Automated platoon manipulation in merging scenarios using trajectory estimation of connected vehicles

Rory Alston[a], Daniel Pierce[a], Patrick Smith[a], Robert Lyle[a], David Bevly[a], Scott Heim[b]

[a]Mechanical Engineering, 1418 Wiggins Hall, Auburn University, AL, USA 36849; [b]U.S. Army Combat Capabilities Development Command (CCDC) Ground Vehicles Systems Center (GVSC), Warren, MI, USA.

ABSTRACT

A method of integrating non-automated vehicles into a platoon of Cooperative Adaptive Cruise Control (CACC) Class 8 Heavy Duty Trucks was developed and tested at the National Center for Asphalt Technology (NCAT) test track. As CACC platooning becomes more commonplace, the interaction between automated and non-automated vehicles increases. This work explores this interaction during a highway merging scenario. Merging of vehicles occurs when a vehicle enters the highway system where an automated platoon is operating at that time. This scenario is handled by predicting the trajectories of the automated platoon and the non-automated vehicle entering the highway. Based on these trajectories, the automated platoon decides to create a gap for the merging vehicle if necessary. In this work, the non-automated vehicle is assumed to be a passive, connected vehicle, i.e. the platoon has knowledge of the merging vehicle's position and velocity. The merging scenarios are handled by a CACC platooning system with Dedicated Short Range Communication (DSRC) for Vehicle to Vehicle (V2V) communication. This algorithm was implemented as an addition to Auburn University's CACC system and was able to successfully complete automated merging scenarios. The merging scenarios in the front, middle, and back of the platoon were completed at the NCAT test track.

Keywords: Merging, CACC, V2V, Automated Platooning, Connected Vehicles, Trajectory Estimation

1. INTRODUCTION

Automated truck platooning systems are an increasingly popular area of research due to the increased safety and fuel savings for each vehicle. Specifically, the platooning vehicles often are able to communicate information and act cooperatively, like the four vehicle platoon shown in Figure 1. Furthermore, automated merging is of great interest due to the high number of accidents that occur each year. Merging is defined as when two roadways come together to form one or a single roadway reduces its number of lanes. Each year, approximately 36% of all freeway accidents occurred on highway entrance ramps and between 20-30% of all trucking accidents occurred on or near entrance ramps as well. [1,2] Automating the merging scenario provides a great opportunity for improving overall highway safety.

Figure 1. Auburn CACC system on Highway I69 with four vehicle platoon.

A number of researchers have looked into formulating algorithms for lane change models intended for the use in traffic simulations. For example, the MOBIL lane change model introduced in [3] . Despite the large amount of research available, automated merging algorithms are only recently being developed. A number of different types of algorithms exist and typically fall into two categories: Vehicle to Vehicle (V2V) communication and Infrastructure to Vehicle (I2V) communication. Some examples of V2V algorithms are provided by [4,5,6] . The algorithm presented in [4] utilizes GPS data and Dedicated Short Range Communication (DSRC) to determine what order each vehicle should merge. [5] and [6] present protocols to determine merge order in cases where lane change maneuvers are necessary, for example when a road narrows due to a decrease in available lanes. An example of an I2V algorithm is provided by [7] , which explores a way to optimize fuel consumption and travel time with a closed form solution by using a First-In First-Out (FIFO) logic to decide merge order.

The goal of this paper is to design and implement an algorithm for merging scenarios using GPS measurements and DSRC communication for the connected, merging vehicle. Specifically, the algorithm uses trajectory estimation and FIFO logic similar to [7]. First, the necessary equations are presented to represent the merging scenario. Then, the FIFO logic used for the decision making is explored in a MATLAB simulation to check its validity for a real merging scenario. Next, the algorithm was implemented into Auburn University's Cooperative Adaptive Cruise Control (CACC) platooning system. The CACC system is then described as it is implemented on two Class 8 Peterbilt 579 trucks. A separate "plug and play" device allows any vehicle to be a connected vehicle, i.e. the vehicle is able to communicate its GPS state information. The merging algorithm was tested with a two vehicle truck platoon and a merging vehicle at the National Center for Asphalt Technology (NCAT) test track. The results and analysis from these tests are presented to show the effectiveness of the designed system.

2. BACKGROUND

The FIFO logic described in [7] operates by taking an initial queue of vehicles in the order at which their information is received. This queue order is then optimized by a controller that minimizes the overall fuel consumption of each vehicle, while also avoiding collisions. The FIFO control logic presented in this paper differs by using a simplified merge point, rather than a defined zone as in [7] . Furthermore, this approach does not consider constraints or optimizations for the merging order solution. More importantly, this version of FIFO logic creates the merge order based on which vehicle is estimated to get to the defined merge point first, as opposed to when a vehicle's information is received. In summary, this FIFO control logic requires each vehicle's estimated travel times to a set merge point and then decides a new order based on which vehicle has the shortest time. For example, if the merging vehicle has a shorter time in comparison to the platoon lead truck, the merge vehicle will go first.

A number of ramp geometries exist, all of which may have varying ramp curvatures and lengths depending on the surrounding environment. To simplify the problem, a straight line trajectory was assumed for each vehicle since, as stated in [7], merging ramps tend to straighten out near the end as they combine with the highway. In this description, the merge point is defined as the first common point of the merging ramp and the highway. From this, the problem can be formulated using a scalar distance of each vehicle to the merge point (d).

2.1 Trajectory Estimation

In order to estimate the time to a defined merge point, the motion of each vehicle must be described. One of the simplest ways to do so is using kinematic equations. The distance to the merge point (d) is modeled as a function of time (t) given initial conditions on distance (d_0) and speed (v_0) is

$$d(t) = d_0 - v_0 t - \frac{1}{2} a t^2 \ . \tag{1}$$

For vehicles traveling at constant speed, Equation (1) simplifies with $a = 0$ to

$$d(t) = d_0 - v_0 t \ . \tag{2}$$

From this, the time required to reach the merge point is calculated by solving Equation (2) with $d(t) = 0$ as

$$\hat{t}_k = \frac{d_k}{v_k} \ . \tag{3}$$

In Equation (3), d_k is the current distance to the merge point, v_k is the current speed, and $\hat{t}_k$ is the estimated time to merge point. For platoon vehicles, the time estimate from Equation (3) is sufficient by assuming each vehicle's velocity is nominally constant at steady state operation. However, this time estimate does not suffice for the merging vehicle since the vehicle is typically accelerating to match the higher highway speed limits. As such, it is necessary to include the vehicle's acceleration when calculating its time estimate. The motion profile of a merging vehicle is assumed to be constant acceleration until a.) the merge point is reached ($d = 0$) or b.) the speed limit is reached ($v = v_{max}$). An example of this motion profile is shown in Figure 2.

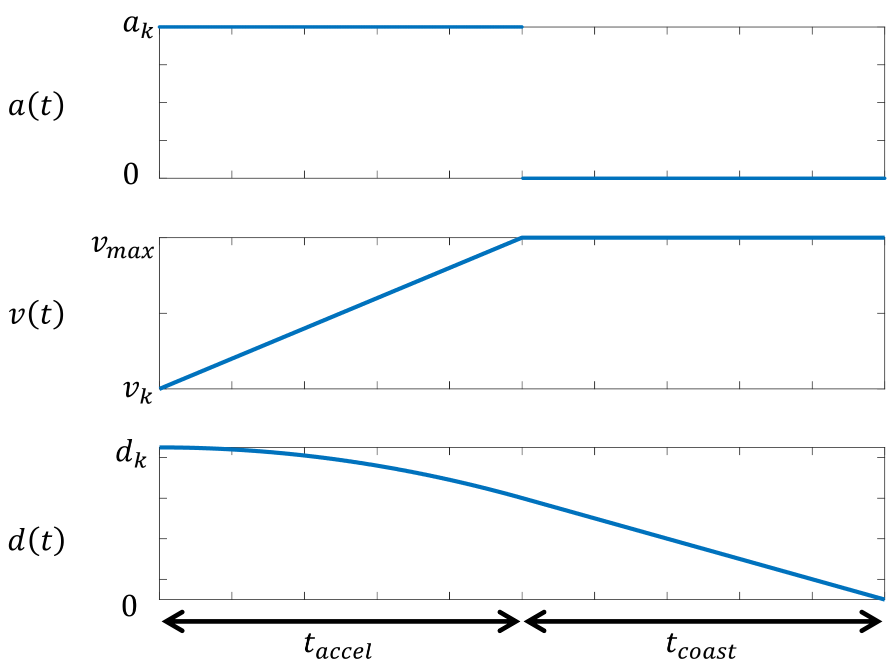

Figure 2. Trajectory prediction profile of merging vehicle.

For an accelerating merge vehicle, the first step in estimating the time to merge point is to calculate the time required to reach max speed, t_{accel}. This calculation uses the current speed (v_k), the speed limit (v_{max}), and current acceleration (a_k) as shown in Equation (4).

$$t_{accel} = \frac{v_{max} - v_k}{a_k} \tag{4}$$

The next step is to calculate the remaining distance to the merge point, d_{accel}. This is done by plugging t_{accel} into Equation (1) and solving for distance as

$$d_{accel} = d(t = t_{accel}) = d_k - v_k t_{accel} - \frac{1}{2} a_k t_{accel}^2 \ . \tag{5}$$

If the result of Equation (5) is negative ($d_{accel} < 0$), then the vehicle is expected to reach the merge point during the acceleration period. In this case, the time to the merge point is calculated by setting $d(t) = 0$ in Equation (1) and solving for time as

$$\hat{t}_k = \frac{-v_k + \sqrt{v_k^2 + 2a_k d_k}}{a_k} \ . \tag{6}$$

Otherwise, if there is still distance remaining after the acceleration period ($d_{accel} > 0$), the constant speed equation is used to calculate the remaining time. The time elapsed over the remaining constant velocity period (t_{coast}) is calculated as

$$t_{coast} = \frac{d_{accel}}{v_{max}} \ . \tag{7}$$

The total time to the merge point is then calculated by combining the two times as shown below.

$$\hat{t}_k = t_{accel} + t_{coast} \tag{8}$$

2.2 FIFO Logic

The merging scenario defined for this work is a two vehicle CACC platoon traveling along the right lane of a highway and a single merging vehicle traveling on the oncoming ramp that needs to merge onto the highway with the platoon. The FIFO algorithm presented in Figure 3 runs on the follower truck of the platoon, and simply listens for the position and velocity information of the merging vehicle over the DSRC radio network. The follower vehicle also uses the position and velocity information of the platoon lead already being passed over the DSRC radio network for CACC operation. Once this information is received, the algorithm determines if the merging vehicle is still accelerating by checking if v_{merge}, the velocity of the merging vehicle, is less than the speed limit, v_{max} . If the merging vehicle is still accelerating, its acceleration, $\hat{a}_k$, is estimated using Equation (9), described in the following section. Afterwards, the distance (d_{lead}, d_{follow}, d_{merge}) and time to the merge point, $\hat{t}_k$, for each vehicle is calculated as described in the previous section. These time estimates are compared to a new parameter $t_{decision}$, which determines how far out from the merge point the decision is made. For example, if $t_{decision}$ is set to 4 seconds, then this algorithm will update the measurements and estimates at each iteration until the merging vehicle is estimated to be below the set $t_{decision}$. Once this occurs, the algorithm takes the latest time estimates for each vehicle, and compares them using the FIFO logic to decide the new merging order. The overall architecture of the FIFO algorithm will operate as summarized below in Figure 3.

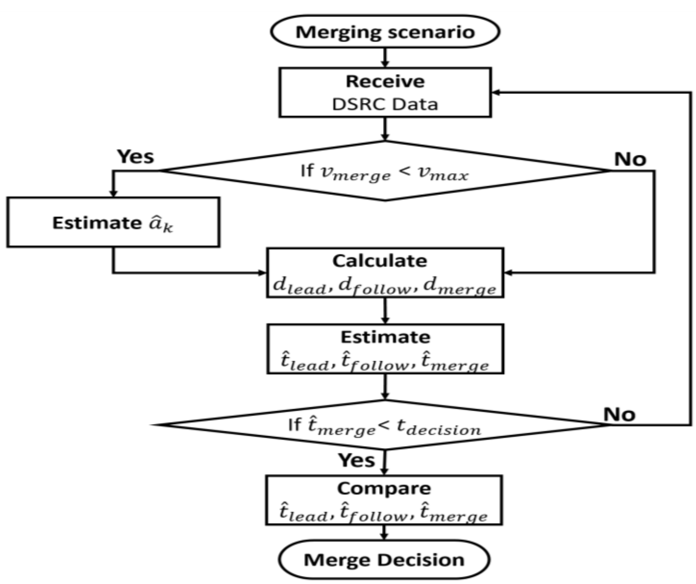

Figure 3. Merge algorithm architecture.

2.3 Acceleration Estimation

In practice, GPS is used to obtain measurements of distance ($\tilde{d}_k$) and speed ($\tilde{v}_k$). The distance measurement, $\tilde{d}_k$, is obtained by differencing GPS positions of the vehicle from the predefined merge point position. Similarly, $\tilde{v}_k$ is easily measured as the magnitude of the GPS velocity. Additionally, the previously described algorithm for trajectory estimation requires knowledge of the merge vehicle's acceleration (a_k). In the proposed setup, this acceleration cannot be directly measured, and must be estimated with the given measurements. Assuming the acceleration changes linearly with time, the acceleration is estimated as

$$\hat{a}_k = \frac{\tilde{v}_k - \tilde{v}_{k-1}}{t_k - t_{k-1}} \tag{9}$$

where $\tilde{v}_k$ and $\tilde{v}_{k-1}$ are the GPS velocity measurements at time t_k and t_{k-1}, respectively.

A number of acceleration tests were performed on a straight section of highway to confirm the accuracy of the estimate presented in Equation (9). Measurements of GPS position and velocity were taken using a Novatel GPS receiver, which was mounted to the roof of a Nissan G35. Additionally, measurements of acceleration were taken using a Crossbow 440 IMU mounted near the CG of the vehicle. Each run was performed by beginning at a complete stop, and then accelerating up to ~ 35 mph. Results from one of these tests is presented below in Figure 4.

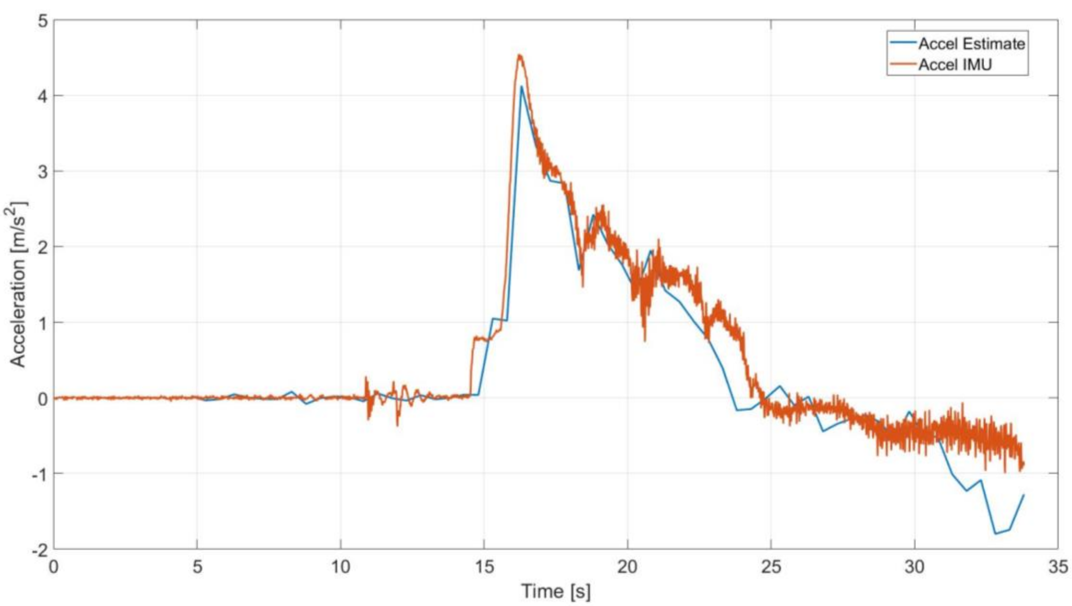

Figure 4. GPS estimated and IMU accelerations.

The estimated acceleration follows the general trend of the measured acceleration from the IMU, with a maximum error of ~1 m/s^2 between the estimate and the measured acceleration. This test shows that the acceleration estimate is adequate to use for the previously unknown merging vehicle's acceleration.

3. SIMULATION

A merging scenario simulation was built using MATLAB in order to initially test the previously described approach. Specifically, the straight line trajectory assumption and the FIFO logic were tested to check for any unsafe scenarios that may occur. GPS position data from the American Center for Mobility (ACM) test track was used as the merging ramp scenario for simulation runs. A map of the track is provided below in Figure 5.

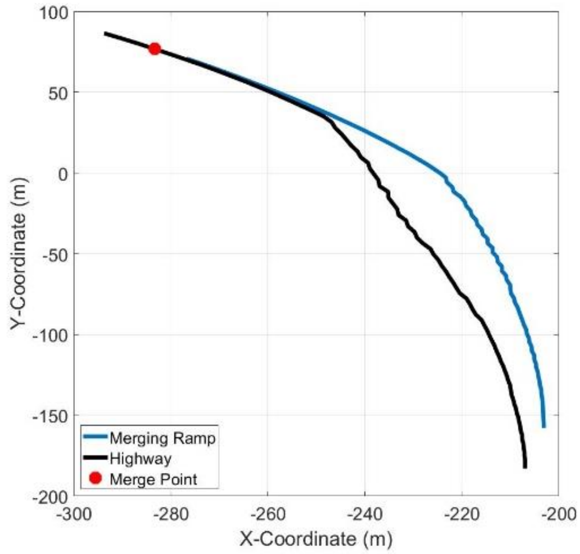

Figure 5. GPS position data for ACM test track.

A four vehicle platoon was created as point masses in the environment, each with a given position, speed, allowed deceleration rate, and set spacing of 15.24 m (50 ft). Similarly, a merging vehicle was created as a point mass and given a starting position, speed, and acceleration rate. The vehicles followed the predefined GPS path with their given velocity and acceleration profiles, and Equations (3) and (8) are used to estimate the time for each vehicle. These times were continuously calculated until the merging vehicle fell within the decision time parameter, at which point the most recent state of vehicles was taken to finalize the decision and initiate the merge maneuver. A number of scenarios were simulated at a nominal speed limit of 24.59 m/s (55 mph). Additionally, speed variation, acceleration, and starting position of the merging vehicle were changed to study the outcomes.

One issue that arose during simulation testing was the risk of rear-end collisions. In some scenarios it is possible for the merging vehicle time estimate to be extremely close to the estimate of another platooning vehicle, which can cause vehicles to end up dangerously close to each other, under .61 m (2 ft) in some cases. As a result, a set time cushion was created in order to prevent vehicles from merging too close to each other. This time cushion ($t_{cushion}$) is calculated with the following formula

$$t_{cushion} = \frac{x_{desired}}{v_{max}} \tag{10}$$

where $x_{desired}$ is the minimum safe distance between vehicles and v_{max} is the set speed limit. The time cushion was then implemented into the FIFO control logic, such that for a vehicle to be put ahead in the merge order, the following equation must be true

$$\hat{t}_{vehicle_1} < \hat{t}_{vehicle_2} - t_{cushion} \tag{11}$$

where $\hat{t}_{vehicle}$ is the estimated time calculated from Equations (3) or (8), and $t_{cushion}$ is the time cushion calculated from Equation (10). This ideally prevents any rear end collisions because it forces vehicles to merge in front of another only if they end up with a safe gap between them. After implementing the time cushion, no other unsafe situations arose in simulation. Also, selection of the decision time parameter is important to allow the platooning vehicles enough time to open a gap for the merge vehicle. In other words, if the platooning vehicle takes longer to slow down and create space for the oncoming vehicle, the decision time must increase to allow more time for the vehicle to react.

4. EXPERIMENTAL SETUP

After simulation, the algorithm was implemented for real time testing as an addition to Auburn University's CACC platooning system. As stated previously, the algorithm runs on the follower vehicle of the platoon and simply listens for the merging vehicle's position and velocity information over the DSRC radio network. A separate plug and play device, termed the Mergebox, was developed to facilitate a connected merging vehicle. The Mergebox was equipped with an onboard computer, a Novatel GPS Receiver, and a Cohda Wireless MK5 DSRC radio, which allow the platooning vehicles to receive the merging vehicle's position and velocity needed for the algorithm. For testing, the Mergebox was placed in a Lincoln MKZ that was manually driven. The algorithm was tested on the straightaways of the NCAT test track, where a predefined ramp length of approximately 168 meters was mapped onto the test track to serve as the length of a highway merging ramp. A top view photo of this scenario was taken using a drone and is provided in Figure 6 as a visual reference. The platoon was set to run at a constant 15.56 m/s (35 mph) and a set following distance of 30.5 m (100 ft). The merging vehicle began each run at a complete stop and accelerated up to 15.56 m/s before reaching the merge point at the end of the ramp.

Figure 6. Bird's eye view of defined merge ramp at NCAT.

The time estimates have inherit error from both the measurements and estimations used in the calculation. Therefore, the relative positions of the platoon vehicles with respect to the merge vehicle is used for reference. The relative position between a platoon and merging vehicle was found using differential GPS methods. Differential GPS gives centimeter-level relative accuracy due to cancellation of common errors in each signal, thus providing an accurate reference of the true merging scenario that occurred. The GPS course angle was used to resolve the relative positions in a frame attached to the respective platoon vehicle. The vehicle frame is defined with the x-axis pointing forward and the y-axis out the passenger side as shown in Figure 7. The x component of the resulting vector was used to determine if the merging vehicle was in front of or behind a platoon vehicle. Note that in this coordinate frame a positive x represents the merge vehicle in front of a platoon vehicle, while a negative x represents a merging vehicle behind a platoon vehicle.

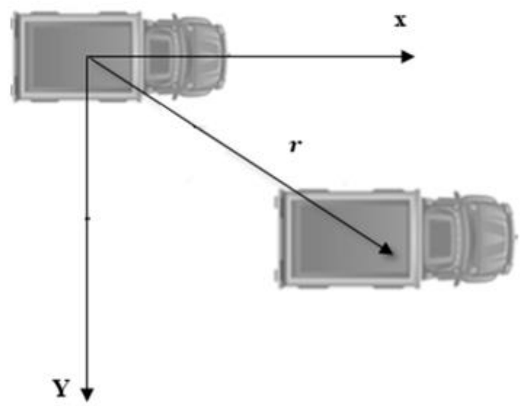

Figure 7. Coordinate frame to find relative distance between vehicles.

The range between platoon vehicles is estimated to use as feedback to the CACC longitudinal controller. The range estimation process fuses measurements from a forward-facing RADAR with Dynamic-Base Real-Time Kinematic (DRTK) GPS to produce a reliable estimate of inter-vehicle range and range rate [8]. The range variable, r , is defined as the distance from the RADAR to the rear-most point on the lead vehicle. Neighboring vehicles in other lanes are also tracked using the RADAR. Using this, in combination with a forward predicted path as described in [9] , allows for a platooning vehicle to determine when a vehicle has cut-in to the platoon. This feature allows the CACC system to respond if the merge decision is in-between the platoon. In the merge middle case, there are two scenarios that can happen: the following vehicle will fall back until it has reached its new safe following distance (200 ft) or fall back until the following vehicle detects a cut in. This functionality makes sure the following vehicle does not continue to fall back unnecessarily if the merge vehicle has already entered onto the road. Upon detecting that the merge vehicle has left the platoon, the desired gap then returns to its nominal value.

5. EXPERIMENTAL RESULTS

Using the previously described setup, a number of tests were performed to study the efficacy of the proposed merging algorithm. For each run, the merging vehicle began accelerating at different starting positions relative to the platoon in order to invoke different merging scenarios. With the CACC platoon of two vehicles, there are three possible outcomes for the merging vehicle: merge in the front, middle, or back of the platoon. For the front and back cases, the CACC platoon will continue normal operation and take no action. Furthermore, the platoon will create a gap for the merge vehicle if the FIFO algorithm calculates the middle scenario. Results from each of these cases are presented below. The relative position shows the distance between the merging vehicle and each vehicle in the platoon. This value is used to show the position of the merging vehicle relative to the platoon, i.e. in front, middle or behind. The vertical dashed line on each plot indicates when the FIFO algorithm calculated the merge decision. The time estimates shown are calculated for their respective vehicles using Equations (3) and (8).

5.1 Merge Middle

The following results present a case where the merging vehicle joins the middle of the platoon. When this scenario occurs the follower platoon vehicle accounts for the merging vehicle and increases its following gap to allow space for the merging vehicle. Figure 8 shows the relative distance between the merging vehicle and each vehicle in the platoon, as well as the time estimates for each vehicle. In this run, the merging vehicle remains in front of the follower platoon vehicle, as noted by the positive relative position, and behind the lead platoon vehicle, as noted by the negative relative position. At the decision point, the merge vehicle is ~5 m ahead of the follower platoon vehicle, and ~25m behind the leader platoon vehicle, thus providing a scenario where the merging vehicle should merge in-between the platoon. The time estimates show that the merging vehicle is estimated to beat the follower platoon vehicle to the merge point throughout the run. At the decision point, the merge vehicle was estimated to reach the merge point ~0.15 seconds faster than the follower platoon vehicle. While extremely close, this was above the set $t_{cushion}$ of 0.125 seconds used for testing, thus triggering the platoon to adjust its set following gap to allow room for the merging vehicle.

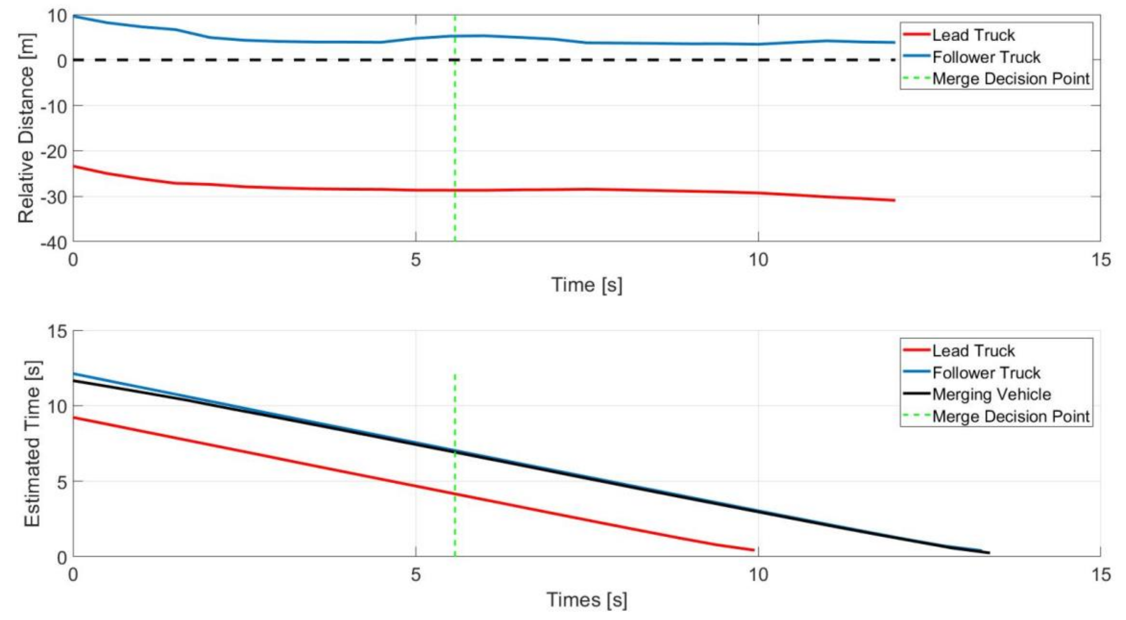

Figure 8. Relative distances and time estimates of merge and platoon vehicles during Merge Middle.

Once the decision is made at around 30 seconds, the following vehicle responds as shown in Figure 9. The new desired range was changed at a steady ramp input in order to prevent any use of the foundation brakes of the truck, since a step input of the reference would cause a large range error. This new reference can be modified to any desired input, but the optimal reference will depend on many factors. For example, considerations include how fast the desired gap needs to open, what speed the platoon is traveling, or even an optimal maneuver for fuel usage. The following truck tracks this changing reference and increases the gap until it reaches the desired 60 m gap. The truck maintains this gap until the merging vehicle cuts in front of the follower vehicle, seen by the large spike in range around 48 seconds, at which point the follower vehicle resets its reference back to the original 30.5 m gap and ranges off of the merging vehicle until it leaves the platoon. The merging vehicle leaves the platoon around 55 seconds, at which the follower vehicle resumes normal operation by ranging off the platoon leader and speeding up to close the gap.

Similarly, Figure 10 shows relative distances and time estimates of each vehicle during another run. In this run the merge vehicle begins slightly in front of the lead platoon vehicle, but is eventually passed around 2 seconds. Conversely, the merge vehicle remains well ahead of the follower platoon vehicle during the entire run, thus presenting a scenario where the merge vehicle should again join the middle of the platoon. The time estimates agree with this, as it is estimated that the merge vehicle will beat the follower platoon vehicle. As a result, the decision at 9 seconds triggers the platoon to again adjust its following gap for the merge vehicle.

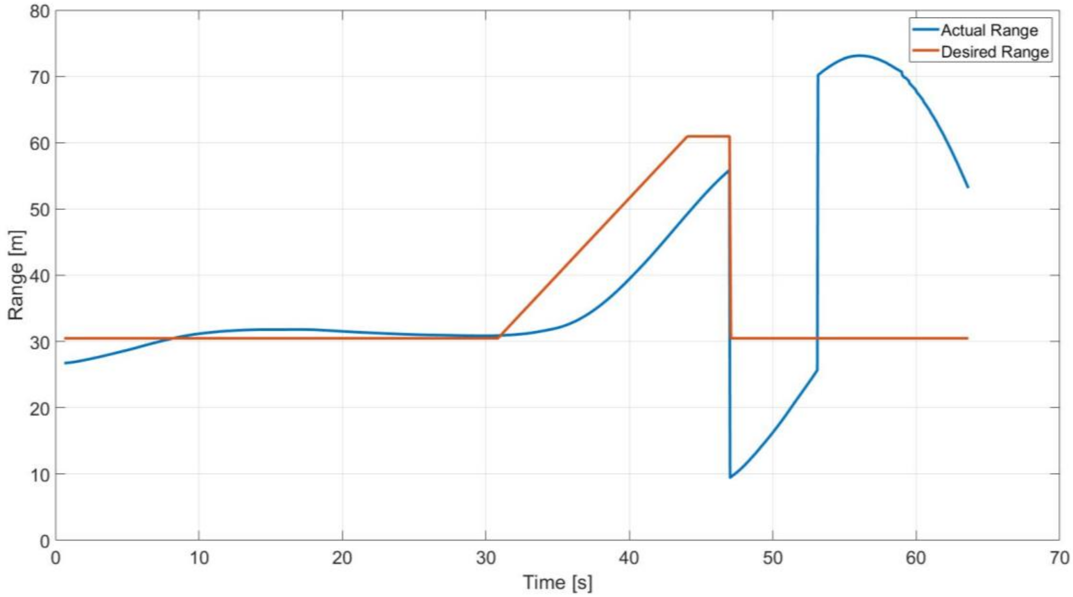

Figure 9. Follower vehicle range information during Merge Middle.

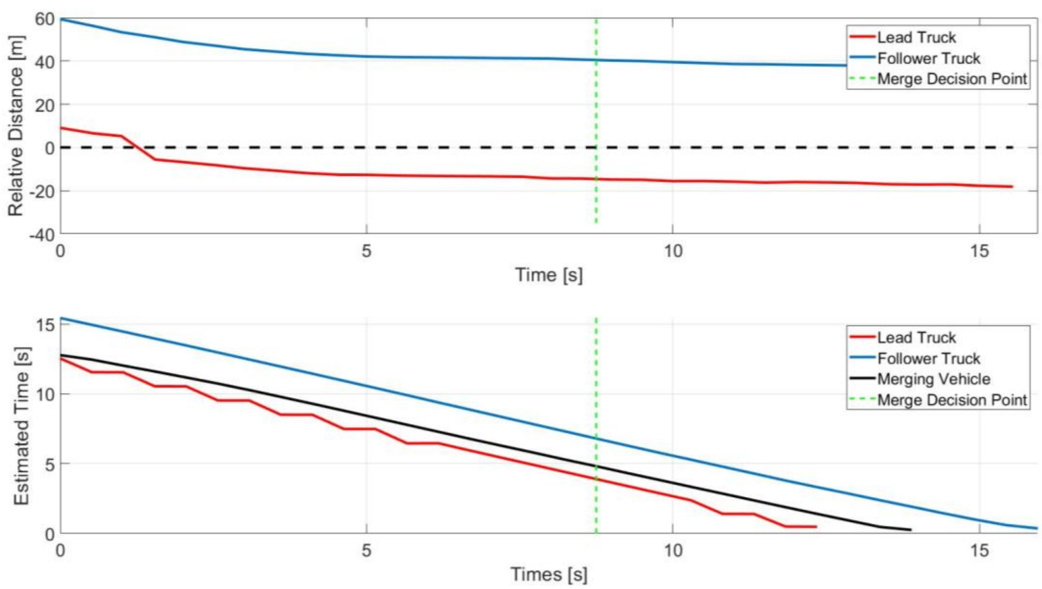

Figure 10. Follower vehicle range information during Merge Middle.

The follower platoon vehicle's response to this decision is presented around 32 seconds in Figure 11 below. Similar to the previous run, the follower vehicle drifts back due to the changing reference. However, the merge vehicle cuts in front of the truck before it reaches it new gap, thus causing the truck to begin ranging off the merge vehicle prior to opening the entire gap. This can be seen around 42 seconds by noting the sudden spike in range and desired range. The truck continues to range off the merge vehicle, and then at 50 seconds the merge vehicle leaves the middle of the platoon. This causes the truck to resume ranging off the lead platoon vehicle and close the large gap created.

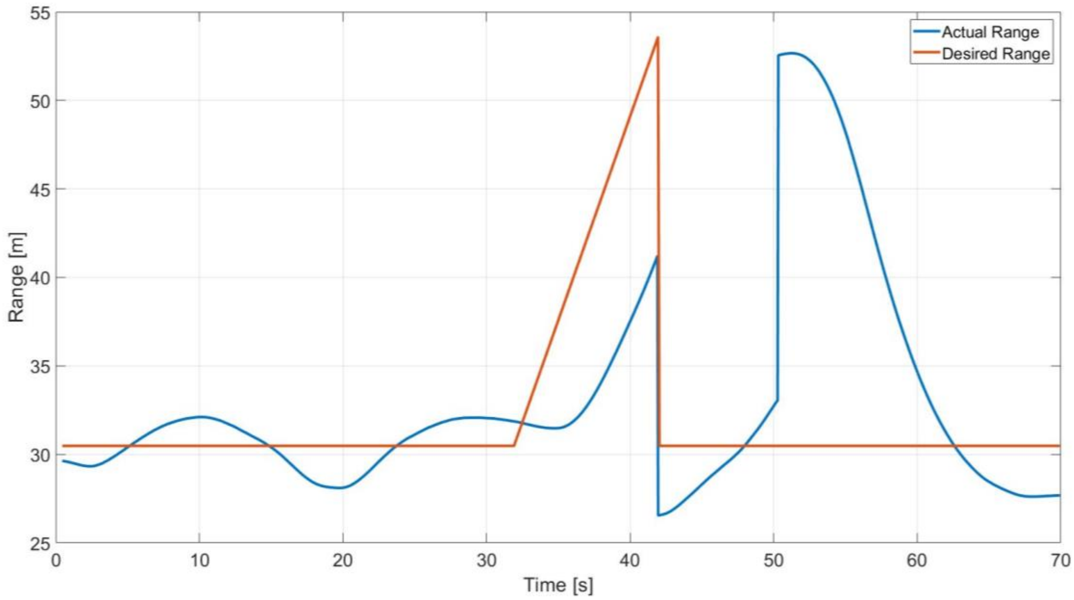

Figure 11. Follower vehicle range information during Merge Middle.

Video of a merge middle scenario was taken using a forward-facing camera mounted to the follower vehicle. The video is presented below, Video 1, as a visual aid in understanding the scenario presented.

Video 1. Video from follower truck during Merge Middle. http://dx.doi.org/doi.number.goes.here

5.2 Merge Front

The following results present a case where the merging vehicle joins the front of the platoon. In this scenario, no action is required by the platoon vehicles and the merging vehicle simply needs to drive to the front of the platoon. Figure 12 shows the relative distances and time estimates for each vehicle. As seen, the merging vehicle remains in front of both the lead

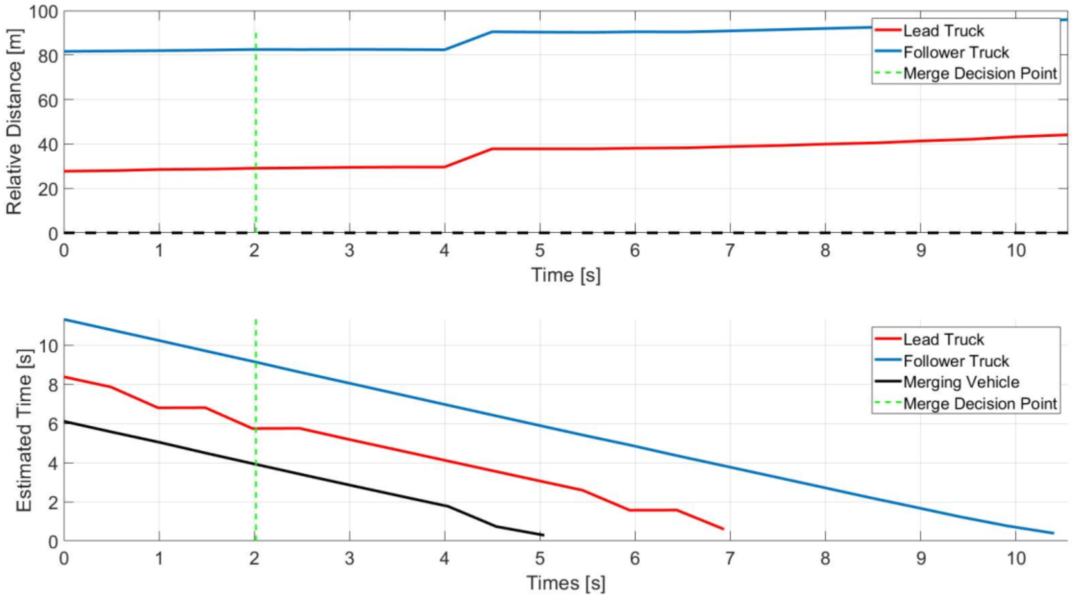

Figure 12. Relative distances and time estimates during Merge Front.

and follower vehicles during the run, giving a scenario where the merging vehicle should go in front of the platoon. The time estimates agree with the case presented, as it is estimated the merge vehicle will beat both platoon vehicles. Because of these estimates, the decision at 2 seconds is for the platoon to take no action and continue normal operation. Figure 13 shows corresponding following vehicle response during this situation. As shown, the platoon does not change from normal operation as the desired range stays at the initial 30.5 m (100ft) gap.

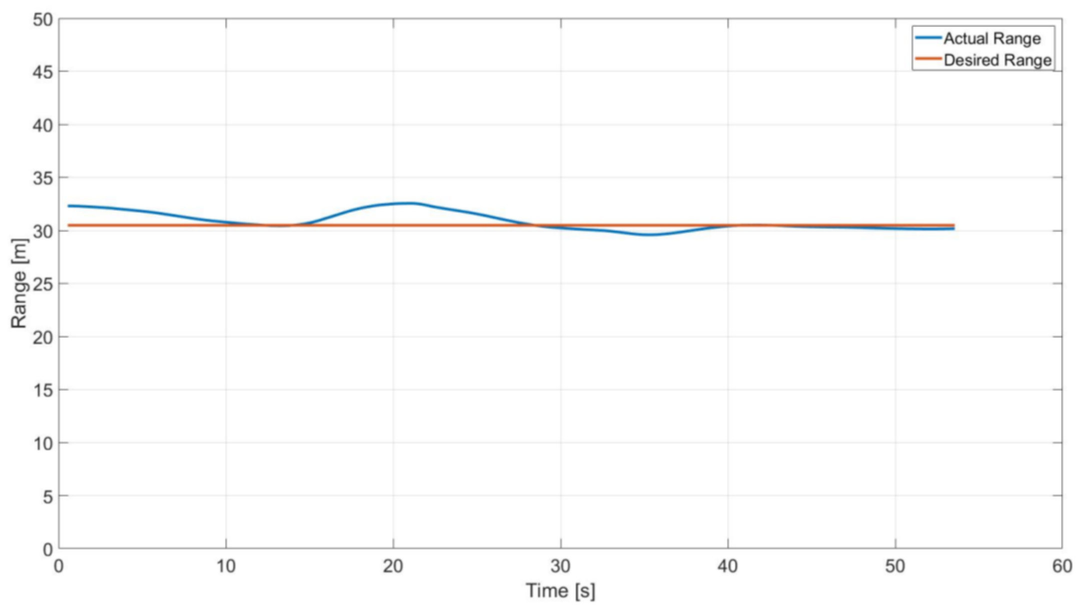

Figure 13. Follower vehicle range information during Merge Front.

5.3 Merge Behind

The following results present a case where the merging vehicle joins the back of the platoon. In this scenario, no action is required by the platoon vehicles and the merging vehicle simply needs to drive to the back of the platoon. Figure 14 shows the relative distances between platoon vehicles and the merging vehicle during this scenario. As seen, the merging vehicle remains behind both the lead and follower vehicles during the run, providing a scenario where the merging vehicle should go to the back of the platoon. The time estimates agree with this case, as it is estimated the entire platoon will beat the merge vehicle. Because of these estimates, the decision at 10 seconds is for the platoon to take no action and continue normal operation. Similarly, Figure 15 shows the follower vehicle response. Once again, the platoon keeps its set gap distance throughout the duration of the merge scenario.

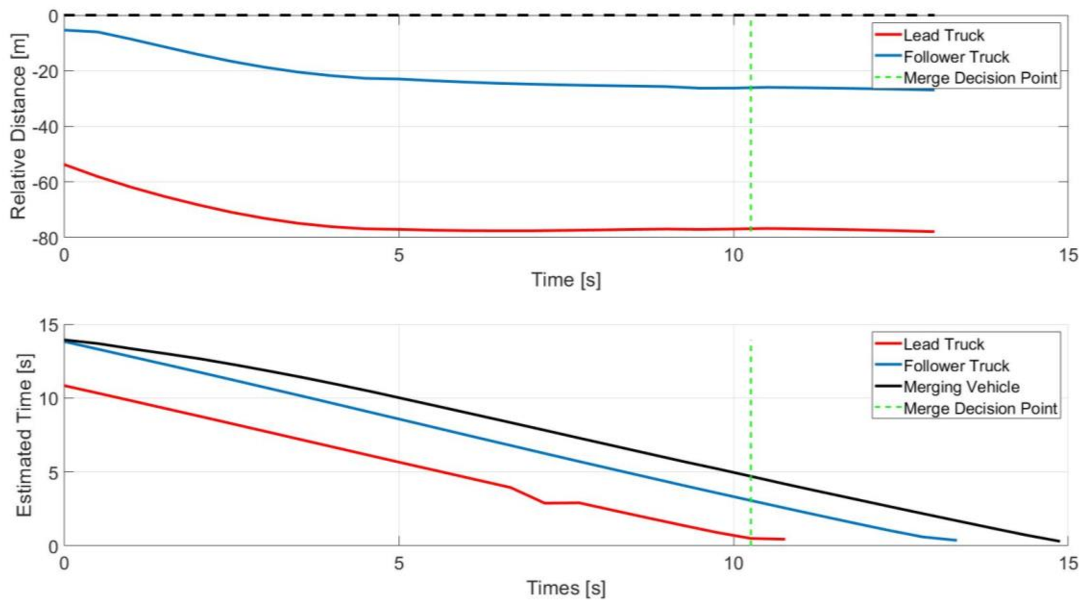

Figure 14. Relative distances and time estimates for Merge Behind.

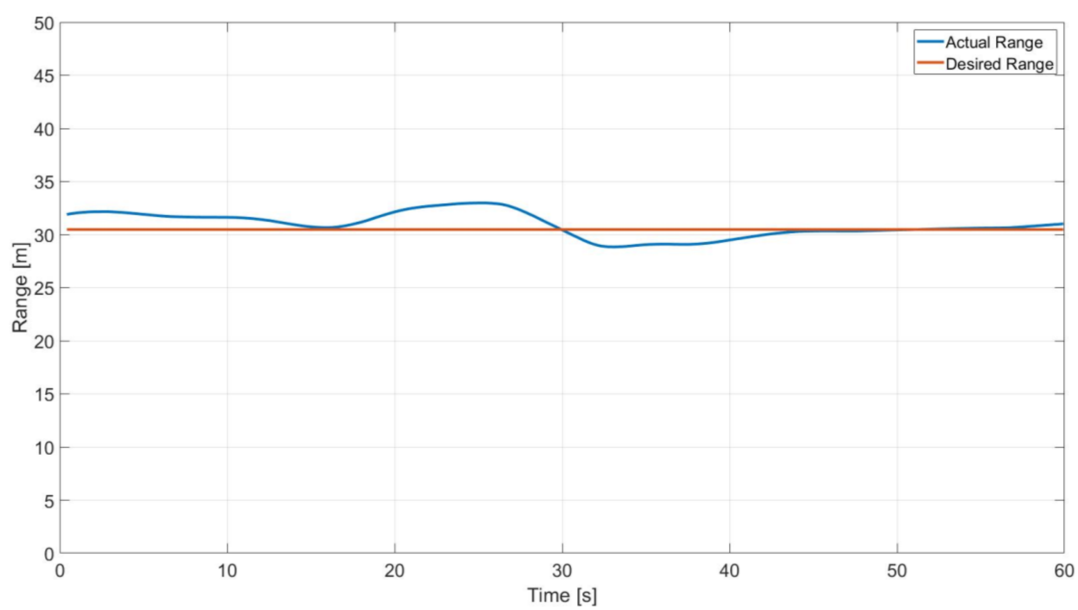

Figure 15. Follower vehicle range information during Merge Behind.

6. CONCLUSIONS/FUTURE WORK

Overall, a method for estimating the trajectory of a connected, non-automated vehicle was developed, and the results were used to successful demonstrated merging scenarios. This merging algorithm was implemented onto Auburn University's CACC platoon and tested in merging scenarios at NCAT. The algorithm's decision and the system response show that a gap was successfully created as needed for the merging vehicle. The results were shown on a test track, but future plans include implementation on real highway merging ramps. Besides the proper CACC system and vehicles, only a pre-defined merge point and the speed limit of the highway must be known to implement on a real merge ramp. Furthermore, infrastructure to vehicle (I2V) communication, such as a Road Side Unit (RSU), could also be used in place of a connected vehicle. This approach will also be studied in future work. In conclusion, the presented approach is a simple, effective method for merging of connected vehicles into an automated platoon.

Acknowledgements

The authors would like to acknowledge the U.S. Army Combat Capabilities Development Command (CCDC) Ground Vehicles Systems Center (GVSC) for funding of this project. The testing and results were made possible by the NCAT test track and the professional truck drivers.

References

[1] A.T. McCartt et al., "Types and characteristics of ramp-related motor vehicle crashes on urban interstate roadways in Northern Virginia", Journal of Safety Research, vol.35, 2004, pp. 107- 114.

[2] Bruce N. Janson, Wael Awad, Juan Robles, Jake Kononov, Brian Pinkerton, "Truck Accidents at Freeway Ramps: Data Analysis and High-Risk Site Identification", Journal of Transportation and Statistics, January 1998, p 75 – 92.

[3] Kesting, A., Teriber, M., and Helbin, D., "General Lane-Changing Model MOBIL for Car-Following Models", JTR, 1999(1), 86-94(2007).

[4] Ahmed, Md Salman, et al. "A Cooperative Freeway Merge Assistance System using Connected Vehicles." *arXiv preprint arXiv:1805.00508* (2018).

[5] Lombard, A., Perronnet, F., Abbas-Turki, A., Moudni, A., "On the cooperative automatic lane change: Speed synchronization and automatic "courtesy"", IEEE Design, Automation & Test in Europe Conference & Exhibition, (2017).

[6] Kazerooni, E., Ploeg, J., "Interaction Protocols for Cooperative Merge and Lane Reduction Scenarios", IEEE 18th International Conference on Intelligent Transportation Systems, (2015).

[7] Rios-Torres, Jackeline, and Andreas A. Malikopoulos. "Automated and cooperative vehicle merging at highway on-ramps." IEEE Transactions on Intelligent Transportation Systems 18.4 (2017): 780-789.

[8] T. Sherer, "Radar probabilistic data association filter with GPS aiding for target selection and relative position determination," Master's thesis, Auburn University, Auburn, Alabama, 2017.

[9] D. Baum, C.D. Hamann & E. Schubert (1997) 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.