2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Real-time Priority-based Cooperative Highway Merging for Heterogeneous Autonomous Traffic

Nilesh Suriyarachchi[1], Faizan M. Tariq[1], Christos Mavridis[1] and John S. Baras[1]

Abstract— **Highway on-ramp merge junctions remain a major bottleneck in transportation networks. However, with the introduction of Connected Autonomous Vehicles (CAVs) with advanced sensing and communication capabilities modern algorithms can capitalize on the cooperation between vehicles. This paper enhances highway merging efficiency by optimally coordinating CAVs in order to maximize the flow of vehicles while satisfying all safety constraints. Focus is also placed on the effect of varying priorities of different vehicle classes in selecting the best merging sequence. Our algorithm is capable of real time operation through parallel computation, optimized merge sequence generation and management of the diverse needs of heterogeneous (multi-class) traffic. Results are verified through a realistic traffic simulation software.**

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564994

I. Introduction

Highway on-ramps still remain one of the biggest causes of traffic in modern highways. Due to the lack of coordination and limited visibility among vehicles, it is very common to see a traffic buildup around highway on-ramps [1]. The delays caused at on-ramps are some of the major contributors to overall system efficiency degradation [2]. In periods of high demand, the effect of slow moving vehicles attempting to merge into a stream of fast moving vehicles creates a bottleneck that often leads to traffic buildup in a large radius around the merge junction. Solving this problem requires vehicles to cooperate with each other and create gaps to enable smooth merging at high speeds [3]. This unfortunately is extremely difficult to achieve with human driven vehicles. However, with the advent of CAVs a lot more information has been made available for improving this overall process.

Modern CAVs have improved 360° local sensing with on-board sensors such as Lidar, radar and camera systems. Furthermore, advancements in networking technologies like 5G have led to significant improvements in Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication capabilities, even beyond the point discussed in [4]. Overall latency or delays in networks have been reduced and the bandwidth available for data communication has increased. Due to this, recent research has focused on solving the merging bottleneck by harnessing the cooperation capability of CAVs.

The general highway merging problem can be divided into two stages. The selection of the optimal merging sequence (order in which to merge) and controlling the actuation of all the vehicles in order to achieve the selected sequence. Sequence selection can become very difficult, as the number of possibilities grows exponentially with the number of vehicles involved. Another consideration in the merging problem is the challenges introduced by the presence of heterogeneous traffic. Different types of vehicles may prioritize different aspects such as speed maximization or minimization of acceleration and deceleration tasks. For example an emergency vehicle will prioritize speed. Therefore, a practical controller for solving the merging problem should be capable of finding an optimized solution in real time while considering the varying needs brought about by mixed traffic.

Literature review

The problem of improving the throughput in highway merge junctions has been evolving with the advent of new technologies. Prior to the introduction of CAVs, most work was focused on ramp metering strategies [5], which decide how many vehicles can merge [6], but don't consider the micro-simulation aspects of the merging process. However, as vehicles became more advanced with improved sensing, decision making and communication capabilities, we now see a lot of effort placed on improving the merge process by taking advantage of the cooperativeness between connected vehicles. These efforts are often categorised either as rule based methods or optimization based methods.

Rule based methods attempt to obtain near-optimal solutions with minimal computation cost. Some initial work on a zip based method where vehicles take turns in merging is presented by Sarvi and Kuwahara [7] and Scarinci *et al.* [8]. The work by Awal *et al.* [9] and more recently Ding *et al.* [10] showcase the improvements cooperation between vehicles can provide. While these algorithms are capable of real time operation the result is usually non-optimal. There is no guarantee that the optimal sequence is found among the merging sequences generated, and it is difficult to model the interaction between vehicles.

The optimization based methods aim to overcome this and obtain the optimal merge sequence and optimal control for all vehicles near the merge junction. However, this process generally has a very high computation cost. Most existing methods aim to circumvent this problem using different types of assumptions. The work by Rios-Torres *et al.* [11], [12] demonstrate how the unconstrained optimization problem can be solved in real time with Hamiltonian analysis. However, when the full constrained problem incorporating collision constraints is considered the method presented cannot be used. Optimization is restricted to one ramp vehicle

[1]Electrical and Computer Engineering Department and the Institute for Systems Research, University of Maryland, College Park, Maryland, USA. Email: {nileshs,mftariq,mavridis,baras}@umd.edu

Research partially supported by ONR grant N00014-17-1-2622

and one mainline facilitating vehicle in the work by Zhou *et al.* [13]. Research by Chen *et al.* [14] attempts to solve the constrained problem using a two level Mixed Integer Quadratic Programming (MIQP) optimization formulation. However, this is very computationally costly and cannot be solved in real time. In fact they only compute sequences with one ramp vehicle for a very short time duration. This same problem is faced by other optimal control based approaches such as Li *et al.* [15].

Another interesting approach to formulating this merging problem is the virtual slot based approach [16]. Here a central controller creates virtual slots on the highway which each vehicle is then allocated to. This allows vehicles to be controlled centrally by controlling the slot parameters. This type of formulation performs poorly in the presence of mixed traffic and also does not provide an optimal solution to maximizing merging throughput. A similar virtual vehicle concept is also introduced by Uno *et al.* [17]. While all cooperative merging algorithms require some form of vehicle to vehicle or infrastructure communication, the work by Ito *et al.* [18] shows how the communication overhead can be reduced by using different broadcast strategies.

Contribution

Because of the nature of the highway merging problem, rule-based methods tend to yield sub-optimal solutions while methods based on end-to-end optimization are, by definition, computationally expensive and can hardly be used in real time. Moreover, the existing methods make the assumption of homogeneous traffic, while the potential additional complications of mixed/heterogeneous traffic conditions have not been adequately studied.

The main contribution of this work is to ameliorate both these issues by creating a real time operation capable hybrid rule and optimization based merging algorithm with a parallel computation design. This hybrid approach leads to better performance without sacrificing real time computation capabilities. This also allows the solution provided by the algorithm to be improved depending on the available computation resources. The varying requirements of different classes in heterogeneous traffic are handled by incorporating the concept of individual vehicle priorities into the optimization formulation.

Finally, the algorithm is tested on a realistic traffic simulator under varying traffic conditions, and the results and comparisons are presented. We show that our method leads to improved traffic performance metrics, such as higher throughput, lower density, reduced delay and increased fuel efficiency.

II. Problem Definition

In this section we formally define the fully automated highway merging problem as a large optimal control problem, and show that, by decomposing it in to two simpler decoupled optimization problems, which employs a parallel computation architecture, a near-optimal solution can be computed in real time.

A. Modelling the physical system

Consider the abstracted model of a highway on-ramp as shown in Fig. 1. We define a control zone, where all vehicles in this zone communicate with a central controller (V2I communication) and each other (V2V communication) to decide individual optimum velocities and paths to be followed. The control zone encompasses both the main road and on-ramp. We denote the number of vehicles in the control zone by n, with m vehicles located in the mainline, and $r = n - m$ located on the on-ramp. Vehicles are allowed to merge from the on-ramp onto the main road in the merge zone, which is located at the end of the control zone.

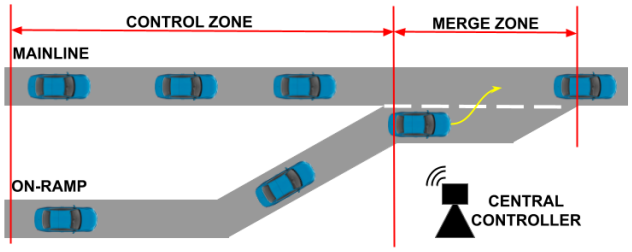


Fig. 1: On-ramp Merging Regions and Infrastructure Model

In practice, the actual dynamics of the vehicles are unknown and highly non-linear, and, therefore, cannot be accurately modeled by the solver. In this regard, we assume that the low-level control of a vehicle is managed by a local controller w_i, which is also responsible for the control of lateral motion that keeps the vehicle in lane. Therefore, the vehicle can be modelled as a point mass moving along the center of the lane according to a non-linear differential equation:

$$\dot{s_i} = f(t, s_i, w_i), \qquad s_i(t_i^0) = s_i^0 \tag{1}$$

where t^0 is the time the vehicle enters the control zone. Throughout this manuscript, $s_i(t)$ will be measured as the distance of the vehicle i from the merge zone. Therefore, we can define the high-level vehicle dynamics by the following velocity control scheme:

$$\begin{aligned} \dot{s_i} &= v_i \\ v_i(t) &= u_i(t) \end{aligned} \tag{2}$$

where $s_i(t)$, and $v_i(t)$ denote the position and velocity of each vehicle i, respectively, along the direction of the lane, for $i \in \{1, \ldots, n\}$.

Remark 1: The control input $u_i(t)$ is the command velocity for vehicle i. It is necessary that the target set by this control is reachable by the low-level vehicle controller w_i and system (1). It is understood that, since u_i will be the solution of the optimal merging problem, it should be designed to guarantee safe operation under small delays induced by the dynamics in (1).

In addition to $s_i(t)$ and $v_i(t)$, we assign a priority value $p^i \in \mathbb{R}_+^2$ to each vehicle i, which affects the merging decision, and an indicator variable $b^i \in \{0, 1\}$ which signifies whether the vehicle is on the on-ramp or the mainline. The

priority value has two scalar components, speed prioritization (p_s^i) based on the focus placed on vehicle velocity and speed variation prioritization (p_v^i) based on the tolerance to changes in velocity. For each vehicle we also obtain its length l^i, and bounded acceleration capabilities characterized by its maximum acceleration a_{max} and maximum braking a_{min} capabilities, which carry information about lower-level variables such as the mass of the vehicle. Therefore, all cars are connected autonomous vehicles (CAVs) completely defined by the state vectors:

$$x_i(t) = [s_i(t), v_i(t), p^i, b^i, l^i, a^i_{max}, a^i_{min}]^{\mathrm{T}} \tag{3}$$

for $i \in \{1, \dots, n\}$. Lastly, regarding the parameters of the main road and the on-ramp, we assume speed limits $\bar{v}_m$ and $\bar{v}_r$, respectively, and we denote their length of the control zones by L_m and L_r. Without loss of generality, we will assume in this work that $\bar{v}_m = \bar{v}_r = \bar{v}$, a constant value that can be decided by the designer. The constant parameter tuple $M_s = \{M_{sr}, M_{sl}\}$, which denotes a safety margin between two vehicles, is also defined, and can be chosen by the designer. Here, M_{sr} and M_{sl} represents the safety margins needed to prevent rear-end collisions and lateral collisions respectively.

B. Optimal merging control formulation

The highway merging problem, can be formulated as an optimal control problem, where the goal is to estimate the optimal command velocities u_i for all $i \in \{1, \dots, n\}$, according to an appropriately defined objective function. Essentially, we want to minimize the time it takes for all vehicles to merge, which can be written as a minimization of the merging time of the last vehicle to be merged:

$$\begin{aligned} &\min_{\{u_i\}} \quad \max_i \ t_f^i \\ &s.t. \qquad C(\{x_i\}, \bar{v}, L_m, L_r, M_s) \end{aligned} \tag{4}$$

where $t_f^i \in \arg\min_\tau \{s_i(\tau) = 0\}$ is the merging time of vehicle i. The set C is a set of constraints, to be defined later, that depend on the state vectors $\{x_i\}$ of the vehicles, equation (2), and the parameters introduced in Section II-A. Based on the CAV assumption, near an optimal configuration, we expect that all the velocities $v_i(t)$ will be close to the limit $\bar{v}$. Thus, we can expect a near-optimal solution to the minimum-time optimization problem (4) by solving:

$$\begin{aligned} &\min_{\{u_i, q\}} \quad \sum_{i=1}^{n} p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1-\lambda)(u_i - v_i)^2 \\ &s.t. \qquad C(\{x_i\}, \bar{v}, L_m, L_r, M_s, q) \end{aligned} \tag{5}$$

The first term of the new objective function tries to keep the velocity command as close to the speed limit as possible, since higher velocities result in shorter merging times. The second term refers to Remark 1, and acts as a regularization term, penalizing a fast change in the velocity command u_i with respect to the current velocity of the car. Lastly, p_s^i and p_v^i correspond to the priority values explained below and define a heterogeneous traffic model. We note that this new objective function consists of a weighted sum that yields a Pareto optimal point with respect to the parameter λ which can be chosen by the designer.

However, the optimization problem (5) is still practically intractable, since the set C contains constraints with respect to the order of the vehicles in their respective queues in the two lanes. Observe that the solution to the optimal control problem (5), also yields the optimal sequence q^* with respect to which the vehicles should merge. This makes it a mixed-integer optimization problem which is typically solved by solving (5) over all possible merging sequences q after exhaustive search. This essentially decouples the two problems of generating a sequence, and finding the optimal velocity commands u_i. However, the total number of possible sequences q grows exponentially with the number of vehicles in the control zone. As iterating through such a large number is impossible in a real time applications, in the next section we propose a heuristic approach to search inside the sequence space. This approach can reduce the computational complexity by orders of magnitude, while the complexity-performance trade-off can be controlled by the designer with respect to the computational power of the system. We also transform the remaining optimization problem to a set of quadratic programming problems, one for each generated sequence, that can be solved in parallel and in real time.

III. Real-time Priority-based Highway Merging

The implementation of a fully automated highway merge system can be decomposed into two major tasks: (a) the selection of the optimal merging sequence in which vehicles should merge, and (b) the implementation of this merging sequence by controlling each individual vehicle.

The initial task of the centralized controller is the allocation of a unique ID to all vehicles entering the control zone. This ID takes values between 1 and N, where N is the maximum number of vehicles that can fit inside the control zone. The vehicles are then added to either the mainline queue or the on-ramp queue.

A. Priority assignment

Most algorithms found in the literature, treat all vehicles as equal in the merging process and use methods such as first in first out (FIFO) to decide the merging sequence. However, this approach is sub optimal and performs very poorly in the presence of heterogeneous traffic. This is further complicated, since, different vehicles should be assigned different priorities based on multiple factors such as:

- Vehicle type, size and mass
- Emergency vehicle
- On ramp or main line vehicle
- Current speed
- Vehicle's future intent

Some examples of this are as follows. A loaded heavy 16 wheeler truck on the highway should not be forced to slow down drastically to enable an on-ramp vehicle to merge. Vehicles should give way to emergency vehicles. A vehicle intending on taking the next off-ramp can be slowed to enable other vehicles to merge in front.

To handle these requirements, prior to the optimization stage which selects the optimal merging sequence, a priority assignment node updates all the priority values of the vehicles in the control zone. The algorithm considers two types of priority values for every vehicle:

1) Speed Prioritization (p_s^i)
2) Speed Variation Prioritization (p_v^i)

Speed priority is concerned with how much a vehicle desires to maximise it's own speed. In mixed traffic this is especially important for emergency vehicles. On the other hand Speed Variation priority is based on the vehicles deterrence to change it's current velocity. While emergency vehicles would not mind sacrificing speed variation to gain higher speed the opposite is true for trucks since a high speed variation would involve incurring a high fuel cost. The steps used in updating these priorities are as follows. Each vehicle class has a baseline speed priority and variation priority value. This value is then adjusted depending on the vehicles current speed, whether its on the mainline or not and whether it has been waiting in the queue for a long time. This modification in priority depending on vehicle delay acts as a fairness factor to ensure that even low priority vehicles will get a chance to merge eventually. Finally, an adjustment for the flow over effect caused by high priority vehicles is made. This effect involves increasing the priority of all vehicles ahead of a high priority vehicle in a single line queue. This ensures that the high priority vehicle will not be slowed down too much by low priority vehicles ahead of it in the queue.

B. Optimal sequence generation

When deciding the best sequence in which to merge highway traffic the total number of possible sequences grows exponentially with the number of vehicles in the control zone. For example in the practical case with $n = 30$ vehicles ($m = 15$ mainline and $r = 15$ ramp) in the control zone the number of possible sequences is $\frac{(m+r)!}{m!r!} = 155117520$. As iterating through such a large number is impossible in a real time application, this approach starts at a initial seed sequence based on time to merge and computes perturbations to this sequence depending on available compute power. Once we have a good selection of sequences to check, then our problem reduces to choosing which of these possible sequences provides the best performance. Therefore, our algorithm will analyze all possible vehicle inputs in order to make the optimal selection. In order to do this we need to calculate the optimal vehicle control input (acceleration/deceleration) needed by each vehicle to achieve each desired merge sequence. This calculation of inputs for all of the vehicles in the control zone for a specific merge sequence is formulated as a quadratic programming optimization problem.

In the process of generating the possible sequences which the optimizer needs to compare, we find that the number of possible highway merge sequences is limited due to two factors. The length of the control zone which decides the maximum number of vehicles in the sequence and the requirement for order in individual lanes which means mainline vehicles can't overtake other mainline vehicles and the same is true for ramp vehicles. Using this fact, in this algorithm, a sequence is formed by injecting the on-ramp vehicles into the string of mainline vehicles. There are two extreme scenarios that are considered when injecting ramp vehicles into the main line vehicle queue. The first criterion is time to merge (TTM) (t_M) which takes into account the vehicles speed and position. In this criterion, slow moving vehicles may be allocated to the end of the sequence even though they are close to the merging point. The other criterion is distance to merge (DTM) (d_M). Here, vehicles moving fast will need to slow down in order to allow a slow moving vehicle closer to the merging point to merge. We find that the optimum sequence lies in between these two extremes. Therefore, this algorithm follows a scalarization approach, and performs a trade-off analysis between DTM and TTM controlled by the parameter α:

$$(1-\alpha) * t_M^i + \alpha * d_M^i, \quad \forall i \in \{1, \ldots, n\} \tag{6}$$

to compute the merge criteria from which the test sequences are generated. In order to generate a range of sequences, we vary α from 0 to 1 and the resolution is selected according to the desired quality of the sequences, available compute power and traffic levels in the control zone. When the number of vehicles in the control zone increases the resolution of α can be made fine at the cost of computation.

A key feature of this algorithm is that the optimization process for each sequence is designed to run in parallel. This means that we can increase the α resolution depending on the available compute cores without sacrificing real time performance. Every sequence thus generated is then fed into the optimizer in order to rank and select the best sequence and control variable values.

C. Optimal velocity computation

The optimal control problems, one for each specific sequence under consideration q, seek to find the optimal command velocities of each vehicle u_i, which would facilitate the merging of vehicles according to the sequence q in an optimal way.

1) Objective function: The objective function to obtain u_i for a given sequence q, is formulated as:

$$J(u_i|q) = \sum_{i=1}^{n} p_s^i \lambda (u_i - \bar{v})^2 + p_v^i (1-\lambda)(u_i - v_i)^2 \tag{7}$$

In addition to the interpretation given in Section II-B, the objective function $J(u_i|q)$ can be viewed as a trade-off between minimizing the control effort (fuel consumption) and maximizing the throughput of vehicles through the merge zone. Throughput maximization is achieved through the minimization of the difference between the vehicles command velocity (u_i) and the speed limit ($\bar{v}$). This stems from the fact that the throughput through the merging bottleneck is maximized when the speeds of all the participating vehicles are maximized. This formulation focused on mean velocity maximization is also based on the inherent nature of CAVs

to maximize velocity in order to reach target destinations in the shortest possible time.

2) Constraints: The constraints (set C in (5)) imposed on this optimization problem ensure safety of operation during merging (Remark 1), by preventing collisions between vehicles, while providing controls which lie within the capabilities of each vehicle.

First, the command velocity $u_i(t)$, for all $i \in \{1, \ldots, n\}$ and $0 \leq t \leq t_f^i$, should be bounded based on the speed limits:

$$0 \leq u_i(t) \leq \bar{v} \tag{8}$$

as well as based on the acceleration capabilities of each vehicle:

$$a_{min}^i \Delta t \leq u_i(t) - v_i(t) \leq a_{max}^i \Delta t \tag{9}$$

where Δt is set as the time resolution in our computations.

The next constraint refers to the dynamics of the system and considers the new expected position of vehicle i after time Δt:

$$s_i(t+1) = s_i(t) - \Delta t \frac{v_i(t) + u_i(t)}{2} \tag{10}$$

The assumption that u_i is chosen such that the vehicle can reach it in time Δt is justified by (9) and the second term of the objective function (7).

The expected position parameter computed by (10) is used to ensure rear-end collisions do not occur on the mainline and on-ramp respectively:

$$|s_j(t+1) - s_{j'}(t+1)| \geq l^j + M_{sr} \tag{11}$$

$$|s_k(t+1) - s_{k'}(t+1)| \geq l^k + M_{sr} \tag{12}$$

for all $j, j' \in \{1, \ldots, m\}$ mainline vehicles with $j \neq j'$ and for all $k, k' \in \{1, \ldots, r\}$ ramp vehicles with $k \neq k'$.

Here, M_{sr} represents the safety margin used to prevent rear-end collisions.

The equations (11) and (12) are then simplified by substituting the new expected position $s_i(t+1)$ values from equation (10). The norm is removed using the ordered precedence assumption which prevents vehicles overtaking each other. We then obtain the form given in equations (13) and (14) for the prevention of all rear-end collision on both the mainline and on-ramp:

$$u_j - u_{j+1} \geq (v_{j+1} - v_j) + \tfrac{2}{\Delta t}(s_j - s_{j+1} + l^j + M_{sr}) \tag{13}$$

$$u_k - u_{k+1} \geq (v_{k+1} - v_k) + \tfrac{2}{\Delta t}(s_k - s_{k+1} + l^k + M_{sr}) \tag{14}$$

for all $j, j+1$ vehicles belonging in the mainline and $k, k+1$ vehicles located on the ramp.

We then compute time to merge t_m^i for each vehicle i by:

$$t_m^i = \frac{s_i(t+1)}{u_i} \tag{15}$$

Then the constraint,

$$t_m^i \leq t_m^{i+1} - \tfrac{l^i}{u_i} - \tfrac{l^{i+1}}{u_{i+1}} - M_{sl} \tag{16}$$

for all consecutive vehicles $i, i+1$ in sequence q, is responsible for ensuring that individual commands u_i are chosen such that vehicles merge in the order of the sequence under consideration q. Here, M_{sl} represents the safety margin used to prevent lateral collisions during merging.

By substituting from equations (10) and (15), we simplify equation (16) to (17). We also eliminate the l^{i+1} term since we restrict distance measurements to be always taken from a vehicle's front bumper. Therefore the following equation:

$$u_{i+1}(s_i - \tfrac{\Delta t}{2}.v_i + l^i + M_{sl}) \leq u_i(s_{i+1} - \tfrac{\Delta t}{2}.v_{i+1} - M_{sl}) \tag{17}$$

for all consecutive vehicles $i, i+1$ in sequence q, plays the role of both preventing lateral collisions during merging as well as enforcing the order of merging to follow the tested sequence q.

The constraints given by equations (8),(9),(13),(14) and (17) along with the objective function in (7), form the optimization problem for optimal vehicle command velocity computation, given a test sequence q.

As a last comment, we note that the presence of the variable b^i in the set of constraints C in (5) causes the initial formulation to be a mixed-integer quadratic programming (MIQP) optimization problem, which would be difficult to solve in real-time for this large scale problem. By incorporating the effect of b^i into the computed priority p^i and by setting up separate constraints for the mainline and ramp, as described above, we have reformulated the problem as a quadratic programming (QP) optimization problem. This is a very important result that has enabled fast computation of the solution of each optimization problem in real-time.

3) Optimal Sequence Selection: If a feasible solution is provided by the optimization above, the final cost value for the tested sequence q and the output u_i for each vehicle are stored. Once the optimization is carried out for each possible sequence q, the feasible sequences are sorted with respect to the minimum objective value achieved. Then the best sequence q^* at current time t is compared to the sequence assigned in the previous computation iteration $q^*_{(t-1)}$. Generally, the computed best sequence q^* along with the suggested u_i command velocities are broadcast to all the vehicles. However, if the previous sequence $q^*_{(t-1)}$ was also recalculated in the current cycle, and if the cost reduction in selecting the new sequence q^* when compared to the previous sequence $q^*_{(t-1)}$ is less than a predetermined switching threshold, then the previous sequence $q^*_{(t-1)}$ is transmitted to all the vehicles along with the new suggested u_i command velocities corresponding to this sequence. This ensures that the algorithm does not oscillate between two closely related sequences with comparable costs.

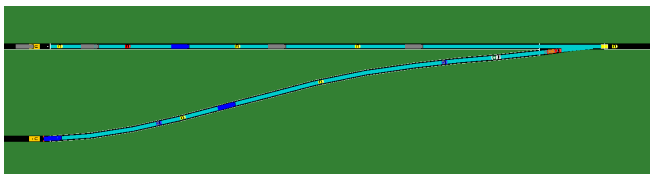
It is also important to note that this method allows the merging sequence to be changed at every update cycle, which is an essential feature for heterogeneous traffic. For example, when an emergency vehicle (EV) enters the control zone, the merging sequence should be modified to give precedence to the EV. This ability to switch rapidly and adapt to a new merge sequence is a core component of our approach.

4) Low-level vehicle controller: The optimal sequence and suggested u_i command velocities are received by each of the vehicles in the control zone. In this formulation we do not consider the delay of transmission and assume the data is available almost immediately via technologies such as 5G connectivity. The low-level vehicle controller in each vehicle will calculate the optimal control parameters (acceleration and deceleration) needed in order to achieve the desired u_i command velocity within the control time Δt duration.

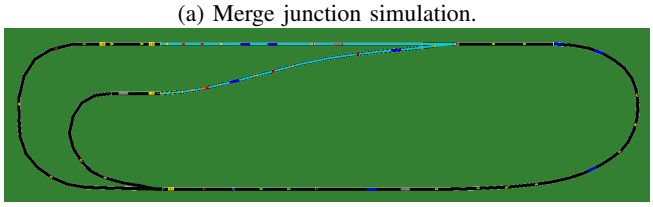
This low level controller should be made robust enough to handle perturbations in both sensing and actuation. In practical scenarios one cannot assume perfect information and actuation. There will always be certain failures in sensing such as receiving incorrect information broadcast from other vehicles or malfunctions in on-board sensing systems. Therefore, the low-level vehicle controller will attempt to reach the target velocity as long as it is safe to do so. The multitude of sensors on-board CAVs can be leveraged to check whether it is safe for the vehicle to reach it's target velocity.

IV. Experimental setup and Results

We evaluate the performance of the proposed approach in a highway section simulation using the SUMO simulator [19]. The simulation setup for the merge junction is shown in Fig. 2a. The quadratic programming optimization problems (Section III-C) are solved numerically using the Gurobi software (version 9.1.1) [20]. The controller communicates with the simulator using the TraCI traffic controller interface. All simulations and optimization algorithms run in a personal computer with an Intel i7-8750H CPU and 32GB of RAM.



(a) Merge junction simulation.



(b) Highway loop simulation.

Fig. 2: Experimental highway merge setup.

To simulate a continuous stream of vehicles and ensure that on average the ratios of different vehicle classes remain constant, we implemented a highway loop network as shown in Fig. 2b. Probabilistic rerouting is used to decide what percentage of the overall traffic is sent through the ramp. As we consider only one mainline lane the rerouting parameter is set to $r_p = 0.5$ for all tests. Multiple tests are conducted for different loop densities l_d (number of vehicles in the cyclic highway simulation). The loop density takes values in $l_d \in [20, 80]$, while ensuring that all vehicle classes were suitably represented. All tests were performed for a 2 hour (7200s) time period, which reduces the variance of the values of the performance scores of each algorithm tested. All the parameters used in the tests conducted are given in Table I.

TABLE I: System parameters used in testing

Parameter	Value
Control zone length	300 m
Simulation duration	2 hrs
Time step (Δt)	100 ms
Maximum number of sequences	12
Speed limit ($\bar{v}$)	27 m/s
Mainline-Ramp rerouting ratio (r_p)	0.5
Objective trade-off (λ)	0.7
M_{sr}	2.0
M_{sl}	10.0

A. Performance metrics

The key performance metrics used to evaluate merging algorithms are (a) *throughput* m_T (number of vehicles that can merge onto the highway per hour), (b) *density* m_D (number of vehicles on the link), and (c) *delay* m_L (Average delay experienced by vehicles compared to the ideal travel time). As density and throughput are related, we also consider their combined effect using (d) *mean velocity* m_V, which is obtained by $m_V = {}^{m_T}/_{m_D}$. Since we focus on mixed traffic special attention is paid to the delay faced by emergency vehicles. The total energy expended, (e) *fuel consumption* m_F, is also taken into consideration which shows the importance in the minimization of large fluctuations of the control commands. In this case we also specifically compare fuel consumption in heavy vehicles, such as trucks, as they consume larger quantities of fuel during acceleration tasks.

B. Algorithms

In order to showcase the properties and appropriately assess the performance of our methodology, we compare it with three different approaches. First, we make a comparison with the baseline case of on-ramp merging without cooperation among vehicles. In this baseline case, ramp vehicles always give way to mainline vehicles. Furthermore, we also compare against two commonly used collaborative merging strategies, which better showcase the strengths and potential weaknesses of our approach. The first of the two is a FIFO-based approach, in which each vehicle is assigned merging velocities in a greedy manner based on the order it enters into the control zone. The second method, utilizes a more advanced heuristic strategy known as zipper merging, in which the merge sequence is roughly chosen based on distance to merge with alternating selection between ramp and mainline. It is important to note that all these methods do not consider the heterogeneous nature of the traffic and all vehicles are treated equally. To our knowledge, as this is the first work to introduce a real time end-to-end optimization-based merging algorithm for heterogeneous traffic, no direct comparison with other optimization-based algorithms found in the literature would be fair or informative. This is also

compounded by the fact that other constrained optimization based methods cannot compute a real time solution for the large scale simulation used in this comparison.

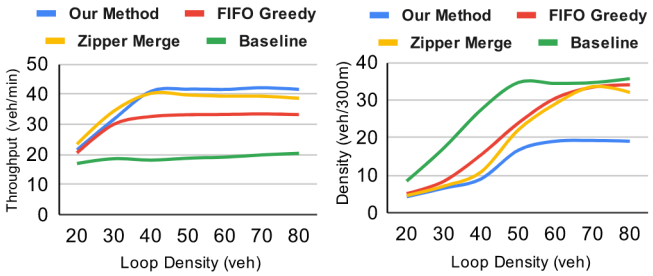
C. Results

The main objectives in the highway merge problem involve maximizing the mean velocity (m_V) and throughput (m_T), while minimizing the density (m_D) on each lane. It is also necessary to consider the minimization of the overall fuel consumption (m_F) during merging and the delay (m_L) experienced by individual vehicles.

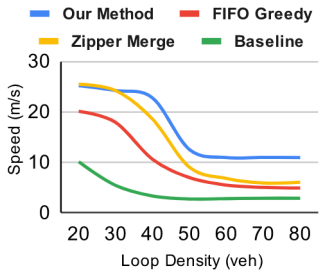
1) Main performance metric results: From the fundamental equation of traffic flow, we know that density m_D, the throughput m_T, and the mean velocity m_V, are linked according to equation (18).

$$m_T = m_V * m_D \tag{18}$$

Therefore, these three factors are considered in unison. In practice, the main goal of a highway merge control algorithm is to keep the traffic flowing as fast as possible with minimum density buildup on the lanes.



(a) Throughput variation (b) Density variation



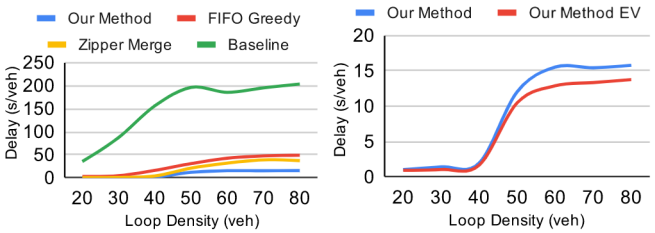
(c) Mean velocity variation

Fig. 3: Comparison of performance for varying loop density

The values of these performance metrics for each of the methods compared as a function of the loop density of the vehicles are shown in Fig. 3. We observe that the methods that utilize cooperation between vehicles perform drastically better than the baseline approach, in which vehicles do not cooperate with each other. In Fig. 3b we observe that our method is capable of maintaining low levels of lane density when compared to the other methods. This property becomes more pronounced as the loop density on the highway section increases. While in Fig. 3a, zipper merge is found to have similar throughput levels to our method, the benefit of our method becomes evident when we consider the combined effect of throughput and density which is shown in Fig. 3c. We observe that the effective mean velocity achieved in our method is considerably higher than all the other methods, especially in high loop density scenarios. Our method achieves a 282% improvement in comparison to the baseline case for mean velocity of flow. When compared to zipper merge, an improvement of 81% is achieved at high loop densities. However, we see that at low loop densities (few vehicles on the road) the zipper merge algorithm performs comparably to our method. This follows our intuition that when there are very few vehicles on the road we don't require a complex optimization based algorithm and a simple heuristic based algorithm performs adequately. However, as the traffic on the road increases, which often leads to the formation of bottlenecks at merge junctions, the benefit of our optimization based method becomes apparent.

2) Delay metric results: The overall delay faced by the individual vehicles on the highway is also a key factor in merging control. The values for the delay metric are compared in Fig. 4, in which our method shows considerably better performance. In Fig. 4a which compares the average delay in seconds experienced by vehicles, we observe that our method achieves a 92% and 58% improvement when compared to the baseline case and zipper merge respectively.

Once again this improvement is more pronounced at high loop densities. When considering delay in heterogeneous traffic it is important to consider the impact of delay on different vehicle classes. This is specially true in the case of emergency vehicles (EVs). As the other algorithms tested consider all classes as equal, the delay faced by EVs is the same as the delay faced by other vehicle classes. In contrast, EVs in our approach face 17% less delays than other vehicle classes. For example, at a loop density of 80, the average delay faced by EVs is $13.7\ sec$ in our method and $203.4\ sec$ in the baseline case. Fig. 4b depicts how EVs experience less delays than the other vehicle classes in our method. It is important to note that the proposed method offers adjustable parameters which can be tuned to further reduce EV delay at the cost of reduced mean velocity for overall traffic flow.

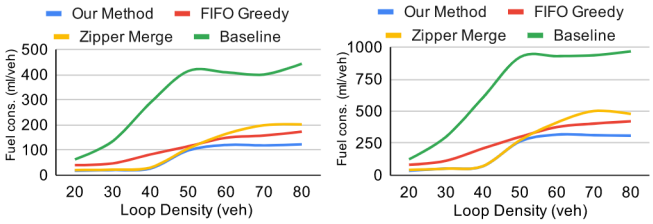


(a) Delay variation (b) EV Delay variation

Fig. 4: Comparison of delay faced with varying loop density

3) Fuel consumption metric results: The effect of these algorithms on overall fuel efficiency is depicted in Fig. 5. Here, Fig. 5a and 5b show the fuel consumption per vehicle in all classes and the truck class respectively. As expected, all cooperation based algorithms have much lower fuel consumption than the baseline case. Additionally, even among the cooperative methods, we see that at higher loop densities

our proposed method yields the best fuel efficiency. While under the current parameters, the fuel savings percentages for trucks are similar to other classes, our method allows tuning the parameters to further reduce the truck class fuel usage, at the cost of reduced overall throughput.



(a) Fuel consumption variation (b) Truck Fuel cons. variation

Fig. 5: Comparison of fuel consumption with loop density

4) Real-time operation: The algorithm in our optimization based approach operates at a frequency of $f_s = 10Hz$ (updates all control commands every 100ms). The time required for the optimizer to compute control outputs for a single sequence was around 25ms. Based on our computation power, in the tests carried out, up to 12 different sequences would be checked in each time step which, if computed sequentially would require roughly 300ms. However, as this algorithm was designed to run each sequence optimization in parallel, the optimization task for all sequences takes less than 50ms on a 12 core CPU. This showcases that the proposed approach is more than capable of real time operation, as this is considerably less than the desired recalculation rate of 100ms It is important to note that the optimality of the generated output control commands can be improved by searching over a larger sample of sequences. However, in order to achieve real time operation this would require more computation resources. This means that if needed, by utilizing a processing unit with more computational resources (e.g. higher core count), the performance of this method can be further enhanced, while continuing to operate in real time.

V. Conclusion

We introduce a novel constrained optimization based approach to cooperative highway on-ramp merging of heterogeneous CAVs. The key advantage of real time operation while abiding to all safety constraints was achieved by a hybrid dual architecture capable of parallel operation. Here, initially we generate multiple merging sequences which are evaluated in parallel by the optimizer to select the best control. Furthermore, this work introduces priority assignment inside the optimization process to cater to the varying requirements introduced by heterogeneous traffic. The performance of this method was verified through simulations using the SUMO platform. Future work in this area would involve understanding the macro-level effects of this algorithm along with the inter-vehicle interactions necessary for multi-lane highways. Another interesting direction would be the creation of a decentralized version of this algorithm based on V2V communication.

References

[1] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[2] K. Chung, J. Rudjanakanoknad, and M. J. Cassidy, "Relation between traffic density and capacity drop at three freeway bottlenecks," *Transportation Res. Part B: Methodological*, vol. 41, no. 1, pp. 82–95, 2007.

[3] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[4] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, 2002.

[5] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 271–281, 2002.

[6] H. Hadj-Salem, J. M. Blosseville, and M. Papageorgiou, "Alinea: a local feedback control law for on-ramp metering; a real-life study," in *Third International Conference on Road Traffic Control*, 1990, pp. 194–198.

[7] M. Sarvi and M. Kuwahara, "Microsimulation of freeway ramp merging processes under congested traffic conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 470–479, 2007.

[8] R. Scarinci, B. Heydecker, and A. Hegyi, "Analysis of traffic performance of a merging assistant strategy using cooperative vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2094–2103, 2015.

[9] T. Awal, L. Kulik, and K. Ramamohanrao, "Optimal traffic merging strategy for communication- and sensor-enabled vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 1468–1474.

[10] J. Ding, L. Li, H. Peng, and Y. Zhang, "A rule-based cooperative merging strategy for connected and automated vehicles," *IEEE Trans. on Intelligent Transp. Systems*, vol. 21, no. 8, pp. 3436–3446, 2020.

[11] J. Rios-Torres, A. Malikopoulos, and P. Pisu, "Online optimal control of connected vehicles for efficient traffic flow at merging roads," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 2432–2437.

[12] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2017.

[13] Y. Zhou, M. E. Cholette, A. Bhaskar, and E. Chung, "Optimal vehicle trajectory planning with control constraints and recursive implementation for automated on-ramp merging," *IEEE Trans. on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3409–3420, 2019.

[14] N. Chen, B. van Arem, T. Alkim, and M. Wang, "A hierarchical model-based optimization control approach for cooperative merging by connected automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[15] S. Li, C. Wei, and Y. Wang, "A ramp merging strategy for automated vehicles considering vehicle longitudinal and latitudinal dynamics," in *2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE)*, 2020, pp. 441–445.

[16] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 900–906.

[17] A. Uno, T. Sakaguchi, and S. Tsugawa, "A merging control algorithm based on inter-vehicle communication," in *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, 1999, pp. 783–787.

[18] Y. Ito, M. A. S. Kamal, T. Yoshimura, and S. Azuma, "Multi-vehicle coordination on merging roads based on pseudo-perturbation-based broadcast control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4008–4013.

[19] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, November 2018, pp. 2575–2582.

[20] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com