time" and "stable time" values. In these plots, the vehicle orders are the same as shown, where the plots of the merging vehicle (the silver vehicle) are shown in red.



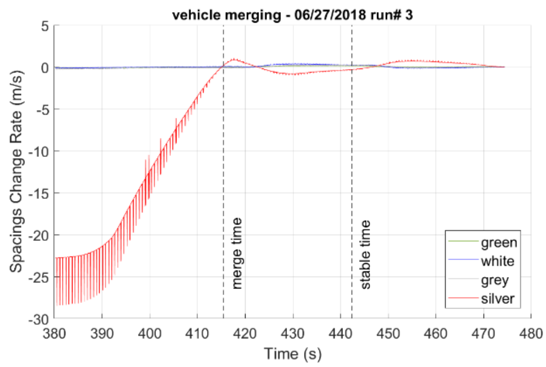
(a) Smooth Headways



(b) Smooth Spacings



(c) Headway Rate of Change

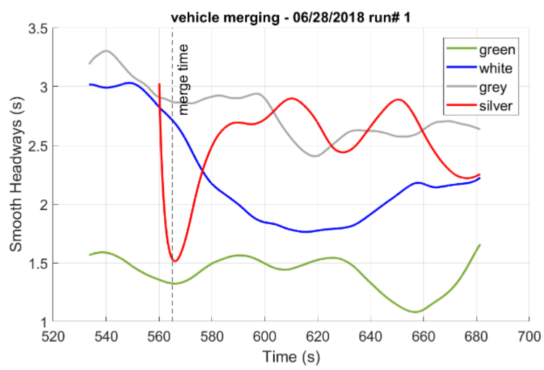


(d) Spacing Rate of Change

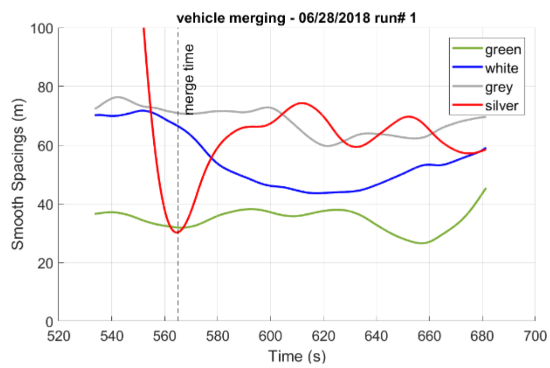
Figure 8. Four figures showing headway and spacing characteristics during a typical cooperative merge event.

The same merging analysis is also performed a manual experiment run. In the manual runs, neither of the two defined criteria are met during the experiments, and thus "stable time" is not reported. That is because the spacing and headway oscillations in these runs are significantly greater than in the automated runs, and thus the manual runs cannot satisfy the criteria according to the defined parameter set shown in Figure 9. With the assumed parameters, the results show an average value of

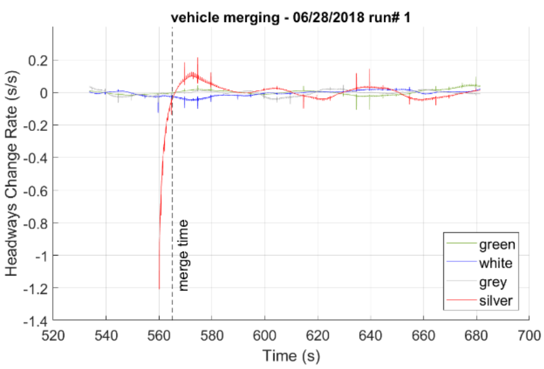
36 seconds for T^{m}. This result shows no significant difference between the T^{m} values of the automated and manual runs. That is probably because this time depends so heavily on the geometrical features of the experiment (e.g., the on-ramp length and location of the merging vehicle staging area) and the speed of platoon at the merging area, which are common in both automated and manual experiment runs. The acceleration of the merging vehicle is also similar in both cases since, in the automated case, it was selected for driver comfort, and in the manual case, the driver accelerated at a comfortable rate. Figure 9 shows the merging analysis results for a typical manual run with the obtained "merge time" values.



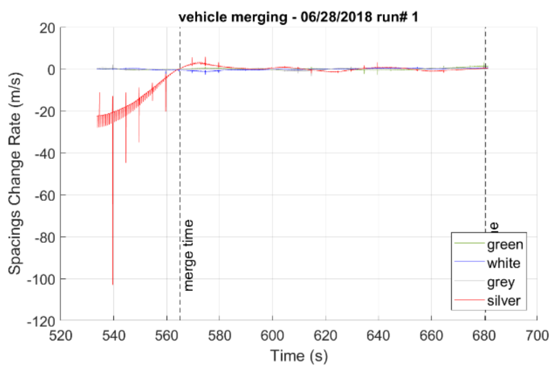
(a) Smooth Headways



(b) Smooth Spacings



(c) Headway Rate of Change



(d) Spacing Rate of Change

Figure 9. Four figures showing headway and spacing characteristics during a typical manual merge event.

Capacity Projection

Through the use of the experimental dataset, it is also possible to do extrapolation to calculate the capacity enhancement due to the bundled application. Note that this is only a projection of capacity extrapolated from the one-platoon experimental data using various assumptions. The purpose is to shed light on capacity enhancement using the real-world data we collected. The results, as discussed later, are also comparable to the traffic simulation analysis for the same bundled application.

First, we calculate the CACC pipeline capacity. We calculate the average time headway (or gap) between adjacent vehicles during experimental runs. We also assume other system variables for the analysis, such as platoon size of 10 and inter-platoon gaps of 1.5 seconds, similar to our simulation study (*16*). Second, we derive the merge area capacity (measured downstream of the merge area). Similar to the pipeline capacity above, we can use the experimental data to obtain the required gap for the merge vehicle from its preceding vehicle (across experimental runs). As shown in the example data above, we can see the actual headway between the merge vehicle and the fourth vehicle (the merge vehicle is fifth) varies from 0.5 to 1.5 seconds (mostly at the higher end). Therefore, we can assume that the merge vehicle will need 1.5 seconds of headway for the merge maneuver at the merge area. While the platoon may become stable and operating with smaller gaps after the merge, the capacity can be assumed to be constrained by the merge area: in a 10-vehicle platoon, the lead vehicle follows the inter-platoon headway of 1.5 seconds and the following 8 vehicles use the intra-platoon headway of 0.84 (average measurements from the experimental data) and the last (merging) vehicle uses 1.5 seconds of the intra-platoon gap. Then the capacity reduces from 3973 vphpl to 3703 vphpl, a 7% decrease. This reduction is attributed to the cooperative merge with back-join during which merge vehicles look for inter-platoon gaps that are large enough for a vehicle to merge into. With the assumption of 0.84 and 1.5 for the intra- and inter-platoon headways, respectively, the results in Table 1 are calculated. For example, the 10-vehicle platoon pipeline capacity = 10 vehicles * 3600 seconds / (9 vehicles * 0.84 second + 1 vehicle * 1.5 seconds) = 10 * 3600/(9.06) = 3973 vphpl. Note that although platoon size of 15 seems to be better than the size of 10, the allowed merge volume (on-ramp volume) actually decreases.

Table 1. Capacity analysis based on field experimental data.

Platoon Size	Human Capacity* (vphpl)	Pipeline Capacity (vphpl)	Improvement of Throughput (%)	Capacity w/ Merge (vphpl)	Reduction from Capacity w/ Merge (%)	Improvement of Throughput w/ Merge (%)
Average Intraplatton Headway = 0.84 second (Gap = ~0.6 second)						
5	2350	3704	57.6	3261	12.0	38.8
10	2350	3973	69.1	3703	6.8	57.6
15	2350	4072	73.3	3879	4.7	65.0
Average Gap = 0.94 second (Gap = ~0.7 second)						
5	2350	3422	45.6	3092	9.6	31.6
10	2350	3614	53.8	3422	5.3	45.6
15	2350	3683	56.7	3548	4.7	51.0

* human-driven traffic capacity derived using simulation in (*16*)

In the simulation studies for the same application by the research team (*16*), the intra-platoon gap follows a distribution: 0.6 second for 57 percent of the time they were car following, 0.7 second for 24 percent of the time, 0.9 second for 7 percent of the time, and 1.1 seconds for 12 percent. The weighted average of the gap is 0.7 second. The results in Table 1 when the intra-platoon gap is 0.7 seconds can be compared to the simulation case with 100% penetration rate. The pipeline capacity is about 3300 vphpl in simulation (*16*). Data from the field experiment indicate a range of 3422 to 3683 vphpl depending on the length of the platoon. The theoretical results based on experimental data are a little higher than the simulation value considering that not all platoons will be 10 vehicles in real simulation (due to the distribution of desired gaps and traffic arrival patterns) and there may be some traffic oscillations that can reduce the throughout to some extent. It is also the same case with the theoretical estimation of capacity with cooperative merge and the difference can also be attributed to the disturbance and oscillation of traffic at the merge area. With that in mind, along with the error introduced by a small number of vehicles involved in the field experiment vs. the simulation, the pipeline capacity and capacity with cooperative merge in the two cases compare favorably.

CONCLUSIONS

In this study, we propose a bundled CAV application that involves platoons of equipped vehicles governed by an integrated set of cooperative adaptive cruise control (CACC), cooperative merge, and speed harmonization applications. Building upon existing vehicle platforms and embedded algorithms (*6*), we implemented the bundled application in a fleet of five vehicles at the FHWA Saxton Transportation Operations Lab. Experiments were conducted to collect and compare data on CAV and human-driven behavior. Based on the experimental data, our results show that the performance of the CAV operations, including platooning and cooperative merging under varying I2V speed commands, show string stability, superior to human-driven platoons in our experiment.

The experiment serves as one of the first CAV experiments that occurs not only with physical CAVs but also interacts with physical roadway infrastructure. Throughout the paper, we have discussed challenges and lessons learned during our implementation. We found that the actual vehicle performance in platooning and merging are quite different from the simulation literature, because of additional factors that may impact the platoon performance, such as roadway curvature and communication quality. We also pointed out additional insights, such as "stable time", that captures realistic CAV operational components. Despite all the challenges of implementing the experiments, we also concluded with encouraging results when we discuss the platooning and merging performance and projection of roadway capacity enhancement. We believe future experiments with onboard algorithms that explicitly consider real-world operations will create even larger benefits to the traffic systems.

Further experiments should be conducted to collect data on lateral behavior. Data for CAV operations along with human-driven vehicles are also of interest. It is also important to collect some additional longitudinal and lateral behavioral data for CAV car following and lane changing model calibration and validation. Also, this paper only demonstrates a single platoon operation. Some techniques, such as hardware-in-the-loop simulation testing (*20, 21*) can be adopted in future studies to incorporate the impact of surrounding traffic conditions.

ACKNOWLEDGMENT

This study is funded in part by the U.S. Department of Transportation (Project Number: DTFH61-12-D-00020), National Science Foundation CMMI # 1901998, and the University of Cincinnati Office of Research. The authors want to thank a few other team members for their contributions to the field experiment and paper manuscript: Kyle Rush, John Stark, Steven Shladover, Xiao-Yun LU, Robert Ferlis, Fang Zhou, Shuwei Qiang, and Michael McConnel. The work presented in this paper remains the sole responsibility of the authors.

DATA AVAILABILITY STATEMENT

Some or all data, models, or code generated or used during the study are available from the corresponding author by request (data analysis codes and a portion of the experimental data).

REFERENCES

1. Nowakowski, C., J. O'Connell, S.E. Shladover, and D. Cody, 2010, "Cooperative Adaptive Cruise Control: Driver Selection of Car-Following Gap Settings Less Than One Second", 54th Annual Human Factors and Ergonomics Society Meeting, San Francisco, CA.

2. Nowakowski, C., O'Connell, J., Shladover, S., and Cody, D. Cooperative Adaptive Cruise Control: Driver Acceptance of Following Gap Settings Less Than One Second. Proceedings of the Human Factors and Ergonomics Society 54th Annual Meeting, 54(23), pp. 2033-2037, San Francisco, CA, September 27 - October 1, 2010.

3. Shladover, S.E., Su, D., and Lu, X.-Y. Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow. Transportation Research Record No. 2324, pp. 63-70, 2012

4. Milanés, V., Shladover, S.E., 2014. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. Transportation Research Part C: Emerging Technologies 48, 285-300.

5. Shladover, S., C. Nowakowski, X. Lu, and R. Ferlis, "Cooperative Adaptive Cruise Control (CACC) Definitions and Operating Concepts," Transportation Research Record: Journal of the Transportation Research Board, No. 2489, TRB, National Research Council, Washington, D.C., 2015, pp. 145-162 . http://dx.doi.org/10.3141/2489-17

6. Raboy, K., Ma, J., Stark, J., Zhou, F., Rush, K., & Leslie, E. (2017). Cooperative Control for Lane Change Maneuvers with Connected Automated Vehicles: A Field Experiment (No. 17-05142). Transportation Research Board Annual Meeting.

7. Chou, F.C., Shladover, S.E. and Bansal, G., 2016, December. Coordinated merge control based on V2V communication. In Vehicular Networking Conference (VNC), 2016 IEEE (pp. 1-8). IEEE.

8. Federal Highway Administration, Synthesis of Active Traffic Management Experiences in Europe and the United States, FHWA-HOP-10-031, 2010.

9. Talebpour, A., Mahmassani, H.S., and Hamdar, S.H. (2013). "Speed Harmonization: Evaluation of Effectiveness Under Congested Conditions," Transportation Research Record 2391, 69–79, Transportation Research Board, Washington, DC.

10. Ghiasi, A., J. Ma, F. Zhou, and X. Li. Speed Harmonization Algorithm using Connected Autonomous Vehicles. Presented at 96th Annual Meeting of the Transportation Research Board, Washington, D.C., 2017

11. Ma, J., Li, X., Shladover, S., Rakha, H.A., Lu, X.Y., Jagannathan, R. and Dailey, D.J., 2016. Freeway speed harmonization. IEEE Transactions on Intelligent Vehicles, 1(1), pp.78-89.

12. Ma, J. & Leslie, E. (2018). Managed Lanes for Early Deployment of Connected and Automated Vehicle Applications: Concept of Operations. Submitted to Transportation Research Board Annual Meeting, Washington, D.C.

13. Liu, H., Kan, X., Shladover, S. and LU, X. 2017. Modeling impacts of Cooperative Adaptive Cruise Control on mixed traffic flow in multi-lane freeway facilities. Transportation Research Part C, 95 (2018) 261–279

14. Ma, J., Y. Guo, and E. Leslie. Managed Lanes for Early Deployment of Connected and Automated Vehicle Applications: Concept of Operations. Submitted to Transportation Research Board Annual Meeting, Washington, D.C., 2018.

15. Bujanovic, Pavle, Taylor Lochrane, Jia Hu, Tomislav Bujanovic, and C. Michael Walton. Cooperative Adaptive Cruise Control Algorithm with Priority Weights Assigned to Downstream Vehicles for Increased Safety. No. 18-04157. 2018.

16. Guo, Yi., J. Ma, and E. Leslie. Evaluating The Effectiveness of Bundled Connected Automated Vehicle Applications Applied to Freeway Managed Lanes. Transportation Research Board Annual Meeting, Washington, D.C., 2018.

17. Scarinci, R., Hegyi, A., & Heydecker, B. (2017). Definition of a merging assistant strategy using intelligent vehicles. Transportation research part C: emerging technologies, 82, 161-179.

18. Ma, J., Li, X., Shladover, S., Rakha, H. A., Lu, X. Y., Jagannathan, R., & Dailey, D. J. (2016). Freeway speed harmonization. IEEE Transactions on Intelligent Vehicles, 1(1), 78-89.

19. Learn, S., Ma, J., Raboy, K., Zhou, F., & Guo, Y. (2017). Freeway speed harmonisation experiment using connected and automated vehicles. IET Intelligent Transport Systems, 12(5), 319-326.

20. Ma, J., Zhou, F., Huang, Z., Melson, C. L., James, R., & Zhang, X. (2018). Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for queue-aware signalized intersection approach and departure. Transportation Research Record, 2672(22), 36-46.

21. Ma, J., Zhou, F., Huang, Z., & James, R. (2018, November). Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for cooperative adaptive cruise control. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 2878-2883). IEEE.