Autonomous Intersection Management for Semi-Autonomous Vehicles

Tsz-Chiu Au
School of Electrical and Computer Engineering
Ulsan National Institute of Science and Technology
South Korea
chiu@unist.ac.kr

Shun Zhang
Department of Computer Science
The University of Texas at Austin
U.S.A.
jensen.zhang@utexas.edu

Peter Stone
Department of Computer Science
The University of Texas at Austin
U.S.A.
pstone@cs.utexas.edu

Abstract— Recent advances in autonomous vehicle technology will open the door to highly efficient transportation systems in the future, as demonstrated by Autonomous Intersection Management, an intersection control protocol designed for fully autonomous vehicles. We, however, anticipate there will be a long transition period during which most vehicles have some but not all capabilities of fully autonomous vehicles. This paper introduces a new protocol called *Semi-Autonomous Intersection Management*, which allows vehicles with partially-autonomous features such as adaptive cruise control to enter an intersection from different directions simultaneously. Our experiments show that this protocol can greatly decrease traffic delay when most vehicles are semi-autonomous. Our incremental deployment study reveals that traffic delay keeps decreasing as more and more vehicles employ features of autonomy.

1. Introduction

Recent robotic car competitions and demonstrations have convincingly shown that autonomous vehicles are feasible with the current generation of hardware [1]. Looking ahead to the time when autonomous cars will be common, Dresner and Stone proposed a new intersection control protocol called *Autonomous Intersection Management* (AIM) and showed that by leveraging the control and network capabilities of autonomous vehicles it is possible to design an

intersection control protocol that is much more efficient than traffic signals [2]. By removing human factors from control loops, autonomous vehicles, with the help of advanced sensing devices, can be safer and more reliable than human drivers. The AIM protocol exploits the fine control of autonomous vehicles to allow more vehicles simultaneously to cross an intersection, thus effectively reducing the delay of vehicles by orders of magnitude compared to traffic signals [3].

AIM is designed for the time when all vehicles are fully autonomous. We, however, anticipate that there will be a long transition period during which most vehicles have some but not all capabilities of fully autonomous vehicles. In fact, this transition period has already begun. Since the late 1990s, adaptive cruise control systems and lane departure warning systems have become widely available as optional equipment on luxury production vehicles. The National Highway Traffic Safety Administration acknowledges that fully autonomous vehicles represent just the top level in five levels of vehicle automation [4]. Indeed, they define a level below this top level with vehicles that have limited self-driving automation. The main motivation of this paper is to propose a new intersection control system called *Semi-Autonomous Intersection Management (SemiAIM)* that can accommodate both fully autonomous vehicles and *semi-autonomous* vehicles with limited self-driving automation. There is a high likelihood that human-driven vehicles, semi-autonomous vehicles, and fully autonomous vehicles will *coexist* on the road in the future. SemiAIM takes advantages of this trend and allows autonomous intersections to handle a traffic mixture with different types of vehicles.

The main contributions of this paper are 1) the introduction of the concept of SemiAIM; 2) a specification of constraint-based reservation requests for semi-autonomous vehicles; and 3) detailed empirical results demonstrating the effectiveness of this protocol, especially in moderate traffic levels with a mix of human-driven, semi-autonomous, and fully autonomous vehicles. The remainder of the paper is organized as follows. Section II outlines the architecture of AIM which forms the basis of SemiAIM. Section III and Section IV categorize semi-autonomous vehicles and discuss the interaction between human drivers and semi-autonomous vehicles in SemiAIM. Section V describes the constraint-based reservation system in SemiAIM. Section VI presents the results of the simulation experiments we used to evaluate SemiAIM. The related work and the conclusion are given in Section VII and VIII, respectively.

2. Autonomous Intersection Management

As laid out in detail in [2], an ideal autonomous intersection management protocol must satisfy seven desiderata: 1) allow for fully distributed and autonomous control by the vehicles, 2) have low communication complexity, 3) assume non-expensive vehicle sensors found in production, 4) use a standardized communication protocol, 5) be incrementally deployable, 6) be safe, and 7) be efficient. Modern-day traffic signals and stop signs completely satisfy all but the last one of these properties. Traffic signals and stop signs are very inefficient—not only do

vehicles traversing intersections equipped with these mechanisms experience large delays, but the intersections themselves can only manage a limited traffic capacity—much less than that of the roads that feed into them.

2.1. The Reservation Idea

With these seven desiderata in mind, we have introduced a novel approach to efficient intersection management that is a radical departure from existing traffic signal optimization schemes. The solution is based on a reservation paradigm, in which vehicles "call ahead" to reserve space-time in the intersection [2]. In the approach, we assume that computer programs called driver agents control the vehicles, while an arbiter agent called an intersection manager is placed at each intersection. The driver agents attempt to reserve a block of space-time in the intersection. The intersection manager decides whether to grant or reject requested reservations according to an intersection control policy. In brief, the paradigm proceeds as follows.

- An approaching vehicle announces its impending arrival to the intersection manager. The vehicle indicates its size, predicted arrival time, velocity, acceleration, and arrival and departure lanes.
- The intersection manager simulates the vehicle's path through the intersection, checking for conflicts with the paths of any previously processed vehicles.
- If there are no conflicts, the intersection manager issues a reservation. It becomes the vehicle's responsibility to arrive at, and travel through, the intersection as specified (within a range of error tolerance).
- In the case of a conflict, the intersection manager suggests an alternate later reservation.
- The car may only enter the intersection once it has successfully obtained a reservation.
- Upon leaving the intersection, the car informs the intersection manager that its passage through the intersection was successful.

The best way to get a sense of the paradigm is to watch it in action. [1] Note that this video includes only autonomous cars. We have also extended the system to work when some of the cars are driven by people [2]. As one would expect, the improvements over current systems are smaller; but they are still significant.

Figure 1 diagrams the interaction between driver agents and an intersection manager. A key feature of this paradigm is that it relies only on vehicle-to-infrastructure (V2I) communication. In particular, the vehicles need not know anything about each other beyond what is needed for local autonomous control (e.g., to avoid running into the car in front). The paradigm is also completely robust to communication disruptions: if a message is dropped, either by the intersection manager or by the vehicle, delays may increase, but safety is not compromised. Safety can also be guaranteed in mixed mode scenarios when both autonomous and manual vehicles operate at

[1] Available online at http://www.cs.utexas.edu/~aim/video/fcfs-insanity.mov

intersections. The intersection efficiency will increase with the ratio of autonomous vehicles to manual vehicles in such scenarios.

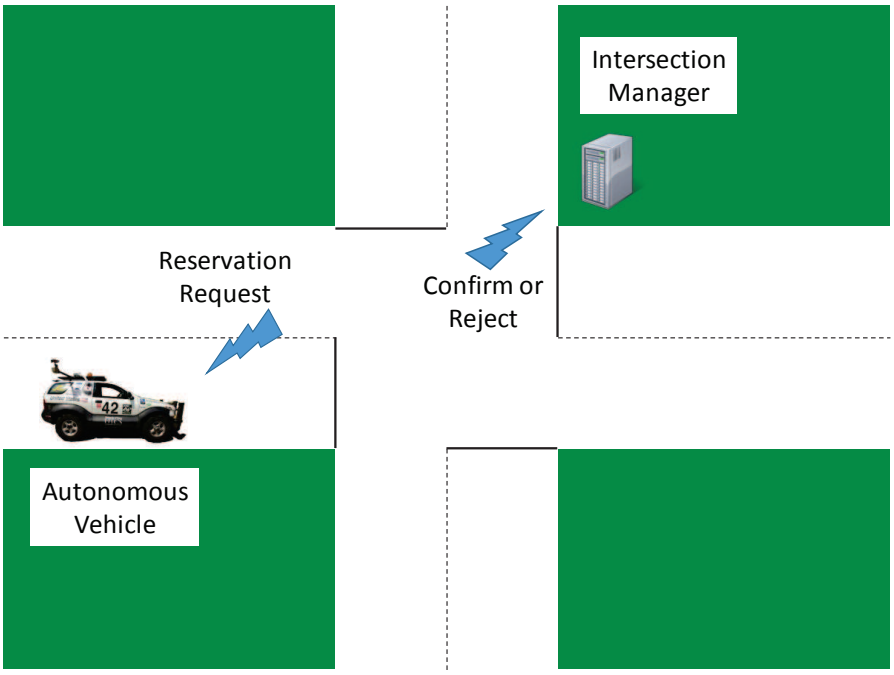
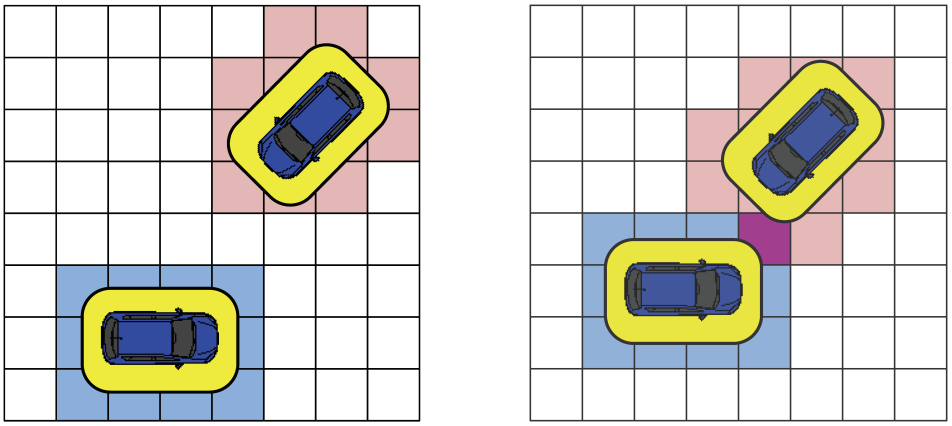


Figure 1. Diagram of the intersection system.

2.2. Intersection Control Policy

Our prototype intersection control policy divides the intersection into a grid of reservation tiles, as shown in Figure 2. (This notation can be generalized for rectangular and irregularly-shaped intersections.) When a vehicle approaches the intersection, the intersection manager uses the data in the reservation request regarding the time and velocity of arrival, vehicle size, etc. to simulate the intended journey across the intersection. At each simulated time step, the policy determines which reservation tiles will be occupied by the vehicle.

If at any time during the trajectory simulation the requesting vehicle occupies a reservation tile that is already reserved by another vehicle, or could be occupied by a human-driven vehicle, the policy rejects the driver's reservation request, and the intersection manager communicates this to the driver agent. Otherwise, the policy accepts the reservation and reserves the appropriate tiles. The intersection manager then sends a confirmation to the driver. If the reservation is denied, it is the vehicle's responsibility to maintain a speed such that it can stop before the intersection. Meanwhile, it can request a different reservation.



(a) Successful (b) Rejected

Figure 2. (a) The vehicle's space-time request has no conflicts at time *t*. (b) The black vehicle's request is rejected because at time *t* of its simulated trajectory, the vehicle requires a tile already reserved by another vehicle.

2.3. Simulation Results

Empirical results in simulation demonstrate that our proposed reservation system can dramatically improve the intersection efficiency when compared to traditional intersection control mechanisms. To quantify efficiency, we introduce delay, defined as the amount of travel time incurred by the vehicle as the result of passing through the intersection.

In experiments, our fully-implemented custom micro-simulator simulates 3 lanes in each of the 4 cardinal directions. The speed limit in all lanes is 55 miles per hour. Different management systems (reservation system, stop sign, traffic signal) are run for at least 100,000 steps, which corresponds to approximately half an hour of simulated time. A video demonstration[2] shows cars driving from all directions and barely slowing down as they traverse the intersection with no collisions.

The results for the experiments are shown in Figure 3. The reservation system (labeled FCFS) performs very well, nearly matching the performance of the optimal policy which represents a lower bound on delay should there be no other cars on the road. At higher levels of traffic, the average delay for a vehicle using the reservation system gets as high as 0.35 seconds, but it is never more than 1 second above optimal. Under none of the tested conditions does the reservation system approach the trip times of the traffic signal system—the best empirical performance over a range of tested cycle times is shown. Likewise, using stop signs increases delays over the reservation system, especially as the traffic level increases. Overall, by allowing for much finer-grained coordination, the simulation-based reservation system can dramatically reduce per-car delay by two orders of magnitude in comparison to traffic signals and stop signs.

[2] Available online at http://www.cs.utexas.edu/~aim/video/fcfs-insanity.mov

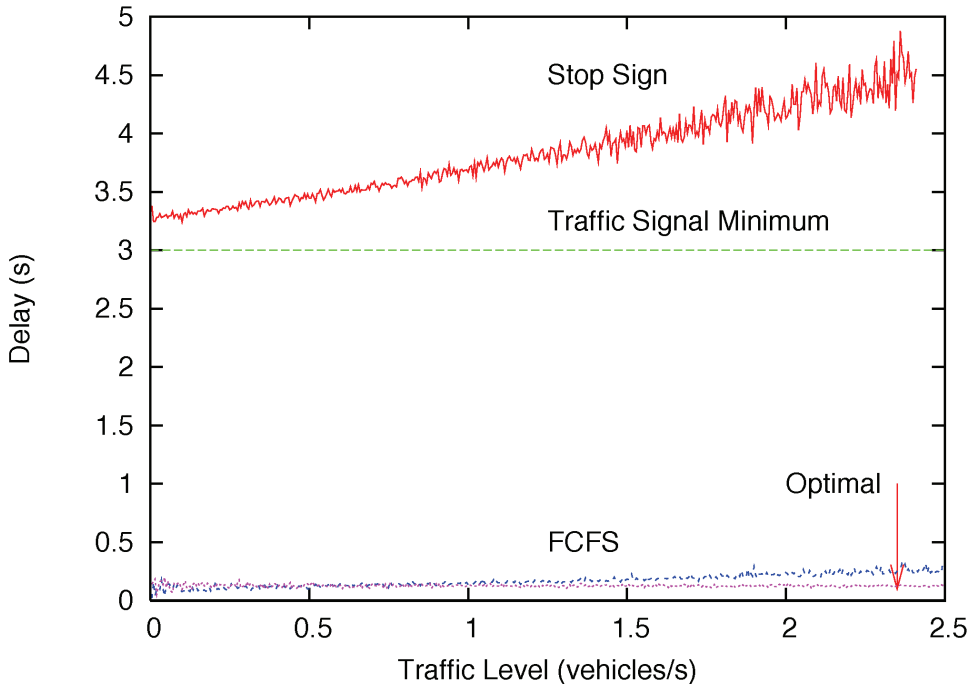


Figure 3. Traffic delay for the reservation system, stop sign, optimized traffic signal, and optimal lower bound when collisions are ignored.

2.4. Worst-Case Analysis and V2V interactions

Additionally, we have studied the impacts on travel times for emergency vehicles; the potential benefits across networks of intersections; and the worst-case effects of vehicle mechanical failure in intersections, showing that despite the apparent potential for "pile-ups" when there is an accident, our protocol prevents such disastrous situations even under conservative assumptions [20]. We have also shown that the benefits of our system can be replicated at low-traffic intersections without any infrastructure in place via vehicle-to-vehicle (V2V) communication [11].

3. Semi-Autonomous Vehicles

A drawback of FCFS is that it does not work with any vehicles that require human operations. Dresner and Stone proposed FCFS-signal (a.k.a. FCFS-Light) to make AIM compatible with human drivers [5]. According to their study, however, when more than 5-10% of traffic is comprised of human-driven vehicles, the average delay time of all vehicles increases significantly under FCFS-signal. The reason for this phenomenon is twofold. First, human-driven vehicles cannot enter the intersection during red signal phases, because they are *not* allowed to

make any reservation. Second, as a consequence of the first, autonomous vehicles often fail to obtain reservations when they are behind the human-driven vehicles stalled at red signals. These issues stem from the fact that FCFS-signal prohibits human-driven vehicles from entering intersections during red signal phases. SemiAIM sets out to overcome these issues by allowing *semi-autonomous vehicles*—vehicles with limited autonomous driving and wireless communication capabilities—to use the AIM reservation system to enter an intersection during red signal phases. While these vehicles are not fully autonomous, they are able to follow a *limited* number of predictable trajectories at intersections more precisely than human drivers. This ability allows them to utilize our constraint-based reservation system to make reservations in the same manner as fully autonomous vehicles and enter an intersection at red signals.

Our proposed reservation system is general enough to accept reservation requests from *any* semi-autonomous vehicles that are capable of following some trajectories and communicating with the IM. To facilitate our discussion, we will focus on semi-autonomous vehicles that use the following set of equipment that is readily available today.

Table 1. Features of Semi-Autonomous Vehicles.

Vehicle Type	Communication Device	Cruise Control	Adaptive Cruise Control
SA-ACC	X	X	X
SA-CC	X	X	
SA-Com	X		

- Communication device (Com): a component in a vehicle's on-board electronic system that enables the vehicle to wirelessly communicate with the transportation infrastructure including the IM. The communication is bidirectional.
- Simple Cruise control (CC): An optional speed control subsystem in vehicles' drivetrain that automatically controls the vehicle speed by taking over the throttle of the vehicles.
- Adaptive cruise control (ACC): an advanced cruise control system that automatically adjusts the speed of a vehicle in order to maintain a certain distance from vehicles ahead.

All of this equipment give semi-autonomous vehicles *some* of the functionality of autonomous vehicles, though human drivers still retain some control of the vehicles. We can equip a semi-autonomous vehicle with more than one of these devices. In this paper we introduce three types of semi-autonomous vehicles that we envision utilizing this equipment: *Type SA-ACC Vehicles*, *Type SA-CC Vehicles* and *Type SA-Com Vehicles*. Table 1 summarizes the equipment used on these different types of semi-autonomous vehicles. Figure 4 shows a screenshot of the SemiAIM simulator we developed to simulate semi-autonomous vehicles at intersections. In this figure, the yellow vehicles are fully autonomous, the green vehicles are Type SA-ACC, the blue vehicles are Type SA-CC, the white vehicles are Type SA-Com, and the magenta vehicles are human-driven vehicles.

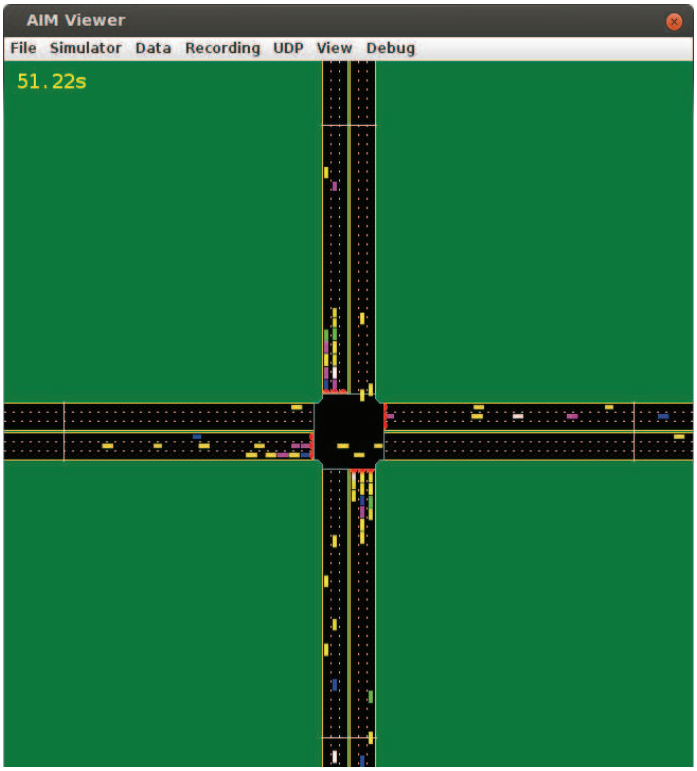


Figure 4. A screenshot of the simulator we developed for the experiments on SemiAIM. This simulator is modified from the open-source AIM4 simulator.

4. Interaction Model

Safety is a main concern when involving human drivers in the control loop of semi-autonomous vehicles. For the semi-autonomous vehicles defined above to be able to go through an intersection safely, we need to define a simple and clean interface for negotiating with the IM and passing control between the human driver and the driver agent. In this section, we describe how our proposed protocol can be realized safely by having an *interaction model* between human drivers and driver agents that simplifies the task of the human drivers.

Figure 5 summarizes the interaction model between human drivers, driver agents, and the IM. We require the inclusion in the vehicle of a single button that signals the driver agent to ask for a reservation. After pressing the button, the driver agent will automatically send a request message to the IM on behalf of the human driver. It is also important that there is a clear “Okay” indicator (such as a green light) installed in the car which indicates when the request has been confirmed. After seeing the okay signal, the driver would have to actively pass control to the driver agent, again by pressing a single button. This way the driver will not be surprised by any sudden autonomous actions of the vehicle.

This interaction model only requires the human driver to perform relatively simple driving maneuvers such as holding the steering wheel at a certain angle (Types SA-ACC and SA-CC vehicles) or driving as if under a traffic signal (Type SA-Com vehicles). These tasks are much simpler than other maneuvers such as lane changing and passing other vehicles, and thus should not be taxing to experienced human drivers.

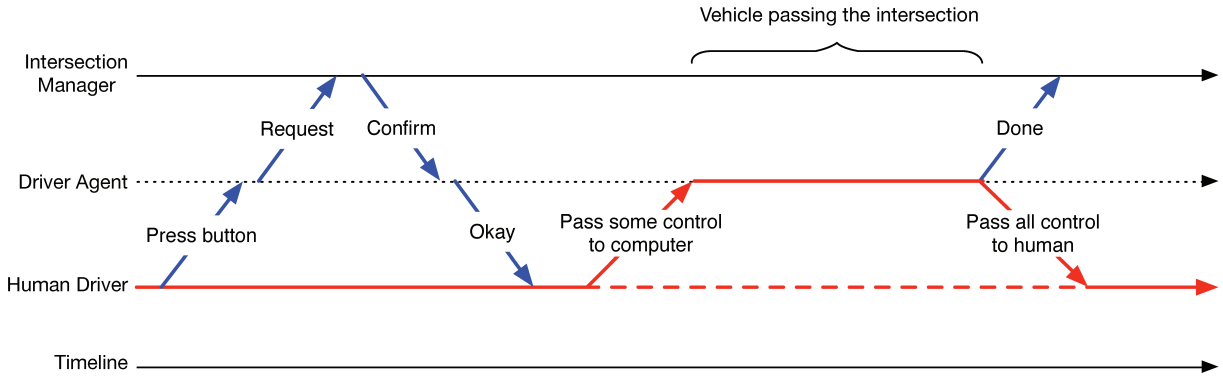


Figure 5. The interaction between human drivers, driver agents, and the IM. The blue lines are message passing, and the red lines are transfer of control. Note that human drivers retain some control of the vehicle inside the intersection (the dashed red line).

5. Constraint-based Reservation Systems

SemiAIM allows human-driven vehicles and semi-autonomous vehicles to make reservations in the same way as fully autonomous vehicles. The key idea of SemiAIM is to turn AIM into a *constraint-based reservation system*, which allows vehicles to make reservations in terms of constraints over 1) their driving profiles such as their arrival time and arrival velocity, and 2) the relationships with other vehicles.

In AIM, a reservation request is a 5-tuple $\langle l_1, l_2, t_0, v_0, p \rangle$, where l_1 is the entry lane, l_2 is the exit lane, t_0 is the arrival time, v_0 is the arrival velocity, and p is the physical characteristics of the vehicle. This information allows the IM to compute the exact trajectory of the vehicle and reserve tiles on the trajectory. However, this computation assumes the vehicle can be controlled *precisely* in the intersection so that it can meet the reservation constraints exactly. Human drivers cannot control their vehicles as precisely, and semi-autonomous vehicles may only be able to control certain aspects of their trajectories. Therefore, we need a new kind of reservation requests that do not rely on this assumption.

5.1. Maneuverability Profiles

Semi-autonomous vehicles only have limited self-driving automation. For example, adaptive cruise control can only provide autonomous longitudinal control, but still depends on the human driver to use the steering wheel to maintain the lateral control. Hence the longitudinal control of Type SA-ACC vehicles can be very precise, but the lateral control cannot be as precise. Furthermore, the human driver cannot make a sharp turn while the adaptive cruise control is active, because the human must slow down the vehicle a lot in order to be able to make a sharp turn precisely. In our system, we use a *maneuverability profile* to encode the *constraints* over the control of human drivers when automation devices are activated. For simple cruise control, the maneuverability profile is written in Lisp syntax as follows:

```
(cc-profile (v verror angle)
  (is-auto-speed-control)
  (not is-auto-steering)
  (< velocity (+ v verror))
  (> velocity (- v verror))
  (< steer-angle angle) (> steer-angle -angle))
```

where v is the target velocity, v_{error} is the maximum error of the target velocity, and angle is the maximum feasible steering angle for the human driver when the cruise control is turned on. The maneuverability profile for adaptive cruise control is

```
(acc-profile (vin d derror angle)
  (is-auto-speed-control)
  (not is-auto-steering)
  (< (dist-from vin) (+ d derror))
  (> (dist-from vin) (- d derror))
  (< steer-angle angle) (> steer-angle -angle))
```

where vin is the vehicle id of the vehicle ahead, d is the target distance from the vehicle ahead, d_{error} is the maximum error for acc to maintain the target distance from vin, and angle is the maximum feasible steering angle for the human driver.

Other types of semi-autonomous vehicles have their own maneuverability profiles. When a driver agent makes a reservation request, it is his or her responsibility to ensure that the trajectories encoded by the reservation request *satisfy* all constraints in the maneuverability profile.

5.2. Constraint-Based Reservation Requests

Different maneuverability profiles can have different sets of constraints. To facilitate communication with all kinds of semi-autonomous vehicles, SemiAIM uses a *unified* language

for vehicles to express their constraints in the same format in their requests. We define *constraint-based* reservation requests as follows. A request message consists of four components:

- An intention statement states the direction in which the vehicle wants to exit from an intersection. An intention statement is formally a disjunction of lane and road identifiers: $(l_1 \vee l_2 \vee \cdots \vee r_1 \vee r_2)$, where l_i is an exit lane and r_i is an exit road. For every lane l_i, there exists only one r_j such that $l_i \in r_j$. Examples of legal intentions are $(l_1 \vee l_3)$ or (r_1). This feasibility facilitates different path planning strategies the vehicle might use.
- A vehicle type indicates the type of the vehicle. The IM maintains a database of all car types in existence, with information such as sizes, shapes, and kinematics of different car types. This information allows the IM to determine how the vehicle will move inside an intersection.
- An entry statement describes how a vehicle enters an intersection. An entry statement is a 3-tuple $\langle(l_1 \vee l_2 \vee \cdots \vee l_n), [t_1, t_1], [v_1, v_1]\rangle$, where l_i is a possible lane from which the vehicle enters the intersection, $[t_1, t_1]$ is the time interval during which the vehicle will arrive at the intersection, and $[v_1, v_1]$ is the interval of the arrival velocity of the vehicle.
- An acceleration profile is a schedule of acceleration a vehicle can use to accelerate through the intersection on a trajectory. An acceleration profile is a list of pairs $\langle(t_1, a_1), (t_2, a_2), \cdots, (t_n, a_n)\rangle$, where a_i is the acceleration the vehicle will use from time t_i to time t_{t+1}. Note that the vehicle may or may not provide the list of acceleration profiles of all possible trajectories in a request message. If the list of acceleration profiles is missing, the IM will generate an acceleration profile based on a simulation of the vehicle movement given the vehicle type, the entry statement, and the intention statement.

A vehicle must generate constraint-based requests that satisfy its maneuverability profile only. In this paper, we do not fully explore the space of how vehicles could generate such requests. However, we define the following two requests that will be used by Type SA-CC and Type SA-Com vehicles:

- A constant-velocity request is $\langle Intent, Type, Entry, AP\rangle$ where $Intent = (l_1 \vee l_2 \vee \cdots \vee l_n)$ in which l_i is a possible lane from which the vehicle exits the intersection; *Type* is the vehicle type; $Entry = \langle(l'_1 \vee l'_2 \vee \cdots \vee l'_n), [t_1, t_1], [v_1, v_1]\rangle$ is the entry statement; and $AP = (\langle(t_1, 0)\rangle)$ is the acceleration profile list. Since the acceleration in *AP* is always zero, the vehicle will move at a constant velocity.
- A whole-row request is $\langle Intent, Type, Entry, AP\rangle$, where $Intent = (l_1 \vee l_2 \vee \cdots \vee l_n)$ in which l_i is a possible lane from which the vehicle exits the intersection; *Type* is the vehicle type; $Entry = \langle(l'_1 \vee l'_2 \vee \cdots \vee l'_n), [t_1, t_1], [v_1, v_1]\rangle$ is the entry statement; and *AP* is the acceleration profile list. In order to reserve the entire row in an intersection, the difference between t_1 and t_2 as well as the difference between v_1 and v_2 must be large enough to cover the entire row, regardless of the acceleration profiles.

Unlike AIM's reservation requests which always correspond to a particular trajectory, a constraint-based request is an *incomplete* description of the trajectory. The IM interprets a

constraint-based request as a description of a set of *possible* trajectories that the vehicle may follow, and reserves all the tiles on the possible trajectories. Once the IM receives a request, it conducts an internal simulation of the vehicle movement, as described in [2], to determine the set of tiles on the possible trajectories and their availability.

Our hypothesis is that more restrictive constraint-based requests will perform better because SemiAIM reserves fewer tiles for them, meaning that some types of semi-autonomous vehicles will work better with SemiAIM. We will test this hypothesis in Section 4.1.

5.3. Anchor Requests

Semi-autonomous vehicles with adaptive cruise control can use a special constraint-based request called *anchor requests* to make reservations. An anchor request is $\langle Type, vin, d \rangle$, where *Type* is the vehicle type, *vin* is the vehicle id of the vehicle ahead, and *d* is the following distance from the vehicle of vin. Currently anchor requests are designed specifically for semi-autonomous vehicles with adaptive cruise control. However, this formulation is general enough to enable some sophisticated cooperative maneuvers such as platooning [6].

6. EXPERIMENTAL EVALUATION

To demonstrate the feasibility of SemiAIM as well as evaluate the hypothesis that SemiAIM can offer substantial improvements over traffic signals and FCFS-Signal, we conducted several experiments using a modified AIM4 simulator that simulates the behavior of vehicles in our constraint-based reservation system. The implementations of fully autonomous vehicles (Type A) and human-driven vehicles (Type H), as described in [2], are provided in the simulator. The concrete implementations of the semi-autonomous vehicles in Section 3 are as follows.

- Type SA-Com vehicles: Send a whole-row request to the IM. If the entire lane is available, the request is confirmed; otherwise, the vehicle resends the request after a few seconds and decelerates enough to be able to stop before the intersection. If it keeps failing to obtain a reservation it stops at the intersection and returns control to the human driver, who must then obey the traffic signal.
- Type SA-CC vehicles: Send a constant-velocity request to the IM when it is close enough to the intersection. If it can enter the intersections by keeping the current velocity, the request is confirmed; otherwise, the vehicle follows rules for Type SA-Com vehicles.
- Type SA-ACC vehicles: If there exists a vehicle ahead which is either autonomous or semi-autonomous and is going in the same direction, it sends an anchor request to the IM. If the request is confirmed, the vehicle follows the vehicle ahead and enters the intersection. If the request is denied or there is no such vehicle ahead, the vehicle follows rules for Type SA-CC vehicles.

Our experiments were conducted in a 3x3 intersection (3 lanes in each direction) as shown in Figure 4. Unless otherwise specified, the simulator spawns vehicles in each lane according to a Poisson distribution with an expectation of 360 vehicles per hour (i.e., traffic level = 360 vehicles/hour/lane). We chose this traffic level as being heavy enough to cause significant delays at signals, but light enough to allow for benefits even if cars are not precisely controlled.

The performance of an intersection is measured by the *average delay*, where delay is computed as the increase in travel time compared to traversing the intersection without slowing down at all, as if no other vehicles were on the road. Thus, lower delays correspond to better performance. This measure is the same one used in [2]. For all the vehicles, both the static buffer and the edge time buffer are 0.25 meters (see [2] for details), when auto-controlled.

Traffic signals are used as a fallback strategy when vehicles cannot get a reservation. We used an *optimized* 6-phase signal schedule that includes relatively long green times (roughly 35 seconds) for East-West and North-South phases, and shorter phases for each of the four single direction phases (roughly 9 seconds). It was optimized using SYNCHRO, a commercial traffic optimization package.

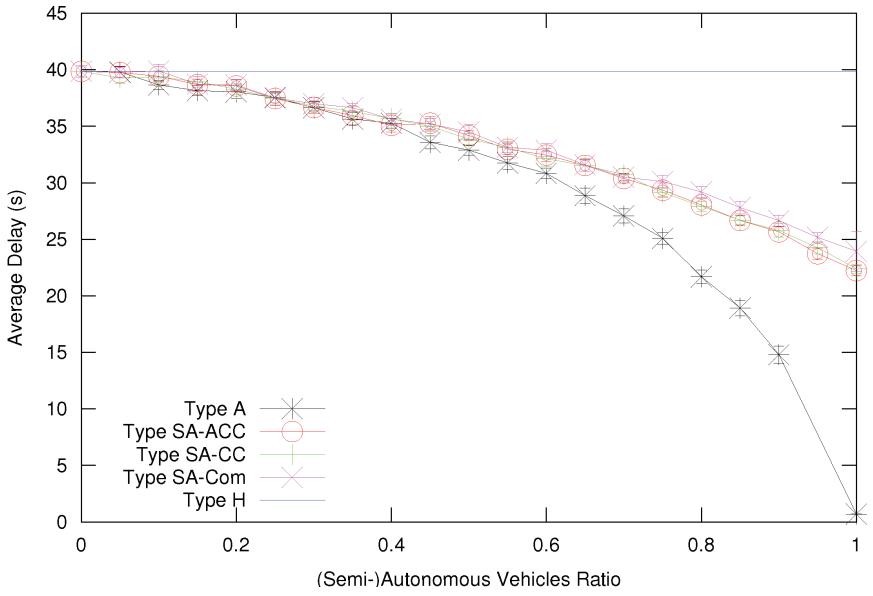


Figure 6. (Semi-)Autonomous vehicles vs. Human-Driven vehicles. Traffic level = 360 vehicles/lane/hour.

6.1. Technology Penetration

Our first experiment studies the effect of an increasing penetration rate of vehicular automation technology. Suppose human-driven vehicles are gradually replaced by a particular type of semi-autonomous vehicle or fully autonomous vehicle until all vehicles become that type. We examine how much benefit SemiAIM provides during the transition period.

In this experiment, the traffic consisted of one of the three types of vehicles we defined in Section III as well as Type A and Type H vehicles. We measured the traffic delay as we gradually increased the ratio of (semi-)autonomous vehicles to human-driven vehicles (Type H) while keeping the traffic level at 360 vehicles/hour/lane. As an example, consider the ratio of Type SA-ACC vehicles. At the beginning, there are 0% Type SA-ACC vehicles and 100% human-driven vehicles, and the ratio is 0. As the number of Type SA-ACC vehicles increases, the ratio increases and eventually becomes 1, which means there are 100% Type SA-ACC vehicles and 0% human-driven vehicles. We repeated the simulation 30 times for 1800s during each time. For each run, we measured the average delay of all vehicles. The average delays are shown in Figure 6. Each data point in the figure is an average of 30 values, and the error bar is the 95% confident interval of the average delay.

According to Figure 6, the performance of semi-autonomous vehicles is very similar to fully autonomous vehicles when the ratio to human-driven vehicles is below 40%. However, when the ratio increases beyond 40%, fully autonomous vehicles increasingly outperform semi-autonomous vehicles. Previous studies showed that FCFS-Signal needs at least 90% of fully autonomous vehicles in the traffic in order to be fully effective [5]. We successfully replicate the result, observing that the average delay drops rapidly when the traffic has more than 90% fully autonomous vehicles, and approaches zero when all vehicles are fully autonomous. Semi-autonomous vehicles cannot achieve the same dramatic decrease in traffic delay, but they, with the help of SemiAIM, manage to reduce the delay by 46% (from 39.9s to 22.4s) compared to human-driven vehicles. As expected, in the presence of semi-autonomous vehicles, SemiAIM provides significant advantages even when there are no fully autonomous vehicles on the road.

Another observation is that both Type SA-ACC and Type SA-CC vehicles have a significantly lower average delay than Type SA-Com vehicles. The difference is consistent with our hypothesis that the use of more constrained requests can increase the performance of intersections since the footprints of the vehicles are smaller and more vehicles can enter the intersection at the same time. Nonetheless, the difference is small because the simulated human drivers in the simulator can control their vehicles quite well.

Table 2. Deployment Schedule in Figure 7.

Schedule	1	2	3	...	31
Type H	90%	87%	84%	...	0%
Type SA-{ACC,CC,Com}	9%	11%	13%	...	69%

Table 3. Deployment Schedule in Figure 8.

Schedule	1	2	3	...	17
Type H	10%	10%	10%	...	10%
Type SA-{ACC,CC,Com}	85%	80%	75%	...	5%
Type A	5%	10%	15%	...	85%

6.2 Incremental Deployment

Our second experiment considered realistic scenarios of *incremental deployment* of autonomous vehicle technology. We believe that the adoption of semi-autonomous vehicles will be much faster than fully autonomous vehicles since the cost of ownership of semi-autonomous vehicles are lower. Thus we foresee there will be a period of time during which semi-autonomous vehicles dominate the roads. However, this domination will be short-lived—eventually fully autonomous vehicles will displace semi-autonomous vehicles due to their greater convenience. This section studies the effects of different levels of incremental deployment of (semi-)autonomous vehicles over time.

We consider two plausible deployment schedules, one for the period of time before the domination of semi-autonomous vehicles, and the other for the subsequent period. In the first deployment schedule (Table 2), Type H vehicles on the road are gradually displaced by (semi-)autonomous vehicles. Assume two out of three car buyers who abandoned their vehicles for newer technology will buy a semi-autonomous vehicles rather than a fully autonomous vehicle. Then eventually about two thirds of vehicles on the road will be semi-autonomous, and one third will be fully autonomous. Table 3 shows the second deployment schedule under which semi-autonomous vehicles are gradually displaced by fully autonomous vehicles. We assume that even when most vehicles are fully autonomous, a small number of drivers still prefer driving themselves.

Figures 7 and 8 show the results of the simulations according to these deployment schedules. The simulation setup is exactly the same as in Experiment 1. For each vehicle distribution, we ran the simulation 30 times for 1800s each run. Thus, each data point in the figures is an average of 30 delay times. The 95% CIs of the average are shown as the error bars in the figures. According to these figures, the average delay of the vehicles keeps decreasing as (semi-)autonomous vehicles are deployed, and the choice of the semi-autonomous vehicle technology does not matter much. Ultimately, the average delay depends on the number of drivers who insist on driving themselves.

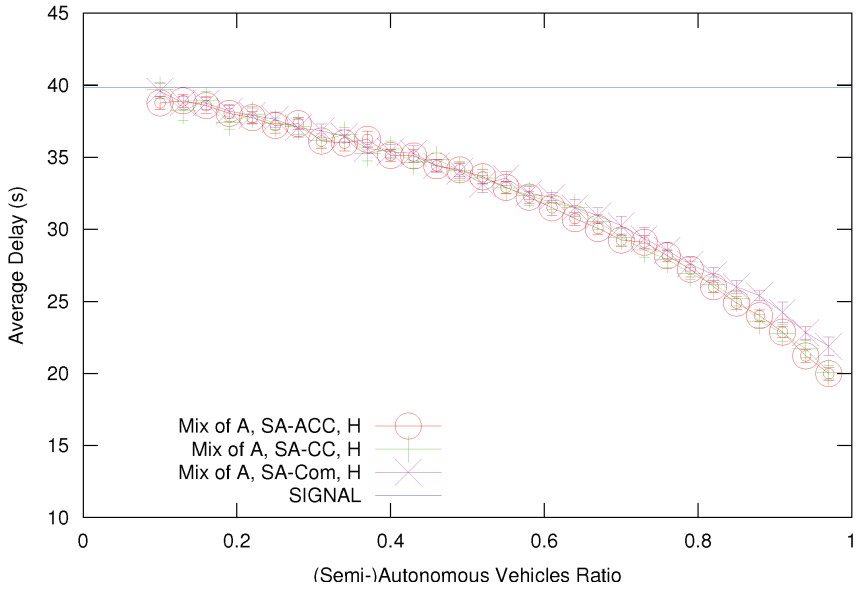


Figure 7. The average delay according to the deployment schedule in Table 2. Traffic level = 360 vehicles/lane/hour.

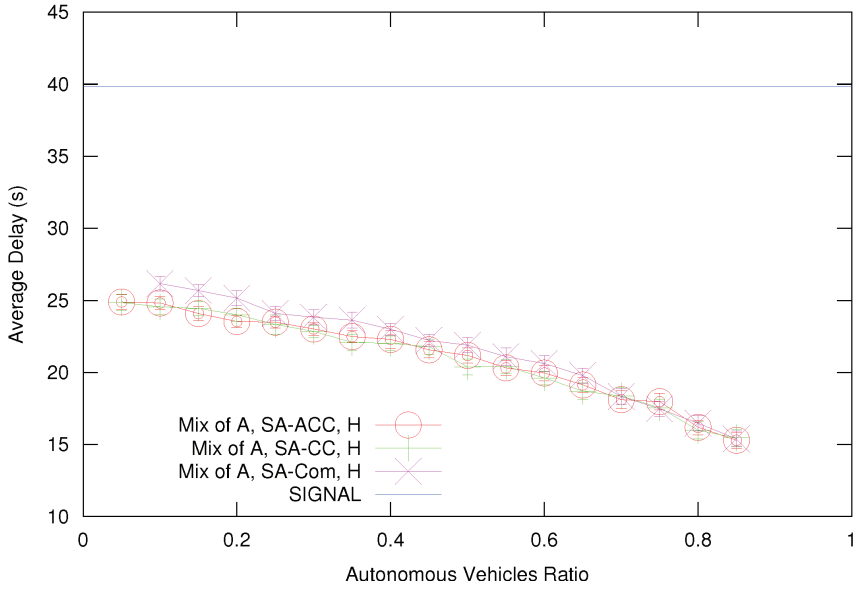


Figure 8. The average delay according to the deployment schedule in Table 3. Traffic level = 360 vehicles/lane/hour.

7. Related Work

The main contribution of our work is an extension to the FCFS-Signal policy proposed by Dresner and Stone [2]. FCFS-Signal policy allows autonomous vehicles to enter the intersection if there is no potential collision with trajectories reserved for human-driven vehicles. Their experimental results indicated that a mixture of human-driven vehicles and autonomous vehicles is possible, and leads to better performance than having all human-driven vehicles, which is the current status quo. However, their experiments indicated that the impact of autonomous vehicles is expected to be relatively small until almost all (90-95%) of the vehicles on the road are autonomous [5]. Our extension to embrace semi-autonomous vehicles shows significant performance benefits all along the technology penetration curve.

Our work is similar to the analysis of adaptive cruise control performance by Jerath and Brennan, who showed that by introducing adaptive cruise control vehicles into traffic, the vehicles would have a more *condensed* formation, thus increasing the efficiency of traffic [7]. They concretely explained the impact of adaptive cruise control (ACC) on congestion due to the formation of self-organizing traffic jams (or "phantom" jams). While the analysis of Jerath and Brennan is focused on highways, we are focusing on the intersection—a place more critical both from the points of view of congestion and safety. Selvakumar and Muthukumaran also makes use of adaptive cruise control technology, but with a different focus on the context of driving [18]. It assumes GPS equipped ACC system that (apart from performing normal ACC functions) slows down the vehicle intelligently when it enters speed restricted zones such as schools and colleges. It is also capable of detecting the speed breakers ahead and controls the vehicle dynamically according to the speed limit set for that part of the road.

There is limited literature on understanding changes to road infrastructure that can facilitate vehicular autonomy. One such project on jointly optimizing autonomous vehicles and road infrastructure is the PATH program, which relies on magnetic markers in the roadway for measuring steering angle and vehicle movements [8]. Kolodko and Vlacic used golf-cart-like Imara vehicles in evaluating an autonomous intersection [9]. Our work differs from these previous research by incorporating semi-autonomous vehicles into the mix, along with autonomous and human-driven vehicles. We do not utilize the potential change of the infrastructure. However, this can be done by dedicate certain lanes to certain type of vehicles.

Our work assumes that the vehicles run towards their destination lanes with smooth trajectories. They are mostly straight lines or smooth curves. Wuthishuwong *et al.* redesigned the trajectories for safety concerns [16]. They also used Vehicle to Infrastructure communication (V2I) approach. By dividing the space in the intersection into discrete grids, they used a dynamic programming method to determine the optimal trajectory. On the other hand, Au *el al.* did the similar work on designing trajectories when a collision is about to happen [17]. In our work, we mark the cases when the vehicles lose control as emergencies, and make the vehicles to break. However, it is possible to devise paths for the colliding vehicles to evade. The plan devised by Au *el al.* guarantees that once an evasion plan is executed, there will be no collision among

vehicles that have not yet collided. This can be incorporated into our work which involves semi-autonomous vehicles.

The complexity of communication between the intersection and autonomous vehicles is discussed in [15]. If a request is rejected by the intersection manager, the autonomous vehicle has to resend the request, which has much redundancy. Yu *et al.* proposed a *ksync* policy, which is inspired by the *rsync* algorithm that solves such redundancy. Only necessary information is transmitted at each time step. They have shown empirical results that the average compression rate of data transferred by each autonomous vehicle can be improved by over 80% based on the experimental data on the platform of the open-source AIM simulator. Such policy can also be applied to the semi-autonomous vehicles defined in our paper.

Cooperation of multiple intersections are also discussed in the literature. As a continuum of [2], [13] applies AIM policy on multiple intersections. Hausknecht *et al.* devised a navigation policy that determines the route the vehicle takes form the source to the destination. For example, one planning approach is time-based A*. As considering a network of intersections, the authors considered the effects of network topology and lane reversal. The concept of Infrastructure-to-Infrastructure (I2I) communication is proposed for an intersection to actively collaborate with its neighbor intersections [14]. An intersection can gather information from its neighbors, so a system that controls all the intersections may not be required.

Another way to manage autonomous vehicles at intersections is based on Vehicle-to-Vehicle (V2V) communication, such that vehicles coordinate in a peer-to-peer fashion when crossing the intersection [10, 11]. The anchor requests in Section 5.3 can be implemented using V2V communications. Naumann *et al.* investigated a distributed policy that uses virtual "tokens" that a vehicle must possess to cross certain contested areas of the intersection [10] and formally evaluated it using petri-nets. The information of the locations, velocities and accelerations are exchanged among the vehicles for the decision of entering the intersection. VanMiddlesworth *et al.* developed an AIM-inspired protocol that enables vehicles to "call ahead" to reserve space-time in the intersection [11]. They assumed an agent placed at the intersection that functions like a stop sign, which does not do simulation as the intersection manager does in this paper. Such policy is only efficient for small traffic level. However, only considering small traffic level may yield more efficient solutions. Neuendorf *et al.* designed a vehicle platoon controller in decentralized autonomous intersection management, which is platoon-stable and has enlarged functionality in comparison to a conventional one [12]. Some other work use Vehicle-to-Vehicle approach with different focus. For example, we can require that vehicles should cross the intersections without stopping [19]. The authors also use a simulation approach, but using a stronger mathematical model.

8. Conclusions and Future Work

This paper introduces SemiAIM, a new constraint-based autonomous intersection management system that enables human-driven vehicles and semi-autonomous vehicles, in

addition to fully autonomous vehicles, to make reservations and enter an intersection in an AIM-like style. To the best of our knowledge, SemiAIM is the first intersection control protocol to leverage limited autonomy of robotic cars to enable smooth interactions between human-driven, fully autonomous, and semi-autonomous vehicles at intersections. Our experimental results showed that our system can greatly decrease traffic delay when most vehicles are semi-autonomous, even when few (if any) are fully autonomous. Our incremental deployment study shows that traffic delay keeps decreasing as more vehicles employ features of autonomy. In the future, we intend to devise better constraint-based reservation requests using more accurate profiling of the vehicles' physical behavior.

REFERENCES

[1] M. Buehler, K. Iagnemma, and S. Singh, "The DARPA Urban Challenge: Autonomous Vehicles in City Traffic." Springer, Berlin, 2009.

[2] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management." In *Journal of Artificial Intelligence Research (JAIR)*, March 2008.

[3] D. Fajardo, T.-C. Au, S. T. Waller, P. Stone, and D. Yang, "Automated intersection control: Performance of a future innovation versus current traffic signal control." In *Transportation Research Record: Journal of the Transportation Research Board*, no. 2259, pp. 223–232, 2012.

[4] NHTSA, "Preliminary statement of policy concerning automated vehicles." National Highway Traffic Safety Administration, Technical Report, 2013.

[5] K. Dresner and P. Stone, "Sharing the road: Autonomous vehicles meet human drivers." In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[6] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles." In *American Control Conference*, 1990, pp. 291–296.

[7] K. Jerath and S. N. Brennan, "Adaptive cruise control: Towards higher traffic flows, at the cost of increased susceptibility to congestion." In *Proceedings of the 10th International Symposium on Advanced Vehicle Control (AVEC'10), Loughborough,* UK, August 22-26, 2010

[8] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown. "Automated vehicle control developments in the path program." In *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[9] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory." In *IEEE Intelligent Systems*, vol. 18, no. 4, pp. 8 – 11, July-August 2003.

[10] R. Naumann and R. Rasche, "Intersection collision avoidance by means of decentralized security and communication management of autonomous vehicles." In *Proceedings of the*

30th International Symposium on Automotive Technology & Automation (ISATA) — ATT/IST, 1997.

[11] M. VanMiddlesworth, K. Dresner, and P. Stone, “Replacing the stop sign: Unmanaged intersection control for autonomous vehicles.” In *AAMAS Workshop on Agents in Traffic and Transportation*, Estoril, Portugal, pp. 94–101, May 2008.

[12] N. Neuendorf, and T. Bruns. "The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles." In *Proceedings of the IEEE International Conference on Mechatronics (ICM)*, pp.375-380, 2004.

[13] M. Hausknecht, T.-C. Au, and P. Stone. "Autonomous intersection management: Multi-intersection optimization." In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4581-4586, 2011.

[14] C. Wuthishuwong and A. Traechtler. "Coordination of multiple autonomous intersections by using local neighborhood information." In *Proceedings of IEEE International Conference on Connected Vehicles and Expo (ICCVE*), pp. 48-53, 2013.

[15] C. Yu, G. Tan, and Y. Yu. "Make Driver Agent More Reserved: An AIM-Based Incremental Data Synchronization Policy." In *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 198-205, 2013.

[16] C. Wuthishuwong and A. Traechtler. "Vehicle to infrastructure based safe trajectory planning for Autonomous Intersection Management." In *Proceedings of IEEE International Conference on ITS Telecommunications (ITST)*, pp. 175-180, 2013.

[17] T.-C. Au, C.-L. Fok, S. Vishwanath, C. Julien, and P. Stone. “Evasion Planning for Autonomous Vehicles at Intersections.” In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1541-1546, 2012.

[18] M. Selvakumar, and D. Muthukumaran. "Real-Time Location based Adaptive Cruise Control System." In *International Journal of Engineering Sciences and Research Technology*, pp. 1359-1362, Vol. 4, No. 4, 2014.

[19] M. Tlig, O. Buffet, and O. Simonin. "Decentralized Traffic Management: A Synchronization-Based Intersection Control—Extended Version", Research Report N° 8500, INRIA (2014).

[20] K. Dresner and P. Stone. “Mitigating catastrophic failure at intersections of autonomous vehicles.” In *AAMAS Workshop on Agents in Traffic and Transportation*, pages 78–85, Estoril, Portugal, May 2008.