Optimizing Departures of Automated Vehicles from Highways while Maintaining Mainline Capacity

Eric Meissner, Thidapat Chantem, *Member, IEEE*, and Kevin Heaslip

***Abstract*—Automated vehicles have the potential to revolutionize our nation's transportation systems as they promise to dramatically reduce congestion, accidents, and fuel usage. Namely, as it becomes possible to precisely exert the control on, and coordinate the movement and placement of, vehicles along a stretch of highway, the separation distance between vehicles can be reduced, thus increasing flow and minimizing congestion. Precise and coordinated vehicle controls and placements also improve predictability, resulting in fewer instances of sudden braking, which reduce fuel usage and accidents. Most existing research has focused on the steady-state behaviors and operations of automated vehicles, such as platooning, assume complete knowledge of the system, e.g., the number of vehicles and their destinations, and neglects dynamic or transition operations such as exiting a highway and lane changing. Uncoordinated lane changing and exiting behaviors by automated vehicles can significantly reduce flow of traffic as vehicles will require larger separations, are forced to slow down, or worse, collide. In this article, we present a collision-free, runtime approach to efficiently organize the departures of automated vehicles from a highway environment while maintaining highway capacity in extremely dynamic conditions. To maximize the number of safe departures, the key ideas are to (1) determine when, and where to, an exiting vehicle should lane change in order to make a successful exit given current traffic conditions as provided by connected vehicle technology, and (2) execute the actual lane change operations using a reservation-based approach. Simulation results show that by coordinating vehicles behaviors, traffic flow can be improved by up to 5 times todays typical flow while ensuring a 100% exit success rate in a collision-free manner.**

***Keywords*—Intelligent transportation system, automated vehicles, automated roadway, intelligent vehicle routing.**

I. Introduction

For the past decades, traffic congestion has become a major hindrance to both city dwellers and rural residents. According to the 2012 Urban Mobility Report [1], drivers in the top 15 largest US metropolitan areas experience, on average, 43 hours of traffic delay and waste 20 gallons of fuel each year. Travel time to destination during peak hours is taking 25% longer, and drivers must now allocate between 80% to 250% more time to reliably reach their destinations on time. In addition, carbon dioxide production during congestion is 3.9 times higher than during freeflow traffic conditions [1]. As such, congestion directly leads to decreases in air quality, productivity, and quality of life, as well as a monetary loss of about $922 on average per driver per year [1].

It is difficult to eliminate or even reduce congestion by relying on existing infrastructure, as current freeway systems are being pushed to capacity. Simply expanding freeway capacity is a challenge due to financial, environmental, and land use considerations, especially in urban areas, and may not be an effective solution to congestion [2]. Non-recurring congestion such as vehicular crashes may also occur and further reduces capacity.

Automated vehicles have the potential to greatly reduce both recurring and non-recurring congestion [3]. The complete control of automated vehicles makes it possible for vehicles to follow each other closer than in the case of human drivers, thus inherently increasing capacity. In addition, since 93% of all traffic accidents were directly caused by human errors [6], automated vehicles in highway environments are expected to reduce the rate of traffic accidents by over 90% [7], as well as alleviating the burden on human drivers. Since traffic crashes continue to be one of the top 10 leading causes of death in the US, with 5.5 million crashes per year (with over 30,000 being fatal), use of automation will also be safer. Last but not least, widespread automated vehicles are expected to be more fuel efficient per mile, at the trade-off of a possible increase in vehicle miles traveled from increased convenience and availability of shared vehicles to underserved populations [8].

Most existing automated vehicle coordination designs and analyses have focused on vehicle platooning [9], [10], where several automated vehicles form a group and follow each other at a close distance. Dynamic or transition operations such as lane changing or freeway exiting have largely been ignored but could significantly affect traffic flow if performed in an ad-hoc manner [11], [12], [13], [14]. According to Tsao, et al. the exit success percentage, which is the number of vehicles successfully exiting a freeway at their intended destination, is well below 100% [3]. This is not only because a single automated lane was considered, but also due to less efficient usage of openings on the highway. Clearly, low exit success percentage is unacceptable to consumers and may hinder a wide-scale adoption of automation. To make a step towards the realization of fully automated freeways and, more specifically, reduce congestion on freeways, we focus our attention on optimizing the behaviors of automated vehicles that need to make an exit from the highway in order to maintain mainline capacity. The key challenge lies in lane changing to the exit lane (if required) with minimum interference to other nearby vehicles and without dangerous and fuel-hungry maneuvers such as excessive acceleration or deceleration.

E. Meissner and T. Chantem are with the Department of Electrical and Computer Engineering, Utah State University, Logan, UT, 84322 USA, e-mail: eric.meissner@aggiemail.usu.edu, tam.chantem@usu.edu. K. Heaslip is with the Department of Civil and Environmental Engineering, Virginia Polytechnic Institute and State University, Arlington, VA, 22203 USA, email: kheaslip@vt.edu.

Manuscript received ...

In this article, we present our approach for allowing automated vehicles to successfully and efficiently exit from a highway without unnecessarily slowing down traffic flow on the highway. Our main contributions are three-fold.

1. We propose a coordination-based algorithm to maximize the number of successful exits by automated vehicles while maintaining mainline capacity and fuel efficiency. Our approach does not require prior knowledge of a vehicle's origin nor destination.
2. To evenly distribute traffic and, hence, improve flow and capacity, we present a mechanism to arrange vehicles in different lanes according to their destination exits. The proposed mechanism also makes it easier for exiting vehicles to leave the highway with minimum interferences on other vehicles.
3. We design, develop, and test a simulation framework, which is used to assess the performance of the proposed approach under varying road and operating conditions.

The rest of the paper is outlined as follows. In Section II, we review existing key literature. Section III provides the system model under consideration. The overview of our approach is given in Section IV and the detailed descriptions in Sections V–VII. Section VIII analyzes the advantages and potential drawbacks of the proposed approach based on simulation data. Section IX concludes the paper with possibilities for future work.

II. Related Work

While there are similarities in existing, manual traffic flow models ([15], [16], [17]), Li et al. showed that traffic flow models of automated vehicles are substantially different, motivating a need for novel traffic flow control systems. Traffic flow control plays a crucial role in eliminating highway congestion and has thus received significant research attention, e.g., [18], [19]. For instance, Chien et al. utilize both vehicle and roadside controllers to monitor and control traffic density along a strech of highway to improve flow [19].

Lane changing is one of the most important and difficult maneuvers a vehicle can make on the highway. When performed in an unorganized fashion, lane changing has been shown to negatively impact traffic flows [11], [12], [13], [14]. Broucke et al. presented a traffic flow theory in automated highway systems and observed that the coordination of vehicles' entrances into a highway can help to improve flows [18]. Several controller designs exist for safe lane changing [20], [21], [22]. The problem of lane assignment has been considered by several researchers [23], [24], [25], [26], [27], [28]. Most existing work adopted a static lane assignment approach [26], [27], [24], [25]. The work by Ramaswamy et al. includes lane assignment to vehicles on automated highway systems [24], [25]. Such an assignment was shown to be optimal but involves solving an origin-destination matrix for the set of vehicles on the highway, which is computationally inefficient and too complex to recompute in the event that vehicles change destinations while on the freeway, or in the event of an anomaly on the roadway such as an accident. Traffic patterns that change throughout the day, such as rush hour vs. night, were accounted for by iteratively solving linear programs [26], [27] and using genetic algorithms [29]. Unfortunately, the use of LP solvers and genetic algorithms at runtime is still computationally intensive. More importantly, it is unclear how successful lane changing is actually ensured (successful lane changes are typically assumed regardless of traffic conditions) and how destination changes, for example, are handled in existing work.

Desiraju et al. discussed a technique to maximize the number of safe lane changes that can be made on a highway in a given time duration [30]. However, there is a lack of research on determining when, how, and why vehicles should make lane changes, especially while ensuring a high exit success percentage. Such insights would, in turn, lead to more efficient, organized, and coordinated automated vehicles. The proposed work is also more general than existing work in that it makes use of judicious lane assignment, lane changing, and load balancing to achieve 100% exit percentage under reasonable road conditions, and is less prone to complete failure (in finding a feasible solution) since it adopts a heuristic approach that is computationally efficient.

Another key concept in automated vehicle coordination is the platooning of vehicles. Platooning involves tightly coupling vehicles into groups that communicate and move along the roadway acting as one larger vehicle. Treating vehicles as a group enables the vehicles to travel at highway speeds at a much closer distance than if they were to travel individually without tight coordination. Platooning is expected to provide major fuel savings to commercial trucking [31]. While we limit our discussion to single automated vehicles in this paper for the sake of clarity, existing research has shown that platoons can make synchronized lane changes as if they were single vehicles [32], [33]. Hence, the proposed approach can be extended to platoons in addition to individual automated vehicles.

III. System Model and Assumptions

We present our system model, state our assumptions, and formally define the problem in this section.

A. Highway Model

First, we assume that highway conditions are homogeneous. We partition the highway under consideration into multiple stretches of fixed length. Given a stretch of highway with m lanes in each direction, we define a set of lanes $\mathbf{L} = \{L_1, \ldots, L_m\}$, where lanes L_1 and L_m are the exit and innermost lanes, respectively. The width of each lane is known *a priori*. Vehicle i stands on lane l_i on road stretch k_i. When in relation to a particular vehicle, the pair (l, k) refers to the center point of the vehicle within the lane. Exit positions can be described as $(0, k')$. Exit queues are not explicitly considered in this work.

Each lane L_l is associated with a steady-state velocity v_l [34]. In order to incentivize vehicles to use all lanes of a highway and, hence, increase traffic flow, the velocities of inner lanes are higher than those of the outer lanes, i.e.,

$$v_1 < v_2 < \cdots < v_m. \tag{1}$$

A vehicle that is not actively performing a lane change while in lane L_l travels at v_l. For safety reasons, a vehicle's velocity and acceleration must not exceed the minimum or maximum velocity or acceleration, v_{min} , v_{max} , a_{min}, and a_{max}, respectively. Finally, we assume the existence of roadside equipment, which allows the infrastructure to communicate traffic updates and lane changing commands with the vehicles on the highway.

B. Vehicle Model

Vehicles are fully automated, use vehicle-to-infrastructure (V2I) communications [4], [5] to make maneuvering decisions, and may be heterogeneous in size and driving characteristics. The set of vehicles under consideration is denoted as $\mathbf{V} = \{1, \cdots, n\}$, with an arbitrary vehicle referred to as vehicle i. There are no restrictions on the lane an exiting vehicle may be in before it can initiate its departure process. When not undergoing the exiting process from a highway, a vehicle travels at the steady-state velocity which is dictated by the lane it is currently in, as described earlier. The frequency and number of vehicles entering a stretch of highway is arbitrary. (In Section VIII, we conducted simulations to test the limit of our systems in terms of maximum flow rates.)

We use a standard freeway section from traditional highway capacity analysis without ramps, assuming that the vehicles rely on some existing methods [35] to safely enter the highway. That said, newly entered vehicles may eventually change lanes as dictated by our proposed approach to improve flow and make it easier for exiting vehicles to leave the highway.

Over a given time period, vehicles traveling in lane L_l at speed v_l cover more distance than those traveling in lane L_{l-1} at speed v_{l-1} where $v_l > v_{l-1}$. Formally, a vehicle i is described by the size and safety buffer distance s_i from its center, velocity v_i, and acceleration a_i. When traveling steady-state in lane L_l, $v_i = v_l$ and $a_i = 0$. As stated earlier, the position of the vehicle i is given by l_i and k_i, namely the lane and stretch occupied by the vehicle. Finally, $e_i = k'$, $k' \geq k_i$, is the stretch of road representing the vehicle's destination exit. Assuming that the value for k increases as a vehicle moves forward, the desired exit position of a vehicle (if it has not yet missed its exit) is $k' \geq k$. Conversely, for a vehicle that has missed its exit, $k' < k$.

C. Problem Definition

<u>Problem 1:</u> Given a stretch of a highway with V2I communications, and a set of n automated vehicles as described earlier, coordinate the behaviors of the vehicles in such a way that the number of successful highway exits is maximized while secondarily minimizing fuel usage associated with exiting and lane changing maneuvers.

IV. System Overview

To exit from a highway, a vehicle may need to lane change and speed up/slow down. To lessen the impacts of such dynamic maneuvers on neighboring vehicles, we propose coordinating exiting efforts among the vehicles iteration by iteration. We define a time period during which only vehicles that are deemed eligible at the beginning of the period may request a lane change during said period (though such a request may not be granted). Ineligible vehicles must wait until at least the beginning of the next time period. Said time period is a user-defined parameter and is kept constant from one iteration to another unless road conditions change. We will discuss the selection of an appropriate time period for an iteration at the end of this section.

We divide each iteration into three main phases, during which all vehicles on the aforementioned stretch of highway are considered. Details on each phase can be found in Sections V–VII.

1. **Phase 1 - Determine eligible vehicles:** Only vehicles that are close to their destination exit or which wish to move to an inner lane to balance traffic are considered for lane changing. To minimize overhead and allow users to change their mind, vehicles are responsible for requesting a lane change instead of having the infrastructure determine eligibility in a global manner.
2. **Phase 2 - Decide destination position for each eligible vehicle's lane change:** The infrastructure takes the set of eligible vehicles and determines where (i.e., what position) a vehicle may potentially lane change to. In this step, the objective is firstly to maximize the number of potential lane changes and secondarily to minimize fuel usage while preserving the total number of lane changes.
3. **Phase 3 - Select vehicles to lane change:** Since not all lane change requests may be honored given the destination positions determined in the previous step, i.e., collisions may occur, the infrastructure assigns a priority to each eligible vehicle to select the set of vehicles allowed to lane change in the current iteration. The priority of a vehicle is based on the distance to the destination exit of that vehicle. In addition, a lower-priority vehicle may be selected for lane changing if its operation will not interfere with those of higher-priority vehicles.

Figure 1 illustrates the main steps of the proposed approach. Vehicles that receive a permission to lane change do so synchronously and complete the maneuver before the start of the next time period, i.e., iteration. Lane changing information is communicated to the vehicles using V2I. An example scenario depicting two consecutive iterations is shown in Figure 2.

Since optimal vehicle routing in general involves solving a large system of differential equations, which is computationally intensive and require a complete knowledge of the destination exit of all vehicles, we adopt a heuristic approach to efficiently solve the problem at runtime. Our approach determines when and where vehicles should lane change to in a distributed manner while leveraging V2I to ensure safety.

We now return our discussion to the selection of the period of an iteration T. The goal is to choose an iteration period long enough to facilitate a large number of lane changes but not so long as to impede a vehicle's ability to make multiple consecutive lane changes to reach an exit in time. Consider a vehicle i that currently resides at position (l, k) and which is to lane change to position (l', k'), $k < k'$, where $|l - l'| = 1$. Assuming that i uses the maximum acceleration and velocity to complete the lane change as quickly as possible, the time

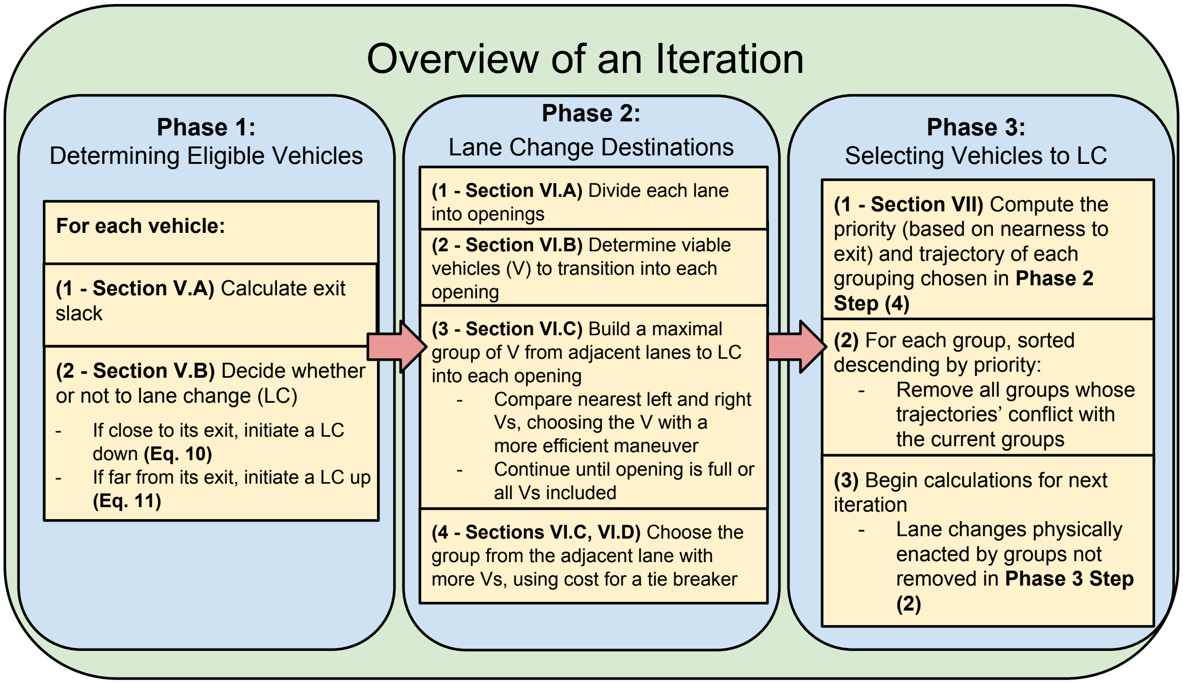


Fig. 1: The major steps of the proposed approach.

to lane change from (l, k) to (l', k') can be expressed as

$$t = t^a + t^v, \tag{2}$$

where t^a and t^v denote the time it takes for i's velocity to reach v_{max} and the time required at v_{max} to reach its destination position, respectively. Mathematically,

$$t^a = \frac{v_{max} - v_l}{a_{max}} \tag{3}$$

$$t^v = \frac{d - x_a}{v_{max}} \tag{4}$$

$$x_a = v_{max} \cdot t^a + \frac{1}{2} a_{max} \cdot (t^a)^2, \tag{5}$$

where d is the distance between (l, k) and (l', k') and can be found by solving for the shortest path distance using Pythagorean theorem. Note that if $x_a \geq d$, t is obtained by solving the following equation

$$d = v_{max} \cdot t^a + \frac{1}{2} a_{max} \cdot (t^a)^2. \tag{6}$$

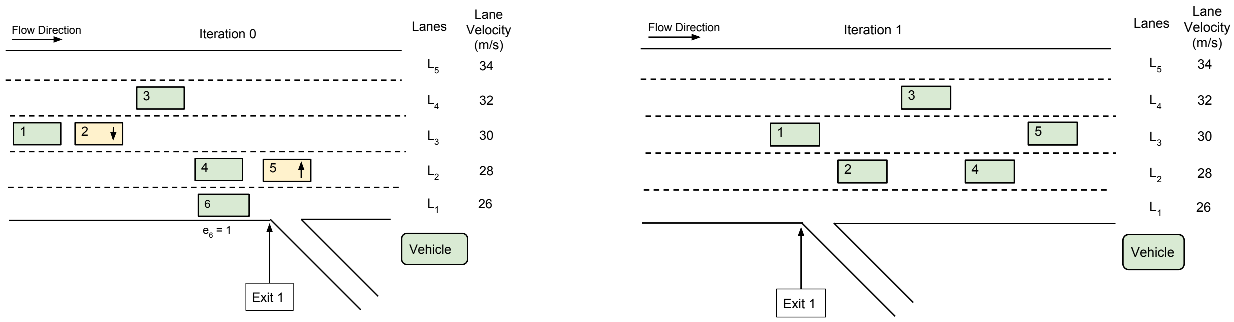
Then,

$$T = \max \{t | \forall L_l, L'_l \in L, |l - l'| = 1, \\ k_{threshold} = k' - k\}, \tag{7}$$

where $k_{threshold}$ places a limit on the maximum distance a vehicle can travel in a single iteration. Guard bands may be added to T to allow lane changing vehicles to reach the steady-state velocities of their new lanes. Note that vehicles are not required to make lane changes at maximum acceleration and velocity in practice, the maximum acceleration and velocity are only used to in the selection of a reasonable iteration duration. In each iteration, lane changes take place at the same time as the execution of the three major steps in the proposed approach (Figure 1); lane changes calculated by the latter in the current iteration will be carried out during the next iteration, and so on.

V. Phase 1 - Determining Eligible Vehicles

As stated in the last section, the proposed approach is iterative. In a given iteration, each vehicle independently determines whether it needs to make a lane change request. There are two reasons for which a lane change may be desired: (1) the vehicle is approaching its destination exit, and (2) the vehicle wishes to transition to an inner lane to travel faster and even out traffic flow across the lanes. That is, a vehicle that remains in lane L_1 for the entire trip on the highway never misses an exit (except, perhaps, for equipment malfunction) but will not have the incentive to do so, as vehicles in inner lanes are allowed to travel at a higher velocity. Clearly, there are situations where it makes intuitive sense for a vehicle to remain in lane L_1, e.g., when its destination exit is quickly approaching. To maximize the number of successful exits while maintaining mainline capacity by evening out traffic across the lanes, we propose equipping each vehicle with a module that will allow the former to determine whether a lane change up (L_l to L_{l+1}) or down (L_l to L_{l-1}) is desired. We focus our attention on changing from lane L_l to lane L_{l+1} or lane L_{l-1}, as a lane change from L_l to L'_l, $|l' - l| > 1$, can be



(a) During the first iteration, vehicles 2 and 5 wish to lane change, as denoted by the arrows, are determined to be eligible for lane changing, and are granted permissions to change lanes. Vehicle 6 is about to make an exit.

(b) 2 and 5 have completed their lane change operations at the start of the second iteration.

Fig. 2: An example scenario illustrating two iterations of the proposed approach.

accomplished using a number of single lane changes. Note that in this step, because vehicles independently determine whether a lane change request is needed, calculations are performed locally in an efficient, distributed manner.

As a general rule, vehicles should be encouraged to use the inner lanes as much as possible for better flow while moving toward an exit lane in a timely manner as to permit successful exits. A main challenge lies in adapting to the dynamic nature of the traffic on the stretch of highway under consideration. For instance, a vehicle may need to request a lane change down earlier in heavier traffic to make a successful exit. To tackle this challenge, we propose an exit-slack based approach to balance between shorter travel times and successful exits. Using said approach, vehicles will be able to determine whether a lane change request should be made to the infrastructure in a given iteration.

A. A Slack-Based Approach to Lane Changing

Since making a successful exit (or transitioning into an inner lane in a timely manner) depends on the time it takes to perform lane changing, which, in turn, depends on traffic conditions, we define a function $\epsilon(t,l)$ to reflect the urgency of a lane change. Specifically, during the t^{th} iteration, $\epsilon(t,l)$ is used to approximate the distance to be traveled before a lane change request from lane L_l is granted based on the current traffic flow. Formally,

$$\epsilon(t,l) = R(t) \cdot T \cdot v_l, \tag{8}$$

where $R(t) = \alpha \cdot \frac{R^{t-1}}{M^{t-1}}$ and is the ratio of the number of vehicles that requested a lane change in the $(t-1)^{th}$ iteration, R^{t-1}, to the number that were actually allowed to make a lane change, M^{t-1}. The variable $R(t)$ denotes the likelihood that a vehicle will be granted the permission to make a lane change during the t^{th} iteration, if such a request is made, based on success history during the $(t-1)^{th}$ iteration. As fewer vehicles are allowed to lane change, $R(t)$ increases, indicating that a vehicle may be required to wait a longer amount of time before it is allowed to change lanes. The parameter α is user-adjustable; a lower value for α implies the user (or traffic engineer) is more willing to risk missing an exit as long as they can travel at a higher velocity. We present an analysis of the impacts of different α values in Section VIII.

Clearly, the value of $\epsilon(t,l)$ may change every iteration. For a given iteration, $\epsilon(t,l)$ is the same for all vehicles in lane L_l. Defined as such, a higher value for $\epsilon(t,l)$ will result in vehicles requesting a lane change earlier to compensate for the expected delay associated with lane changing and increase their chances of successful exits. In addition, if $\epsilon(t,l)$ is smaller, exiting vehicles are less likely to request an earlier lane change, which would otherwise result in them having to travel at a slower speed for a longer time duration. Conversely, vehicles wishing to move to an inner lane would want to do so as early as possible to maximize the time spent traveling at a higher velocity.

We leverage $\epsilon(t,l)$ to form the basis for our slack distance calculations, which will be used by exiting and entering vehicles alike (see the next subsection). Namely, let d_i be the exit-slack distance of vehicle i, which is an estimate on the total distance traveled by i during the times i will spend waiting for lane change requests from the current lane l_i to the exit lane L_1 to be approved. In other words,

$$d_i = \sum_{l'=1}^{l_i} \epsilon(t,l'). \tag{9}$$

For example, if $l_i = 3$, then $d_i = \epsilon(t,3) + \epsilon(t,2) + \epsilon(t,1)$. Figure 3 provides an example scenario and resultant values for $\epsilon(t,l)$ and d_i. It is important to note that the exit-slack distance is an estimate and may be underestimated. We will use simulation data to discuss the effectiveness of using the exit-slack distance to achieve our objective of maximizing the total number of successful exits in Section VIII.

B. Determining When to Request a Lane Change

We now discuss our algorithm for determining when a vehicle should request a lane change by leveraging the exit-

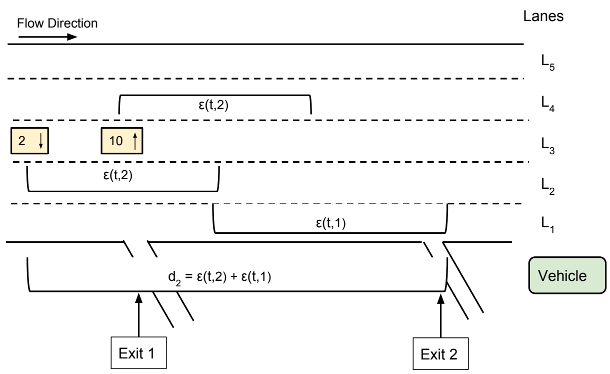


Fig. 3: An example depicting $\epsilon(t, 2)$ for vehicles in lane L_2 for the t^{th} iteration. The distance $\epsilon(t, 2)$ denotes an estimate of the distance 2 and 10 might travel before their lane change requests are granted. The value for the exit-slack d_2 is also shown and is the sum of $\epsilon(t, 2)$ and $\epsilon(t, 1)$.

slack distance concept discussed in the last subsection. There are two cases. In the first case, a lane change down may be required in the event that the estimated distance for a vehicle to lane change to the exit lane is approaching the distance to that vehicle's destination exit. In such a case, the vehicle should request a lane change in the current iteration if the following constraint is satisfied.

$$e_i < k_i + d_i \tag{10}$$

Eq. 10 contains a strict inequality to maximize the chance of a successful exit. In other words, the purpose behind Eq. 10 is to ensure that a lane change request is made in a timely manner.

A vehicle may request a lane change up to travel at a faster velocity, and, inherently, balance traffic flow across the lanes. This lane change request should only be undertaken if doing so will unlikely cause the vehicle to eventually miss its exit, i.e., the exit-slack distance of said vehicle is large enough. We modify Eq. 10 to include the estimated distance required to make a lane change up and the corresponding lane change down that will be necessary to make an exit. Namely,

$$e_i < k_i + d_i + \epsilon(t, l_i + 1) + \epsilon(t, l_i). \tag{11}$$

If Eq. 11 is satisfied, i should request a lane change up during the t^{th} iteration, as moving into a faster lane typically allows a vehicle to reach its destination in less time and more evenly distribute traffic across lanes. The pseudocode for the algorithm when a vehicle should request a lane change is given in Alg. 1. For the sake of clarity, some checkpoints are omitted. Finally, in the event that a vehicle changes its destination, such a change is communicated to the infrastructure and reflected when applying our approach in the next iteration; no other changes are required, as our approach computes eligibility for lane changing in each iteration based on destination exits.

Algorithm 1 Lane_Change_Request($\mathbf{L}$, $\mathbf{V}$, t)

```
for L_l ∈ L do
    compute ε(t, l) as in Eq. 8
for L_l ∈ L do
    for i ∈ V do
        compute d_i as in Eq. 9
        if Eq. 10 is satisfied then
            request a lane change from L_l to L_{l-1} for i
        else if Eq. 11 is satisfied then
            request a lane change from L_l to L_{l+1} for i
        else
            do nothing
```

VI. Phase 2 - Deciding Destination Positions for Eligible Vehicles

Once eligible vehicles have sent their requests for lane changing to the infrastructure, the latter is required to tentatively assign a destination position for each eligible vehicle. Recall that such tentative assignment is used in Phase 3 to determine vehicles that are allowed to perform a lane change in the current iteration, and does not necessarily guarantee that every lane change request will be granted. Given a vehicle, its current lane, and its intended lane, the goal of this step is to find a position in the intended lane that the vehicle can feasibly lane change to, e.g., without exceeding the maximum velocity or acceleration and in such a way that the total number of feasible lane changes found is optimized. Each viable destination position for a vehicle considers the safety buffer both ahead and behind the vehicle, and vehicles are not assigned positions that would result in an unsafe lane change. In the ideal scenario, a destination position can be found for each eligible vehicle.

While the goal set forth is a challenge to accomplish due to the exponential number of combinations to check, we make an important observation that helps to reduce the complexity of the problem under consideration. Since each lane has a well-defined steady-state velocity, the vehicles within a lane are essentially stationary in relation to one another. Hence, the relative distance between any two vehicles in the same lane is a constant. Taken further, vehicles not seeking a lane change can be viewed as stationary within a lane. To help visualize the scenario, let us consider a board game where each vehicle is a tile. Tiles are organized into columns of infinite size. Each column moves the tiles on it at some pre-defined speed. Vehicles that are not lane changing are tiles whose relative positions are fixed, can be calculated, and cannot be moved. Given this setup, it is not possible for fixed tiles to collide with one another. Vehicles making lane changes are tiles that can be moved left or right (up or down a lane), but which cannot jump over other tiles. Moving tiles can also increase their velocity, or slow down, as long as doing so will not impact the other tiles in front or behind it. A tile shift between columns corresponds to a lane change that assigns a vehicle into an open position in the destination lane. In addition, while the tiles in a given column can be considered fixed since vehicles within a lane travel at the same velocity, tiles from the different columns of the board are not stationary with respect to one another. That is, a tile wishing to shift into a faster moving column will have to speed up, while the reverse may not be necessary. To win the game, which entails shifting all the movable tiles to their respective desired columns, tile shifts must not collide with one another, nor are they allowed to touch a fixed tile. All tile shifts occur simultaneously in a given iteration, which is timed such that all lane changing vehicles finish their maneuvers and operate at at their destination lane velocities at the end of each iteration.

We now describe our approach for deciding the destination position for each eligible vehicle, i.e., where and how to assign a tile to its desired column. We conclude this section with a discussion on how fuel efficiency can be incorporated into our method. References to cost in the following subsections refer to fuel usage, and an explicit method for calculating cost is given at the end of the algorithm discussion.

A. Assigning Open Space

To assign vehicles to positions, one solution is to perform an exhaustive search that assesses every possible combination of vehicles and viable positions. Clearly, such a search is too computationally intensive, especially for runtime use where decisions must be made in a timely manner. To accommodate lane changing of a group of vehicles (**VV**), we define an opening w to be a section of open highway in a single lane of size s_w. To reduce complexity, we propose grouping vehicles together and treating each group as a single vehicle, relying on the fact that groups are built such that lane changes within the group do not collide. A number of vehicles may be grouped together if (1) they are currently in the same lane, (2) they share a common destination lane, and (3) they are currently near one another. For a given **VV**, the corresponding opening must be large enough to accommodate the **VV**, i.e. $s_{\mathbf{VV}} \leq s_w$, where $s_{\mathbf{VV}} = \sum_{i \in \mathbf{VV}} s_i$ is the total length of vehicles in **VV**. The idea, then, is to pair each group of vehicles with an appropriately sized opening that can fit all the vehicles.

By pairing a group of vehicles with an opening, a set of lane changes can be determined and is said to be valid if and only if no two groups of vehicles (and, by extension, no two vehicles) will cross paths if these lane changes were to be carried out. Before detailing our proposed algorithm, we discuss two rules to be followed. **Rule 1:** A vehicle that has not requested a lane change will not be forced to modify its driving behavior, e.g., it will not be forced to speed up, slow down, or change lanes. Likewise, a vehicle that has requested a lane change but is not granted a permission will continue traveling on its current lane with no additional maneuvers. That is, it will not be forced to move out of the way to make room for vehicles allowed to lane change. **Rule 2:** A vehicle can only perform one lane change at a time. In other words, a vehicle currently traveling on lane L_l may not switch to lane L_{l+1}, in order to pass vehicles in L_l, change back to L_l, and move on to L_{l-1}, which is its requested lane. Implicitly, the two rules state that a **VV** may not be assigned to an opening which is blocked by one or more vehicles in the same lane as the **VV**. These rules are enacted not only to reduce the complexity associated with finding a match between a **VV** and an opening, but also for fairness and fuel efficiency reasons. Namely, by not forcing vehicles to change their behaviors, e.g., move out of the way, we ensure that no undue burdens are placed on the other vehicles. At the same time, vehicles will likely be assigned open positions that are closer to them, saving fuel and reducing the chance of collisions.

Our algorithm for assigning vehicles to open positions is given in Alg. 2. Given the stretch of highway under consideration, the set of all openings in a given lane L_l is created (Line 3). For each opening, feasible vehicles from both lanes L_{l+1} and L_{l-1} are determined (Lines 4–5). For the exit lane L_1 (innermost lane L_m, resp.), only the vehicles from L_2 (L_{m-1}, resp.) is considered. **VV**s are then constructed (Lines 7–8) using the Build_**VV** algorithm, detailed in the next subsection.

The best **VV** from the upper lane is then compared to the best **VV** from the lower lane (Lines 9–16). Ultimately, the **VV** that contains more vehicles are selected since our objective in this step is to assign as many vehicles to an opening as possible. Ties are broken in favor of the **VV** with a lower cost (Subsection VI-D), as shown in Lines 17–25. Afterwards, vehicles in the selected **VV** are removed from further consideration and the process is repeated for each opening for all the lanes.

B. Determining Viable Vehicles

The pseudocode for the Build_VV algorithm, which is used by Alg. 2, is shown in Alg. 3. In order to form **VV**s, we must first eliminate vehicles that, for efficiency, fairness, or safety reasons, are not good candidates for lane changing into a particular opening in a given iteration. Let us consider an opening in lane L_l. The vehicles wishing to lane change into L_l must be from either lane L_{l-1} or lane L_{l+1}. We now categorize these vehicles into three distinct sets: vehicles directly adjacent to the opening are in **AV**, whereas vehicles to the left (down) and right (up) of the opening are denoted as **LV** and **RV**, respectively. (Separate **AV**, **LV**, and **RV** are maintained for lanes L_{l-1} and L_{l+1}.) Since vehicles in **AV** are the vehicles nearest to the opening and whose lane changes would be most efficient, they are always considered viable and are put in the set **VV** (Lines 1–2).

Next, vehicles in **LV** and **RV** are considered for their viability to make a lane change into an opening using the following two conditions (Line 3). Figure 4 provides an example scenario illustrating the use of such conditions.

- **Condition 1: Maximum Distance.** First, all vehicles in **LV** and **RV** whose distances from the opening under consideration exceed the maximum distance $k_{threshold}$ defined in Section IV are eliminated from further consideration as they are deemed too far away and would require excessive or forbidden acceleration/speed to lane change into that opening.
- **Condition 2: Blocked.** Second, as a consequence of our fairness rules discussed in the last section, any vehicle that would need to move around a vehicle that is not looking to lane change to reach the opening is removed. Specifically,

Algorithm 2 Assign_Open_Positions($\mathbf{L}$, $\mathbf{V}$)

$\mathbf{solution} \leftarrow \emptyset$
for $L_l \in \mathbf{L}$ **do**
  $\mathbf{O_l} \leftarrow$ set of openings in lane L_l
  $\mathbf{U_l} \leftarrow$ set of vehicles currently in L_{l+1} wishing to change to L_l
  $\mathbf{D_l} \leftarrow$ set of vehicles currently in L_{l-1} wishing to change to L_l
  for $w \in \mathbf{O_l}$ **do**
    $\mathbf{VVUp} \leftarrow$ Build_VV($\mathbf{U_l}$, w)
    $\mathbf{VVDown} \leftarrow$ Build_VV($\mathbf{D_l}$, w)
    if $|\mathbf{VVUp}| > |\mathbf{VVDown}|$ **then**
      $\mathbf{solution} \leftarrow \mathbf{solution} \cup \mathbf{VVUp}$
      for $i \in \mathbf{VVUp}$ **do**
        $\mathbf{U_l} \leftarrow \mathbf{U_l} - i$ // Remove i from further consideration
    else if $|\mathbf{VVUp}| < |\mathbf{VVDown}|$ **then**
      $\mathbf{solution} \leftarrow \mathbf{solution} \cup \mathbf{VVDown}$
      for $i \in \mathbf{VVDown}$ **do**
        $\mathbf{D_l} \leftarrow \mathbf{D_l} - i$
    else
      if $cost(\mathbf{VVUp}) < cost(\mathbf{VVDown})$ **then**
        $\mathbf{solution} \leftarrow \mathbf{solution} \cup \mathbf{VVUp}$
        for $i \in \mathbf{VVUp}$ **do**
          $\mathbf{U_l} \leftarrow \mathbf{U_l} - i$
      else
        $\mathbf{solution} \leftarrow \mathbf{solution} \cup \mathbf{VVDown}$
        for $i \in \mathbf{VVDown}$ **do**
          $\mathbf{D_l} \leftarrow \mathbf{D_l} - i$
return solution

let the k-range of the opening w under consideration be $[k_l, k_r]$, with k_l, k_r denoting the left and rightmost positions, respectively. Given a vehicle i in **RV** with the x-position k_i where $k_i > k_r$, i is eliminated from further consideration if there exists a vehicle not wishing to change lanes j whose k_j satisfies the following constraint: $k_r \leq k_j < k_i$. Similarly, a vehicle i in **LV** is eliminated from further consideration if $k_i < k_j \leq k_l$.

C. Building a Vehicle Group

Recall that **AV**, **LV**, and **RV** are created separately for L_{l-1} and L_{l+1} for a given lane L_l. From the previous subsection, all the vehicles in **AV** have been added to **VV** for a given opening. Our goal now is to fill **VV** with vehicles from the updated **LV** and **RV** sets in order to maximize the number of vehicles that are mapped to the opening w under consideration, i.e., we wish to maximize the final size of **VV** while reducing cost (defined in Subsection VI-D).

The intuition for building an efficient, collision-free vehicle group is that vehicles closest to the opening have the easiest and fastest path to make the lane change. A **VV** is iteratively built by compare the nearest vehicle in **VR** against the nearest vehicle in **VL**. The vehicle which is less costly is selected and the process is repeated until either all vehicles in **VR** and **VL** have been allocated to the **VV**, or the size of **VV** is equal to the number of vehicles the opening can hold.

As long as there is open space left in w, i.e., $s_{\mathbf{VV}} < s_w$ and there are vehicles in **LV** or **RV** (Line 4), we continue the process of filling **VV**. Specifically, we turn our attention to the two vehicles in **LV** and **RV** that are nearest to the opening, namely $nlv = \{max(k_i) | i \in \mathbf{LV}\}$ and $nrv = \{min(k_i) | i \in \mathbf{RV}\}$ (Lines 5–6). If either nlv or nrv is too large to be added to **VV** without making $s_{\mathbf{VV}} > s_w$, we cease choosing vehicles from **LV** or **RV**, respectively, because if the nearest vehicle is too large it cannot be added and as such no other vehicles in that set can make the lane change without violating **Rule 2**. In this case, vehicles from the other set, **LV** or **RV** that was not removed are added to **VV** until it is full.

In the case both nrv and nlv can be viably added to **VV**, there are two possible solutions. Either nlv is added to **VV** at this time or nrv is. The solution that costs less is chosen (Lines 9–14).

To add either nlv or nrv to **VV**, we consider the nearest section of opening large enough to safely contain the new vehicle. For either nlv or nrv , let such a placement be (l, q). There are two cases. In the first case, no part of (l, q) has been assigned to a vehicle, in which case nlv claims it. In the second case, some part of (l, q) has already been assigned to another vehicle in **VV**. In this situation, nrv claims (l, q), and the vehicle that previously occupied (l, q) shifts far enough to free up enough space for the new vehicle to enter, shifting into position $(l, q - s_i)$. For nlv, the vehicle that used to claim (l, q) shifts to position $(l, q + s_i)$ (Lines 7–8). Such a "shift" operation is recursive in that if there is already a vehicle assigned to positions $(l, q - s_i)$ or $(l, q + s_i)$, then that vehicle must also shift. Figure 4 provides an illustrative example of the shift operation.

If we assume all vehicles are of a standard size, the problem reduces to a one-dimensional packing problem where for each iteration of the while loop we pack a vehicle into the opening from either **VR** or **VL**, whichever option is less costly.

After building the two new potential vehicle groups using the left and rightmost vehicles, the **VV** with lower overall cost is chosen as the new **VV** and the added vehicle removed from $\mathbf{VL}/\mathbf{VR}$, respectively. This process continues until either **VL** and **VR** are both exhausted, or $s_{\mathbf{VV}}$ is equal or greater than s_w.

We now state some important properties of our algorithms using a number of lemmas, a corollary, and a theorem. We omit the proofs due to space constraints.

Lemma 1: The Shift algorithm, which is used by Alg. 3, always terminates after having added exactly one vehicle to **VV** after at most $|\mathbf{VV}|$ shifts.

Corollary 1: The worst-case time complexity of Shift is $\mathbf{O}(|\mathbf{VV}|)$.

Lemma 2: For this and all following proofs we assume all vehicles are of a standard size. Let S be the maximum number of vehicles that can fit into the opening under consideration and $|\mathbf{V}|$ be the maximum number of vehicles in a single lane adjacent to the opening. Then, the time complexity of the Build_VV algorithm (Alg. 3) is $\mathbf{O}(|\mathbf{V}|^2 + S^2)$.

While the worst-case time complexity of Alg. 3 is not

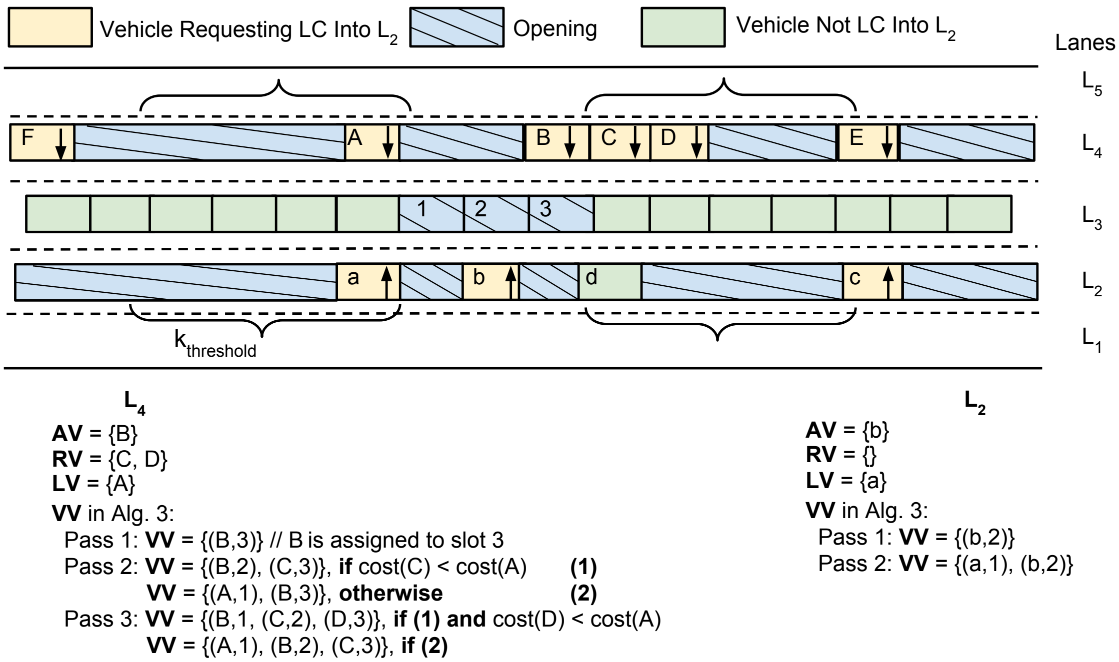


Fig. 4: This figure depicts the vehicles that are considered for a given opening. For simplicity, we assume all vehicles are of the same size. There is space enough for 3 vehicles in the central opening, and we label the individual spaces available as $(3,1),(3,2),(3,3)$ for clarity. From L_4, B is the only adjacent vehicle so $|\mathbf{AV}| = 1$ and $|w|$ is reduced to 2. A is the only vehicle to the left considered, as F is outside the range as dictated by $k_{threshold}$. C and D are considered from the right, but not E as $|w|$ is now 2. From L_2, b is the adjacent vehicle, and a is considered for the lane change from the left. Since d is not lane changing, it blocks c, which is not allowed to lane change despite its intention.

Algorithm 3 Build_VV($\mathbf{V}$, w)

$\mathbf{AV} \leftarrow \{i \in \mathbf{V} | k_l \leq k_i \leq k_r\}$
$\mathbf{VV} \leftarrow \mathbf{AV}$
construct $\mathbf{LV}$ and $\mathbf{RV}$ using Conditions 1 and 2
while $\mathbf{LV} \cup \mathbf{RV} \neq \emptyset$ and $s_{\mathbf{VV}} < S$ **do**
  $nlv = \{max(k_i) | i \in \mathbf{LV}\}$
  $nrv = \{min(k_i) | i \in \mathbf{RV}\}$
  $\mathbf{rvv} = \text{Shift}(\mathbf{VV}, nrv, w)$ // $\mathbf{VV} \cup nrv$, with nrv appended to the right of $\mathbf{VV}$
  $\mathbf{lvv} = \text{Shift}(\mathbf{VV}, nlv, w)$ // nlv is appended to the left of $\mathbf{VV}$
  if $cost(\mathbf{rvv}) < cost(\mathbf{lvv})$ **then**
    $\mathbf{VV} \leftarrow \mathbf{rvv}$
    $\mathbf{VR} \leftarrow \mathbf{VR} - nrv$
  else
    $\mathbf{VV} \leftarrow \mathbf{lvv}$
    $\mathbf{VL} \leftarrow \mathbf{VL} - nlv$
return $\mathbf{VV}$

negligible, we observe that in real-world settings, open space in a lane is unlikely to be contiguous and there will be vehicles directly adjacent to openings, both of which would significantly improve the average-case time complexity of Alg. 3. In addition, it is possible to reduce the length of a stretch of highway to further reduce runtime overhead.

Lemma 3: For a given opening, $k_{threshold}$, and the rules stated in Subsection VI-A, Alg. 3 always returns the **VV** with a maximum number of vehicles.

Note that if vehicles are not assumed to be of identical size, the problem of maximizing the number of vehicles in **VV** becomes computationally difficult, NP-Hard in fact, and our algorithm no longer guarantees a maximum vehicle group for obvious reasons, though does still find a greedy solution.

Theorem 1: The worst-case time complexity of Alg. 2 is $\mathbf{O}(m \cdot S \cdot |\mathbf{V}|^2 + m \cdot S^3)$, where m is the number of lanes on the stretch of highway under consideration, $|\mathbf{V}|$ is the maximum number of vehicles in all the lanes, and S is the maximum number of vehicles that can could fit into the openings.

D. Calculating Cost

In this work, we aim to minimize cost, which is fuel usage. According to Berry, more aggressive drivers use more fuel per mile, where aggression is generally defined as how rapidly and often a driver speeds up and slows down [36]. Such driving behaviors lead to poor uses of gained velocity and frequent speed changes. In the context of automated vehicles, unnecessary acceleration/deceleration during lane changes should be avoided in order to minimize fuel usage. It has been shown that estimates of fuel usage can be reliably derived from instantaneous velocity and acceleration values [37], [38]. Minimizing acceleration and deceleration also serves to maintain passenger comfort during maneuvers.

Since the steady-state velocity of a given lane is fixed in this work, a lane change from L_l to L_{l+1}, for instance, would begin with the involved vehicle starting out at v_l and eventually ending at v_{l+1}. As a result, the fuel cost associated with a lane change can be defined as the fuel used during the transition period, which would mostly be due to the applied acceleration/deceleration. Intuitively, the cost to perform a lane change up into an inner lane would be larger than that required to lane change down into an outer lane, since vehicles in an inner lane travel at a higher speed. That said, it is important to encourage vehicles to move into an inner lane in order to balance traffic across the lanes. For these reasons, we define the cost of an individual lane change to be the average acceleration applied during the transition period, i.e., starting from the steady-state velocity of the departure lane to the steady-state velocity of the destination lane. Note that it is possible for a vehicle to exceed/fall below the steady-state velocity of the departure/destination lane in order to make a lane change and the accelerations/decelerations used are included when computing the average. Formally, the cost of a lane change is

$$cost(i, l_i, l'_i) = \bar{a}(i, l_i, l'_i) - a_\delta(i, l_i, l'_i). \tag{12}$$

where l_i and l'_i denote the departure and destination lanes of i, respectively, $\bar{a}(i, l_i, l'_i)$ is the average acceleration used to execute a lane change, and $a_\delta(i, l_i, l'_i)$ is the base acceleration needed to change from v_{init} to v_{final}. This base acceleration is the average acceleration it would take a vehicle to go from v_{init}, to v_{final} in the course of a single iteration. Consequently, it is more cost effective for a vehicle to lane change into a nearby position instead of one that is farther away, as all lane changing maneuvers must be completed at the end of an iteration. It follows, then, that the cost of all the lane changes in a **VV** is the sum of all the individual lane changes, i.e., by a set of vehicles **V'**,

$$cost(\mathbf{V'}) = \sum_{i \in \mathbf{V'}} cost(i, l_i, l'_i). \tag{13}$$

VII. Phase 3 - Selecting Vehicles to Lane Change

In the last step of the proposed approach, **VV**s have been tentatively assigned to openings in such a way that both maximizes the total number of vehicles assigned a position to lane change to and guarantees no collisions between vehicles in the same **VV**. The challenge, now, is to ensure that no collisions will occur among different **VV**s. We propose a priority-based approach to tackle this challenge. Specifically, a priority value is assigned to each vehicle based on its urgency associated with lane changing, e.g., to make a successful exit. A priority value is then assigned to each **VV** based on the highest-priority value it contains. The goal is to allow for simultaneous, collision-free lane changes of the highest-priority vehicles.

To accomplish this step, we leverage our previous work [30], which aimed to maximize the total number of successful lane changes. The main difference between this work and the previous work is that the latter made no attempt to coordinate the behaviors of the vehicles to achieve a common goal; it merely facilitates as many lane changes as possible without accounting for the fact that some vehicles may miss their destination exits. In the original algorithm [30], a stretch of a highway is divided in a series of 3 contiguous-lane highways. For each 3-lane highway, only vehicles from the left and right lanes wishing to lane change into the center lane are considered. Of these vehicles, the one in front is the first vehicle selected for lane changing. All vehicles wishing to change lanes but whose operations would conflict with the first vehicles cede the right to lane change until the next iteration. The next vehicle that is selected for lane changing in the same iteration is the first vehicle downstream that wishes to change lanes and whose operation would not conflict with the vehicle already selected, and so on.

Instead of granting vehicles the permission to lane change based on their positions, we propose modifying the existing algorithm by directly considering the urgency to lane change associated with a given vehicle. Specifically, a higher priority is assigned to a vehicle as it approaches its destination exit, since the vehicle would have less leeway in making a lane change and, subsequently, its exit. Such a priority-based scheme can also be extended to allow for emergency exits required by the passengers due to low remaining fuel level or unexpected lane merging. More formally, each vehicle i is associated with a priority value p_i, which is the ratio of the exit-slack distance of i and the actual distance to i's destination exit. In other words,

$$p_i = \frac{d_i}{e_i - k_i}. \tag{14}$$

As stated earlier, the priority of a **VV** is taken to be the highest priority of all the vehicles in that **VV**. A **VV**'s priority is then compared to those that have tentatively been assigned to nearby openings, as determined in the last step. In case of conflicts, the highest-priority **VV** is selected. We expect the computational overhead associated with this step to be low, as the total number of **VV**s is expected to be much fewer than the total number of vehicles wishing to make a lane change. In the worse case, this step will require $O(|\mathbf{VVS}|^2)$, where $\mathbf{VVS}$ denotes the set of all the **VV**s in this iteration.

We will rely on existing work [30] to ensure the safety of a given lane change given derived acceleration and velocity values. Since only non-conflicting **VV**s are selected to proceed with the actual lane changes and since vehicles in a given **VV** are guaranteed to not interfere with one another during lane changing, our proposed approach is guaranteed to be collision free.

VIII. Results

Due to the difficulty involved in a large-scale deployment of fully automated vehicles, we opted to perform extensive simulations to assess the performance of the proposed approach. We discuss the results in this section.

TABLE I: Simulation Parameters

Parameter	Value(s)
Length of highway segment	9 km
Number of lanes (each direction)	5
Number of exits/entrances	16
Distance between exits/entrances	0.5 km
Lane Velocities	v_1: 26 m/s v_2 : 28 m/s v_3 : 30 m/s v_4 : 32 m/s v_5 : 34 m/s
Carry-over traffic [1]	20%
Length of a vehicle [2]	$<$ 5 m
Velocity range	21–39 m/s
Acceleration range	2.8–3.2 m/s[2]
Iteration time T [3]	11.35 s
α [4]	1.5
$k_{threshold}$ [5]	75 m

A. Setup

For simulation purposes, we assumed that vehicles are homogeneous in size and driving characteristics (recall that the proposed approach can readily be used in scenarios where vehicles are heterogeneous). The length of a vehicle plus the safety following distance was no more than 5 m and is comparable to the safety distance used by Marinescu et al. [39]. The stretch of highway under consideration was approximately 9 km long with an exit/entrance every 0.5 km starting at the 2.5 km marks, for a total of 15 exit/entrance pairs, to model urban roadway conditions. On this stretch of the highway, there were 5 lanes in each direction. Lane velocities range from 26 m/s to 34 m/s, which are consistent with current highway velocities. The steady-state velocity v_1 of the outermost lane L_1 is 26 m/s whereas that of L_5 is 34 m/s. The parameters used are summarized in Table I and were extrapolated from the parameters used in existing work [3], [36].

[1]Percentage of vehicles that were already on the highway before the simulations began. The vehicles' initial positions and destination exits were randomly generated.

[2]Including safety buffers.

[3]See Eq. 7.

[4]See Eq. 8.

[5]See Subsection VI-B, Condition 1.

The destination exit of a vehicle was randomly generated using a uniform distribution, with all the vehicles exiting before the end of the stretch of highway under consideration. Vehicles either entered the highway through one of the entrances or were already traveling on the highway at the start of a simulation. In the latter case, the initial positions of these existing vehicles were randomly generated. Simulations were conducted for varying flows, ranging from moderate traffic (1,800 vehicles/hour/lane) to highly dense traffic (23,400 vehicles/hour/lane). In today's driving conditions, the typical maximum traffic flow is about 2,400 vehicles/hour/lane [40]. Since it is expected that automated vehicles will significantly increase flows, the aforementioned flow range was used in our simulations. Below, we describe and discuss the critical flow beyond which exit performance of the vehicles are severely degraded and which should be avoided.

It is worth noting that since lanes have fixed steady-state speeds, the traditional traffic theory relations of Greenshields Fundamental Diagram of Traffic Flow between density, flow, and velocity can be simplified [17]. That is, an increase in traffic flow directly corresponds to an increase in density. In fact, density in our system is identical to flow. Flow, which is the number of vehicles/hour/lane that enter the roadway, can be controlled using ramp metering, as is commonplace in urban highways, and thus calculated. In our system, traffic density can be defined as $\gamma = \frac{vehicles/hour/lane}{meters/hour/lane}$. Because the rate of the roadway's parameterized speed is constant, higher rates of entering vehicles imply a larger γ. That is, if the rate of vehicles entering is exactly the rate that a lane moving at the set speed can maximally allow, then there is no open space. In such a case, both the maximal flow and density is reached as (1) there is no open space to allow for more vehicles, and (2) the flow rate of vehicles can no longer increase. Consequently, the maximum flow occurs when $\gamma = 1$. Since all the vehicles plus their safety buffer in the simulation are no larger than 5 m, and since the average steady-state velocity across lanes is 30 m/s, the maximum theoretical flow is bounded at $\frac{30m/s}{5m/vehicle} \cdot 60s \cdot 60m = 21,600$ vehicles/hour/lane$=108,000$ vehicles/hour across the entire roadway, with a static capacity of 200 vehicles/km/lane. This calculation determines the maximum flow between any two points on the roadway. However, because vehicles can choose to exit anytime after they enter the freeway, the end-to-end flow of the freeway can be much larger. In the extreme, if all vehicles that entered the freeway at entrance i exited at exit $i+1$, then the overall flow of the freeway could be no larger than the number of exits multiplied by the flow between any entrance/exit pair. In our case this bounds the maximal flow at 16 exits $\cdot 108,000$ vehicles/hour $= 1,728,000$ vehicles/hour across the roadway.

B. Comparison Points

To draw meaningful conclusions on the effectiveness and efficiency of our proposed algorithm, we would ideally compare its performance to existing work. However, since, to the best of our knowledge, we are the first to define and solve the problem of maximizing the number of successful exits from a highway under dynamic conditions, there are no existing algorithms that can be used for comparison purposes.

To provide an upper bound on how close the solutions in Phase 2 (2) are to the optimal grouping solution, we implemented a brute-force approach which guarantees the optimal grouping of vehicles as well as their opening assignment. Such an approach was too computationally intensive for our benchmarks with 9 kilometers of roadway (the computer used in simulations ran out of memory and crashed). We were able to compare the brute-force approach against Alg. 3 on a mini-benchmark of length 0.5 km, consisting of 3 lanes and a single exit/entrance). For all 500 iterations, Alg. 3 was able to find the optimal grouping/lane assignment in a fraction of the time.

C. Results

We now discuss the exit failure rates of the proposed algorithm and analyze the time overhead associated with our approach.

1) Exit Failure Rates: We use the exit failure rate EFR as the main metric to assess the performance of the algorithms. EFR denotes the ratio of CM, the number of vehicles that have missed their destination exits in a given iteration to make an exit, and CE, the total number of vehicles wishing to make an exit during that same iteration. In other words,

$$EFR = \frac{CM}{CE + CM}. \tag{15}$$

Correspondingly, the exit success rate ESR is

$$ESR = 1 - EFR. \tag{16}$$

The exit failures rates for the different flows are presented in Figure 5. As shown, using our proposed algorithm results in perfect exit success rates at traffic flows that are up to 4.5 times the current maximum flows. Using our algorithm, a 100% exit success rate is maintained for flows of 12,600 vehicles/lane/hour and below, with the EFR only rising to between 2.4–7.1% even with the highest flow rate of 21,600 vehicles/hour/lane.

Tsao et al. have previously reported a 7% drop in exit success rate per exit with a flow of 2,300 vehicles/hour/lane on a 3-lane highway, though only one lane was fully automated with the remaining lanes being occupied by human drivers [3]. Clearly, our proposed approach presents a substantial improvement over both existing automated and non-automated roadways. Such an improvement is due in large part to the proposed coordination scheme since (1) vehicles not changing lanes are never affected by those seeking a lane change, and (2) vehicles are judiciously distributed across the lanes, allowing a high flow rate to be maintained. At the same time, vehicles making lane changes can rely on the predictable behaviors of others on the highway, resulting in safer maneuvers.

2) Time Overhead: To assess the suitability of using the proposed algorithm at runtime, we examine the time overhead associated with our algorithm. Note that the acceptable time overhead would only need to be smaller than the duration of an iteration, as actual lane changes during an iteration and computations for the next iteration can take place simultaneously

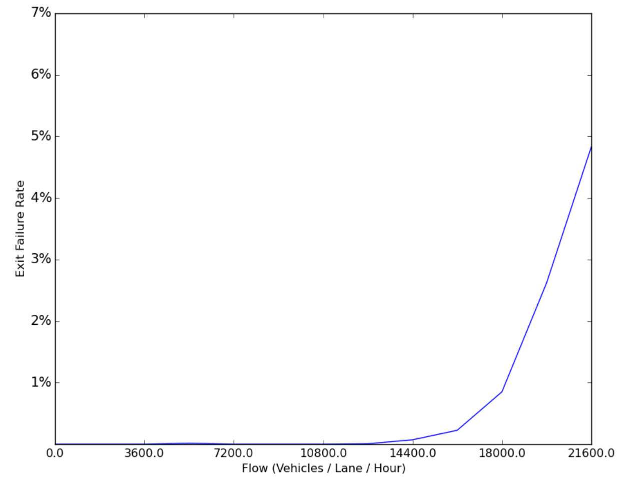


Fig. 5: Exit failure rate as a function of flow.

assuming negligible communication overhead. (If the latter is a concern, the length of an iteration can be increased.) At all flows tested except the highest of 21,600, the proposed algorithm appears to be computationally feasible for online use for an iteration lasting 11.35 s (Table I). The maximum time overhead of 12.44 s at 21,600 vehicles/lane/hour is only slightly above the required compute time. We would like to note also, that the simulator was coded in Python, which is generally not optimized for computational speed, and ran on an AMD A10-5750M APU processor with 8GB of RAM. With more sophisticated hardware platforms, as would be the case with next-generation infrastructure, and optimized code, we believe the proposed algorithm will be suitable for online use at very high flow rates.

3) Sensitivity Analysis on α and $k_{threshold}$: We now examine how user-defined parameters affect the exit failure rate. First, we consider α (Subsection V-A), which dictates when a vehicle should request a lane change, be it up or down. Our sensitivity analysis on α (Figure 6) showed that the optimal value for was set at $\alpha = 1.5$, but that values between $\alpha = 1.25 \cdots 2.00$ were acceptable, implying that vehicles can indeed efficiently choose how much risk to take without significantly impacting the overall flow. The second parameter we performed a sensitivity analysis on, $k_{threshold}$, also showed good convexity and the optimal value was around $90m$. In practice, such sensitivity analyses can be performed prior to system deployment for expected flow rates to achieve a 100% exit success rate.

D. Discussions

It is important to not understate the substantial improvement our proposed system achieves in the context of today's traffic flow. As long as entering traffic is regulated in some manner so that highway traffic does not reach the flow of 14,400 vehicles/hour/lane, which is several times today's maximum flow, traffic jam in the traditional sense can be avoided with the exception of accidents or equipment malfunctions while

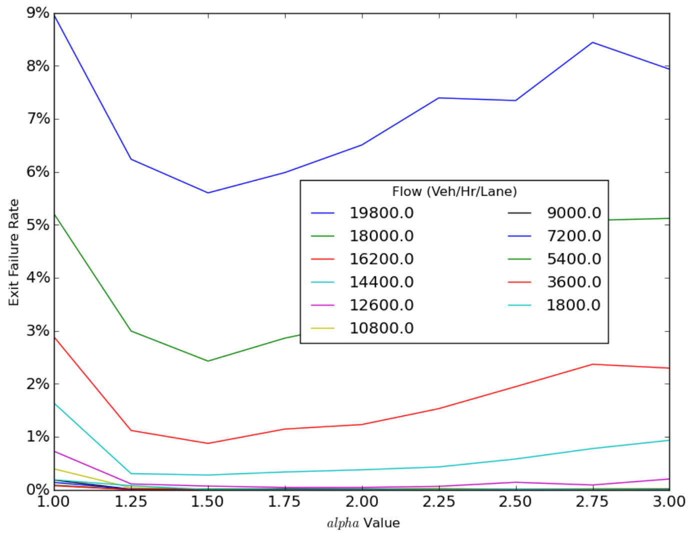


Fig. 6: Sensitivity Analysis of the α-parameter.

maintaining perfect ESR and guaranteeing bounded travel times.

In this work and when coordinating the behavior of automated vehicles in general, a trade-off must be made between higher flow and higher exit success rate, though simulation results show that automation is able to significantly improve on both metrics when compared to the nation's current transportation system. Since coordination and control allow for more predictable driving, e.g., the ability to guarantee an exit can be made at a very high probability at certain flow, the proposed algorithm can potentially help to simplify route planning techniques in both commercial and personal settings.

While it is difficult to quantitatively assess the influence of the proposed algorithm in terms of fuel consumption, especially in the context of today's transportation system, we conjecture that the impact is mostly positive, as fuel-intensive maneuvers such as excessive speeding up/slowing down, additional travel due to missed exits, as well as stop-and-go traffic are dramatically reduced if not eliminated. Existing research has already drawn such a conclusion for automated vehicles under steady-state conditions. The actual fuel savings for dynamic operations remain to be quantified.

IX. Summary & Future Work

This paper presented an approach for safely organizing automated vehicles in a dynamic highway environment to maximize the exit success rate with minimum impact on mainline traffic flow. The proposed method is iteration-based and consists of three main steps: finding eligible vehicles to make lane changes towards an exit, determining destination positions, and selecting vehicles for the actual lane changes. To increase flow, a mechanism was proposed to balance traffic across the lanes by providing vehicles with an incentive to move into inner lanes in order to travel at a faster speed. Since the proposed approach reduces excessive speed up/slow down maneuvers involved in lane changing, it is also fuel-efficient. Simulation results reveal that, using our algorithm, vehicles are able to always make successful exits at flow rates that are up to 3 times today's maximum flow with human drivers. The method presented in this paper can also be applied to stretches of highways where exits are in the left lanes and/or when vehicles need to make an exit to switch to different interstates, for example.

This work can be extended in several directions. First, the proposed algorithm can be leveraged for use when highway conditions are heterogeneous. Second, it can be adapted to assist during platoon splits or formations. Third, since our work provides some insights on vehicle coordination and control in highly dynamic environments, it may be used as a stepping stone towards automation in city streets. Finally, the proposed algorithm can be used in conjunction with route planning software to provide passengers with a more positive travel experience.

References

[1] D. Schrank, B. Eisele, and T. Lomax, "TTI's 2012 urban mobility report," Texas A&M Transportation Institute. The Texas A&M University System, Tech. Rep., Jun. 2012.

[2] G. Duranton and M. Turner, "The fundamental law of road congestion: evidence from US cities," *American Economic Review*, vol. 101, no. 6, pp. 2616–2652, Oct. 2011.

[3] H.-S. Tsao, R. Hall, and B. Hongola, "Capacity of automated highway systems: effect of platooning and barrier," California Partners for Advanced Transit and Highways (PATH), Tech. Rep., 1994.

[4] A. Bohm and M. Jonsson, "Supporting real-time data traffic in safety-critical vehicle-to-infrastructure communication," in *Proceedings of the Conference on Local Computer Networks*, Oct. 2008, pp. 614–621.

[5] J. Santa, A. Gómez-Skarmeta, and M. Sánchez-Artigas, "Architecture and evaluation of a unified V2V and V2I communication system based on cellular networks," *Computer Communications*, vol. 31, no. 12, pp. 2850–2861, Jul. 2008.

[6] D. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, Jul. 2015.

[7] R. Subramanian, "Motor vehicle traffic crashes as a leading cause of death in the U.S., 2002–a demographic perspective," U.S. Department of Transportation, National Highway Traffic Safety Administration, Tech. Rep., Jun. 2005.

[8] A. Brown, J. Gonder, and B. Repac, "An analysis of possible energy impacts of automated vehicle," in *Road Vehicle Automation*, ser. Lecture Notes in Mobility, G. Meyer and S. Beiker, Eds., Jun. 2014, pp. 137–153.

[9] R. Horowitz, C.-W. Tan, and X. Sun, "An efficient lane change maneuver for platoons of vehicles in an automated highway system," California Partners for Advanced Transit and Highways (PATH), Tech. Rep., 2004.

[10] R. Hall and C. Chinan, "Vehicle sorting for platoon formation: Impacts on highway entry and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 405–420, Oct. 2005.

[11] W.-L. Jin, "A kinematic wave theory of lane-changing traffic flow," *Transportation research part B: methodological*, vol. 44, no. 8, pp. 1001–1021, 2010.

[12] J. A. Laval and C. F. Daganzo, "Lane-changing in traffic streams," *Transportation Research Part B: Methodological*, vol. 40, no. 3, pp. 251–264, 2006.

[13] S. Moridpour, G. Rose, and M. Sarvi, “Effect of surrounding traffic characteristics on lane changing behavior,” *Journal of Transportation Engineering*, vol. 136, no. 11, pp. 973–985, 2010.

[14] C. Wang and B. Coifman, “The effect of lane-change maneuvers on a simplified car-following theory,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 523–535, 2008.

[15] S. P. Hoogendoorn and P. H. Bovy, “State-of-the-art of vehicular traffic flow modelling,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, no. 4, pp. 283–303, 2001.

[16] W. Knospe, L. Santen, A. Schadschneider, and M. Schreckenberg, “Towards a realistic microscopic description of highway traffic,” *Journal of Physics A: Mathematical and general*, vol. 33, no. 48, p. L477, 2000.

[17] B. Greenshields, “A study of traffic capacity,” in *Proceedings of the Annual Meeting Highway Research Board*, Dec. 1935, pp. 448–477.

[18] M. Broucke and P. Varaiya, “A theory of traffic flow in automated highway systems,” *Transportation research Part C: emerging technologies*, vol. 4, no. 4, pp. 181–210, 1996.

[19] C.-C. Chien, Y. Zhang, and P. A. Ioannou, “Traffic density control for automated highway systems*,” *Automatica*, vol. 33, no. 7, pp. 1273–1285, 1997.

[20] W. Chee and M. Tomizuka, “Vehicle lane change maneuver in automated highway systems,” California Partners for Advanced Transit and Highways (PATH), Tech. Rep., 1994.

[21] C. Hatipoglu, U. Ozüner, and K. Redmill, “Automated lane change controller design,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 1, pp. 13–22, Mar. 2003.

[22] R. Rajamani, H.-S. Tan, B. Law, and W.-B. Zhang, “Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, Jul. 2000.

[23] D. Moriarty and P. Langley, “Distributed learning of lane-selection strategies for traffic management,” Daimler-Benz Research and Technology Center, Tech. Rep., 1998.

[24] J. Medanic, D. Ramaswamy, W. R. Perkins, and R. Benekohal, “Partitioned lane assignment strategies for balancing excess lane capacity on AHS,” in *Proceedings of the American Control Conference*, vol. 5, Jun. 1995, pp. 3581–3585 vol.5.

[25] D. Ramaswamy, J. Medanic, W. Perkins, and R. Benekohal, “Lane assignment on automated highway systems,” *IEEE Transactions on Vehicular Technology*, vol. 46, no. 3, pp. 755–769, Aug. 1997.

[26] R. W. Hall and D. Lotspeich, “Optimized lane assignment on an automated highway,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 4, pp. 211–229, 1996.

[27] R. W. Hall and C. Caliskan, “Design and evaluation of an automated highway system with optimized lane assignment,” *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 1, pp. 1–15, 1999.

[28] T.-S. Dao, C. M. Clark, and J. P. Huissoon, “Optimized lane assignment using inter-vehicle communication,” in *Proceedings of the Intelligent Vehicles Symposium*, Jun. 2007, pp. 1217–1222.

[29] K. Kim, D.-I. Cho, and J. Medanic, “Lane assignment using a genetic algorithm in the automated highway systems,” in *Proceedings of the International Conference on Intelligent Transportation System*, Sep. 2005, pp. 540–545.

[30] D. Desiraju, T. Chantem, and K. Heaslip, “Minimizing the disruption of traffic flow of automated vehicles during lane changes,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1249–1258, Jun. 2015.

[31] S. Tsugawa, “Automated driving systems: Common ground of automobiles and robots,” *International Journal of Humanoid Robotics*, vol. 8, no. 1, pp. 1–12, 2011.

[32] H.-H. Hsu and A. Liu, “Kinematic design for platoon-lane-change maneuvers,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 185–190, Jan. 2008.

[33] ——, “Platoon lane change maneuvers for automated highway systems,” in *Proceedings of the Conference on Robotics, Automation and Mechatronics*, Dec. 2004, pp. 780–785.

[34] R. W. Hall, “Longitudinal and lateral throughput on an idealized highway,” *Transportation science*, vol. 29, no. 2, pp. 118–127, 1995.

[35] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill, “On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach,” in *Proceedings of the International Conference on Intelligent Transportation System*, Sep. 2012, pp. 900–906.

[36] I. Berry, “The effects of driving style and vehicle performance on the real-world fuel consumption of US light-duty vehicles,” Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

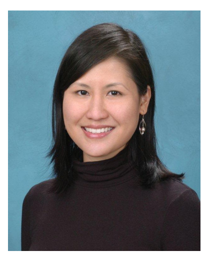
[37] T. V. da Rocha, L. Leclercq, M. Montanino, C. Parzani, V. Punzo, B. Ciuffo, and D. Villegas, “Does traffic-related calibration of car-following models provide accurate estimations of vehicle emissions?” *Transportation research part D: Transport and Environment*, vol. 34, pp. 267–280, 2015.

[38] K. Ahn, H. Rakha, A. Trani, and M. Van Aerde, “Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels,” *Journal of transportation engineering*, vol. 128, no. 2, pp. 182–190, 2002.

[39] D. Marinescu, J. Čurn, M. Slot, M. Bouroche, and V. Cahill, “An active approach to guaranteed arrival times based on traffic shaping,” in *Proceedings of the International Conference on Intelligent Transportation System*, Sep. 2010, pp. 1711–1717.

[40] T. R. Board, *Highway Capacity Manual 2010*. Transportation Research Board, 2010.

Eric Meissner is an undergraduate at Utah State University. He will graduate with a Bachelors degree in Computer Science in 2016, and plans to pursue a PhD in Computer Science. His current research interests include intelligent transportation systems and distributed computation.



Thidapat Chantem (S’05-M’11) received her Ph.D. and Master’s degrees from the University of Notre Dame in 2011 and her Bachelor’s degrees from Iowa State University in 2005. She is an Assistant Professor of Electrical and Computer Engineering at Utah State University. Her research interests include Intelligent Transportation Systems, Cyber-Physical System Design, Hardware-Software Co-design of Embedded Real-Time Systems, and Energy- Aware and Thermal-Aware System-Level Design.

Kevin Heaslip received the Ph.D. degree in civil and environmental engineering from University of Massachusetts, Amherst, MA, USA, in 2007 and the B.S. and M.S. degrees from Virginia Tech, Blacksburg, VA, USA, in 2002 and 2003, respectively. He is an Associate Professor of transportation engineering with the Department of Civil and Environmental Engineering at Virginia Tech. His research interests include intelligent transportation systems and transportation systems operations.