2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

PAIM: Platoon-based Autonomous Intersection Management

Masoud Bashiri*, Hassan Jafarzadeh* and Cody H. Fleming*
* Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA
{mb4bw, hj2bh, fleming}.virginia.edu

***Abstract*—With the emergence of autonomous ground vehicles and the recent advancements in Intelligent Transportation Systems, Autonomous Traffic Management has garnered more and more attention. Autonomous Intersection Management (AIM), also known as Cooperative Intersection Management (CIM) is among the more challenging traffic problems that poses important questions related to safety and optimization in terms of delays, fuel consumption, emissions and reliability. Previously we introduced two stop-sign based policies for autonomous intersection management that were compatible with platoons of autonomous vehicles. These policies outperformed regular stop-sign policy both in terms of average delay per vehicle and variance in delay. This paper introduces a reservation-based policy that utilizes the cost functions from our previous work to derive optimal schedules for platoons of vehicles. The proposed policy guarantees safety by not allowing vehicles with conflicting turning movement to be in the conflict zone at the same time. Moreover, a greedy algorithm is designed to search through all possible schedules to pick the best that minimizes a cost function based on a trade-off between total delay and variance in delay. A simulator software is designed to compare the results of the proposed policy in terms of average delay per vehicle and variance in delay with that of a 4-phase traffic light.**

***Index Terms*—Intelligent Transportation Systems, Autonomous Vehicles, Cooperative Intersection Management, Cooperative Adaptive Cruise Control**

I. Introduction

As bottlenecks of traffic flow, intersections are known to be a major contributor to traffic accidents. According to National Highway Traffic Safety Administration (NHTSA), [1] 40 percent of all crashes that occurred in the United States in 2008 were intersection-related. Moreover, Traffic efficiency is reported to be closely correlated with traffic safety on intersections [2]. Traffic congestion is also among the more important contributors of CO_2 emissions [3] which is the largest constituent of transport's greenhouse gas emissions. Vehicle stop times at intersections also contribute to carbon monoxide (CO) emissions. International energy agency reports that when vehicles are idle at an intersection they emit about 5–7 times as much CO as vehicles traveling between 5–10 mph [4].

Traffic Lights and stop signs are the major methods for traffic control used at intersections. While traffic lights have helped improve traffic flow at intersections, they are still considered inefficient and a contributor to traffic congestion and accidents. Statistically a majority of intersection-related accidents occur in the presence of traffic lights [1]. In the past two decades, adaptive and smart traffic light controllers have been introduced and deployed, with significant improvements in terms of delay and congestion. It has been shown that the performance of traditional traffic lights can be improved through machine learning based approaches such as Fuzzy logic [5], neural networks [6] and Reinforcement Learning [7], and mathematical models such as Mixed-Integer Linear Programming (MILP) [8]. However, traffic lights have remained a major contributor to congestion and traffic accidents. Moreover, while signalized intersections work well with human drivers, they don't necessarily leverage the advantages associated with autonomous vehicles.

Recent advancements of Information Technology and the emergence of Vehicular Adhoc Networks (VANETs) that support Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Vehicle to Pedestrian (V2P) communications have brought forth opportunities for further advancements of intersection management infrastructure. This includes new non-signalized approaches for the intersection management problem, commonly known as Cooperative Intersection Management (CIM), where road users, i.e. vehicles communicate with the infrastructure and/or other users to cooperatively coordinate the traffic flow. In 2014, the Institute of Electrical and Electronics Engineers (IEEE) published the Wireless Access for Vehicle Environments (WAVE) [9] specification. The standards define architectures based on Dedicated Short Range Communications (DSRC) for which the Society of Automotive Engineers (SAE) has specified message types and data elements through various standards.

Due to the limited capacity of current V2X communications, the communication complexity is one of the critical issues for CIM [10]. In [11] we proposed that platooning in the vicinity of intersections could reduce communication overhead by allowing platoon leaders to negotiate with the infrastructure and other platoons on behalf of the followers. Moreover, it can help improve the efficiency of *any* scheduling policy by enabling smooth trajectories in the conflict zone. The simulation results showed that the proposed stop-sign based controller outperformed a regular stop sign by 50% in terms of average delay per vehicle and 40% in variance of delay. The proposed policies decrease computational complexity by only including one platoon per lane into the schedule.

This paper introduces a reservation-based policy that utilizes cost functions that minimize delay, or a combination of delay and variance, to derive optimal schedules for platoons of vehicles. Such schedules would decrease average delay per vehicle

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

while decreasing the variance in delay due to the fairness of the cost functions and as a result increase intersection throughput and decrease the average fuel consumption in the vicinity of an intersection. The proposed policy guarantees safety by not allowing vehicles with conflicting turning movement to be in the conflict zone at the same time.

The rest of this paper is organized as follows. Section II provides a background of previous works on the CIM problem. Section III describes the vehicle dynamics model, fuel consumption model and platooning approach used in this work. Section IV introduces the proposed controller. In section V the results of simulations of a single 4-way intersection scenario are discussed and the proposed method's results are compared with that of a 4-phase traffic light controller. Conclusions are presented in section VI.

II. Cooperative Intersection Management

Several methods have been proposed to leverage autonomous and connected vehicles for the intersection management. The new methodologies for CIM can be categorized into two classes of Centralized and Distributed methods [12]. In centralized methods, a central intersection manager unit receives real-time information and requests from road users and decides how to coordinate the traffic flow, e.g. instruct vehicles how, when and if to pass the intersection. Distributed methods, on the other hand, do not rely on a central control unit. Instead, all vehicles collaboratively plan their trajectories. These methods usually involve negotiation protocols to make decisions on a high level and each user/vehicle makes decisions based on shared objectives given local information from its sensors on a lower level.

A. Centralized Methods

In [13], Dresner et al. propose a centralized resource reservation algorithm based on First Come First Serve (FCFS) policy. The control unit receives requests from all vehicles approaching the intersection, simulates the vehicle's movement through the intersection, given the information in the request, and confirms the request if there is no conflict with previously accepted trajectories, otherwise the request would be rejected and the vehicle has to request at a later time. The authors assume constant speed at the intersection and perform simulations comparing results with a traffic light and an overpass. It was shown that the proposed method outperformed traffic light in terms of average vehicle delay. The authors further developed their work in [14] to add several methods to improve the performance and overcome major disadvantages of the previous work. Moreover, they improved the work in [15] and [16] by adding support for human drivers, considering emergency vehicles that generally have higher priority, and communication schemes for vehicles with different levels of autonomy.

In [17], a case of two merging roads (one lane) is modeled as a polling system with two queues and one server. The polling system determines the sequence of times assigned to the vehicles on each lane to enter the merging road. The arrival times along with the trajectories of the leading vehicle are then used in a coordination algorithm to generate optimal trajectories for each vehicle in the queue.

Lee and Park [18] derive a nonlinear constrained optimization problem to enhance the performance a traffic signal controller in presence of fully autonomous vehicles. A phase conflict map of the traffic signal is used as part of the optimization problem.

B. Decentralized Methods

In [19] a controller model has been proposed along with different V2V-based intersection management protocols to enhance traffic throughput and safety. Each vehicle runs a collision avoidance algorithm that takes in all safety messages that are being broadcast by surrounding vehicles and detects possible collision. Estimations are then used to generate alternative trajectories to avoid collisions.

Wu et al. [20] proposed a decentralized stop-and-go based algorithm that relies on wireless shared information among all approaching vehicles. The vehicle with a shorter estimated arrival time will cross the intersection, while others will need to come to a complete stop until the conflict zone is cleared. Vehicles with non-conflicting turning movements can cross simultaneously.

C. Motivation and Contributions

The existing approaches are limited with respect to at least one of the following factors:

- unrealistic or infeasible bandwidth requirements for communication
- no performance guarantees, i.e. no guarantee that the intersection will behave better than signalized intersections
- no formal safety guarantee
- scalability with respect to number of cars and lanes
- unrealistic assumptions about vehicle behavior

Any solutions for the CIM problem has to be compatible with real world communication capacity of vehicular networks, otherwise such solutions will not be feasible. To address this issue we propose a platoon-based approach where vehicles request a pass as a platoon. The intersection manager utilizes an optimization based policy to allocate slots in time and space for any platoon approaching the intersection. Communicating with platoon leaders, instead of every approaching vehicle, decreases the amount of communication needed. This approach also takes advantage of recent advances in platooning and connected vehicle control.

The performance of the solution has to be verified with respect to various performance metrics such as average delay, intersection throughput, average speed, fuel consumption, etc. We conduct microscopic simulations to verify the performance of the proposed method in comparison to a pre-timed 4-phase traffic light controller. Table I shows the performance metrics used in this work.

The solution has to be scalable to growing number of roads/lanes and vehicles. We achieve a scalable solution by only taking the information from the closest platoons to

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

the intersection at each iteration. This guarantees that the computational complexity of the algorithm remains relatively unchanged as the size of the input grows.

Most of the previously published papers make simplistic assumptions about vehicle dynamics, e.g. second order dynamics where the control input generates instant accelerations, disregarding disturbance forces, and oversimplifying the driving force, which is a function of throttle and brake positions among other factors. In this paper, we utilize a model for vehicle dynamics to generate more realistic results when compared to the relevant literature. These assumptions help design control strategies that are more feasible for use in the real world.

TABLE I: Performance Measure Index

Performance Measure Index	
Performance Index	Unit
Average Delay	s
Delay Standard Deviation	s
Intersection Throughput	veh/hour
Fuel Consumption	ml/veh

III. Vehicle Dynamics and Platooning

In our previous work, a simple linear model was assumed for the vehicle dynamics. Such simplistic assumptions can be found in many of the work in Section II. In what follows, however, a nonlinear model has been adopted and modified. The rest of this section provides the details of this model, which is divided into two parts: equation of motion and rotational velocity.

A. Vehicle Dynamics

We modify the original nonlinear dynamics found in [21] in order to represent the braking and turning required to traverse an intersection.

1) Equations of Motion: A general form of the equation of motion is Equation 1 which is a force balance for the car body.

$$m\frac{dv}{dt} = F - F_d \tag{1}$$

In the above equation, m is the total mass of the car, v is the car's speed, F is the traction force generated from the contact of wheels with the road and F_d is the sum of all disturbance forces due to gravity, friction and aerodynamic drag (Equation 2).

$$F_d = F_g + F_r + F_a \tag{2}$$

Equation 3 models the driving force generated by the engine. The engine torque T is a function of the throttle position controlled by control input $0 \leq u \leq 1$, and the engine speed ω (Equation 4). α_n is the inverse of the effective wheel radius.

$$F = \alpha_n u T(\alpha_n v) \tag{3}$$

$$\omega = \frac{n}{r}v = \alpha_n v \tag{4}$$

In the above equation, n is the gear ratio, r is the wheel radius and v is the speed of the car. In this work, the torque is modeled by the simple representation in Equation 5, where β is a design parameter of the engine and T_m is the maximum torque obtained from the maximum engine speed ω_m. Typical values for these parameters are, $T_m = 190Nm$, $\omega_m = 420rad/s$ and $\beta = 0.4$.

$$T(\omega) = T_m(1 - \beta(\frac{\omega}{\omega_m} - 1)^2) \tag{5}$$

Equations 6, 7 and 8 represent the disturbance forces due to gravity, friction and aerodynamic drag respectively.

$$F_g = mgsin(\theta) \tag{6}$$

$$F_r = mgC_r sgn(v) \tag{7}$$

$$F_a = \frac{1}{2}\rho C_d A v^2 \tag{8}$$

Equation 9 represents the complete form of the equation of motion, where α_n is the reverse of effective wheel radius, given the nth gear position. ρ is the density of air, C_r is the coefficient of rolling friction, C_d is the aerodynamic drag coefficient and A is the frontal area of the car. In this work we use typical values for these parameters as follows: $\alpha_1 = 40$, $\alpha_2 = 25$, $\alpha_3 = 16$, $\alpha_4 = 12$, $\alpha_5 = 10$, $C_r = 0.01$, $\rho = 1.3\frac{kg}{m^3}$, $C_d = 0.32$ and $A = 2.4m^2$. The slope of the road θ is assumed to be zero in this work, thus eliminating the influence of the gravity as a disturbance force in the experiments.

$$\begin{aligned} m\frac{dv}{dt} &= \alpha_n u T(\alpha_n v) - mgC_r sgn(v) \\ &- \frac{1}{2}\rho C_d A v^2 - mgsin\theta \end{aligned} \tag{9}$$

We modify the model in 9 to add a braking component to the system. The braking is modeled as an extra friction component to the equation of motion. The control input u is replaced with two new control inputs $0 \leq h \leq 1$ and $0 \leq b \leq 1$, representing throttle and braking commands respectively, where both are functions of the computed control input $-1 \leq u' \leq 1$ (Equations 10, 11).

$$h(u') = \begin{cases} u' & \text{if } u' > 0 \\ 0 & \text{if } u' \leq 0 \end{cases} \tag{10}$$

$$b(u') = \begin{cases} \mid u' \mid & \text{if } u' < 0 \\ 0 & \text{if } u' \geq 0 \end{cases} \tag{11}$$

Equation 12 shows the modified equation of motion.

$$\begin{aligned} m\frac{dv}{dt} &= \alpha_n h T(\alpha_n v) - mg(C_r + b)sgn(v) \\ &- \frac{1}{2}\rho C_d A v^2 - mgsin\theta \end{aligned} \tag{12}$$

2) Rotational Velocity: Given the model above, the rotational velocity of the vehicle can be calculated by Equation 13.

$$\frac{d\theta}{dt} = \frac{v_0}{L}tan\delta \tag{13}$$

where θ is the vehicle's orientation, v_0 is the speed of the rear wheel(s), L is the wheel base and δ is the steering angle.

B. Fuel Consumption Model

For approximating the fuel consumption of the vehicles, here we adopt a differentiable function of velocity and ac-

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

celeration introduced in [22], where two polynomial estimation functions of velocity and acceleration are approximated through curve-fitting process, given real-world data obtained from a typical vehicle with similar properties compared to the assumptions of this work. The fuel consumption in ml/s is estimated as:

$$f_V = (f_{cruise} + f_{accel}) \tag{14}$$

Where, f_{cruise} represents fuel consumption rate at a steady velocity of v, and f_{accel} is the additional consumption due to presence of acceleration a at velocity v.

$$f_{cruise} = b_0 + b_1 v + b_2 v^2 + b_3 v^3 \tag{15}$$

$$f_{accel} = a(c_0 + c_1 v + c_2 v^2) \tag{16}$$

The consumption parameters are $b_0 = 0.1569$, $b_1 = 2.450 \times 10^{-2}$, $b_2 = 7.415 \times 10^{-4}$, $b_3 = 5.975 \times 10^{-5}$, $c_0 = 0.07224$, $c_1 = 9.681 \times 10^{-2}$, and $c_2 = 1.075 \times 10^{-3}$. Details of the formation and determination of parameters of this fuel consumption model are described in [23].

C. Cooperative Adaptive Cruise Control (CACC)

The CACC (a.k.a platooning) problem [24] has been widely studied in the literature and several solutions have been proposed [25], [26], [27]. We adopt the Predecessor-Leader-Follower (PLF) information flow topology to describe the communication between the vehicles in a platoon.

A proportional-integral-derivative (PID) controller is designed and tuned given the vehicle dynamics to achieve string stability. An interested reader can view a video[1] of a step response of the controller in a simple scenario where a platoon of five vehicles starts from a standstill and is supposed to reach the velocity of $30\frac{mi}{h}$, come to a complete stop at the intersection and accelerate to $25\frac{mi}{h}$.

IV. Platoon-based Autonomous Intersection Management

In this work, we extend the policies from our previous work, create an application layer communication protocol and eliminate some of the simplistic assumptions about vehicle dynamics to achieve the following goals:

1) Achieve a solution under realistic assumptions
2) Improve efficiency in terms of delay and fuel consumption
3) Design a communication protocol that hides the algorithm from the vehicle agents

A. Communication Protocol

A protocol is designed to ease the Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications. Using this protocol, the intersection manager only has to communicate with the leader of a platoon. Each vehicle is broadcasting its state on a $10Hz$ frequency, while receiving incoming packets on the same frequency. There are four types of packets designed for the platoon leaders and two types for the infrastructure as follows.

[1] https://youtu.be/uuxMVm0MWDQ

Vehicle Message Types:

- *Request*
- *Change-Request*
- *Acknowledge*
- *Done*

Infrastructure Message Types:

- *Acknowledge*
- *Confirm*
- *Reject*

Approaching platoon leaders send a $Request$ message once they are in a pre-defined proximity of the intersection, and await the response from the manager. Both *Request* and *Change-Request* messages consist of the unique Vehicle Identification Number (VIN) of the leader, current position, velocity and acceleration of the leader, estimated arrival time at the conflict zone and the size of the platoon, e.g. number of followers.

Upon receiving a request, the manager sends an *Acknowledge* message to the sender, runs the scheduler and responds with either an *Confirm* or a *Reject* message. The manager expects to receive an *Acknowledge* from the corresponding vehicle and if such message is not received, it re-sends the message until the acknowledgement is received. The vehicles can send a *Change-Request* message to the manager if they need to update information in their previous request. Once a vehicle has finished crossing the conflict zone, it is required to send a *Done* message to the manager, which lets the manager know it can remove the corresponding vehicle's platoon from the schedule.

This protocol is designed to hide the scheduling policy from the vehicle agents, therefore allowing for online changes to the policy without requiring any changes in communications.

B. Policy

The scheduling problem is formulated as the following minimization problem.

$$arg\min_{s} \delta(s) = \{s \mid s \in S \wedge \forall s' \in S : \delta(s) \leq \delta(s')\} \tag{17}$$

Where s is a schedule of vehicles (platoons), δ is a cost function designed to penalize total delay and variance in delay, given a schedule, as in Equations 18 and 19.

$$\delta_1(s) = \sum_{j=2}^{N} j \left(d(p_j) + \sum_{i=1}^{j-1} t_c(p_i) \right) \tag{18}$$

$$\delta_2(s) = \sum_{j=2}^{N} \left(d(p_j) + \sum_{i=1}^{j-1} t_c(p_i) \right) \tag{19}$$

In the above equations, N is the number of platoons in the schedule, j is the platoon's turn, d is a function that returns the sum of the current delay of the vehicles within a given platoon, and again s is the given schedule. This delay, d, is the difference between the vehicle's original expected arrival time and its expected arrival time assuming it will be the next platoon going through. t_c is a function that computes the additional delay caused by the platoons that are given higher priorities than the jth platoon in the schedule.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

The scheduler in Equation 17 essentially simulates all possible schedules given the set of platoons in the queue, and returns a schedule that has the lowest score in terms of the cost function $\delta_1(s)$ or $\delta_2(s)$. For the rest of this paper we will refer to $\delta_1(s)$ and $\delta_2(s)$ as *Platoon-based Variance Minimization* (*PVM*) and *Platoon-based Delay Minimization* (*PDM*), respectively.

The scheduling procedure will be called by the controller every time a change is detected in the set of platoons in the schedule. For example, the procedure is called when a platoon finishes crossing the conflict zone and as a result is removed from the schedule, or when a new platoon enters the communication zone. The controller keeps a record of the most recent schedule and sends a message to the platoon that is at the top of the queue. The controller also checks for non-conflicting turning movements in the schedule with that of the platoon at the top of the queue, and lets them cross the intersection simultaneously. The clearance time of the intersection is then updated by the maximum estimated clearance time of the platoons that are crossing the conflict zone.

Algorithm 1 shows the intersection management algorithm in pseudo code.

Algorithm 1 Autonomous Intersection Manager

```
function IntersectionManager
    while True do
        P = getRequests() [a]
        sort(P) [b]
        pool = selectCandidates(P)
        if !pool.isUpdated() then
            continue [c]
        end if
        [platoons, schedule] = getSchedule(pool)
        i = 1
        for platoon in platoons do
            update(platoon, schedule_i)
            i++
        end for
    end while
end function
```

[a] Where P is a map of platoons paired with their respective request information
[b] Sort the platoon list based on their expected arrival time to make the candidate selection run faster
[c] Skip this iteration if the selection pool has not changed

The algorithm considers at most one platoon for each lane that is in the communication range of the central controller, i.e. the leading platoon in each lane. Such design makes the policy scalable, in that the computational complexity of the scheduler remains suitably low as the number of incoming lanes grow.

C. Computational Complexity

The algorithm considers all permutations of the set of platoons, including possible non-conflicting trajectories in which case simultaneous crossing is considered. These permutations can be modeled as a permutation problem to pick from K elements without replacement and placing them in sets of $\{K, K-1, ..., 1\}$ placeholders. Therefore, the worst case computational complexity of such algorithm would be equal to the number of possible permutations, given in Equation 20.

$$T = \sum_{r=1}^{N} \sum_{i=0}^{r-1} (-1)^i \binom{r}{i} (r-i)^N \tag{20}$$

In the above equation, T is the total number of possible schedules, N is the number of lanes, r and i are index counters. One may note that the exponential nature of this complexity can only be acceptable for small number of incoming lanes. It can be shown that the algorithm will have the worst case computational complexity of $O(N^N)$. For example, the 4-way intersection considered in this work would only require 75 possible schedules to be considered in the worst case. Due to space constraints, the details and proof of computational complexity are omitted.

We propose a heuristic for larger intersections to reduce computational complexity. The proposed heuristic does not consider non-conflicting trajectories and therefore reduces the complexity to the number of possible permutations which is exactly $N!$, i.e. computational complexity of $O(N!)$. After a schedule is selected, the controller allows those platoons with non-conflicting trajectories with regard to the selected platoon to cross the intersection simultaneously. Fig. 1 demonstrates how the proposed heuristic ignores the platoons behind the closest platoon to reduce the computational complexity. The platoons/cars in red represent the ignored input to the algorithm. This figure also serves as a visual aid to represent the geometry and turning policy of the intersection that was used for the simulations.

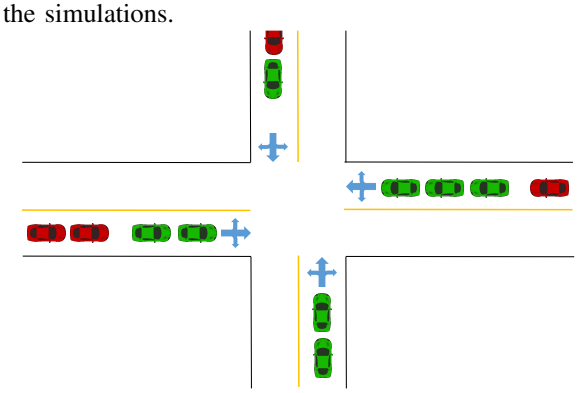

Fig. 1: 4-way Intersection Geometry

V. Simulations

A fixed-time 4-phase traffic light controller is tuned for the 4-way intersection shown in Fig. 1. The phase plan and timing diagram of the baseline traffic light policy are shown in Figs 2 and 3 respectively.

To compare the performance of the two proposed methods against the baseline policy, we design 20 scenarios by choosing different parameter settings in terms of incoming traffic flow

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

and maximum platoon size. Each policy is evaluated for every variation of the two parameters in table II. The incoming traffic flow is equal on all approaches with 70% of the traffic going straight, 20% turning right and 10% turning left. To minimize the influence of randomness, each simulation is run for 60 minutes. Recorded videos of simulation for *PVM*[2] and *PDM*[3]

TABLE II: Simulation Parameters

Parameter Name	Set of Values	unit
Traffic Level	{500,600,700,800}	veh/hour/lane
Simulation Time	{3600}	s
Maximum Platoon Size	{1,2,3,4,5}	veh

are available. Figs 4, 5, 6 and 7 demonstrate the results in terms of delay per vehicle, delay standard deviation, intersection capacity and fuel consumption, respectively. To conserve space and promote readability, the results are aggregated over the set of values for the traffic level parameter.

According to Fig. 4, the proposed methods significantly outperform the traffic light controller in terms of average delay per vehicle. Fig. 5 shows the computed standard deviation of delays throughout the entire simulations for each policy and the set of maximum platoon sizes. It can be seen that the *PVM* method significantly decreases the standard deviation compared to the traffic light, while as expected, *PDM* does not show a meaningful improvement in terms of delay variance as its cost function is designed to solely decrease total delay. One may also note that the maximum platoon size and the *PVM* and the traffic light performance are negatively correlated, which confirms the positive effect of platooning on the performance of any type of traffic controller.

Fig. 6 shows that platoon size and intersection capacity are positively correlated for all policies. For the traffic light policy, larger platoons result in more smooth trajectories with shorter headways, and as a result increases the outgoing traffic flow. Larger platoons also help the proposed policies select better schedules in terms of total delay and variance, which ultimately will increase the outgoing traffic flow. In the simulations, *PVM* policy outperformed traffic light for all incoming traffic flows and platoon sizes.

Fig. 7 demonstrates fuel consumption per vehicle as a function of platoon size. As expected, platoon size and fuel consumption are strongly correlated. *PVM* and *PDM* policies outperform traffic light by 8% and 13% on average, respectively. This result can be explained by the shorter idle times

[2]https://youtu.be/RtN0f7BlFyg
[3]https://youtu.be/qHGv9LF72NA

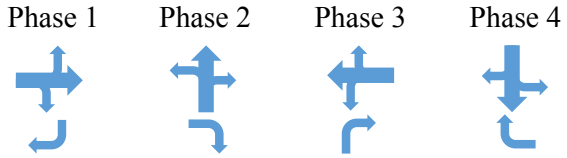

Fig. 2: Phase Plan

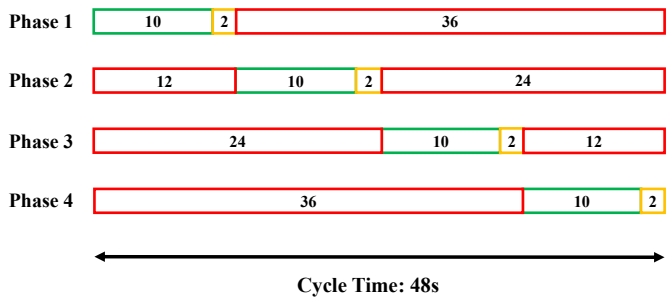

Fig. 3: Timing Diagram

generated by *PVM* and *PDM* compared to the traffic light policy. According to the fuel consumption model adopted in the simulations, the vehicles consume fuel at a rate of 0.15 ml/s when idle.

To compare the overall performance of the policies, results from all configurations are aggregated into table III. All the metrics in this table are averaged over the set of incoming traffic flows that range from 500 to 800 $v/h/l$. Traffic flows are identical for each approach.

The *PVM* method outperforms the traffic light policy on all four metrics. More notably, it decreased average delay per vehicle by factor of 6.56× and decreases the standard deviation to 4.9×, resulting in faster and more reliable traffic flows. The *PVM* policy also increased the intersection capacity by 13.8% compared to traffic light.

The *PVM* and *PDM* policies both outperform the traffic light in terms of fuel consumption by 8% and 13% respectively.

VI. Conclusions

In this paper, a centralized platoon-based controller was proposed for the cooperative intersection management problem that takes advantage of the platooning systems to generate fast and smooth traffic flow at an intersection. A simple communication protocol was designed for V2I communication and two policies were introduced for the controller to minimize total delay and delay variance according to the cost functions tailored for platoons of vehicles.

According to the simulation results, the proposed controller minimizes travel delay and variance while increasing intersection throughput and reducing fuel consumption, when compared to traffic light policies. The simulations also verify the positive effect of platoon size on fuel consumption and intersection throughput.

References

[1] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," Tech. Rep., 2010.
[2] G.-L. Chang and H. Xiang, "The relationship between congestion levels and accidents," Tech. Rep., 2003.
[3] M. Grote, I. Williams, J. Preston, and S. Kemp, "Including congestion effects in urban road traffic co2 emissions modelling: Do local government authorities have the right options?" *Transportation Research Part D: Transport and Environment*, vol. 43, pp. 95–106, 2016.
[4] I. Statistics, "Co2 emissions from fuel combustion-highlights," *IEA, Paris http://www. iea. org/co2highlights/co2highlights. pdf. Cited July*, 2011.
[5] J. Niittymäki and M. Pursula, "Signal control using fuzzy logic," *Fuzzy sets and systems*, vol. 116, no. 1, pp. 11–22, 2000.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.

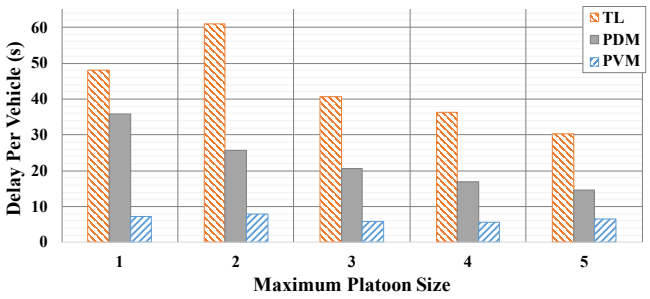

Fig. 4: Average Delay Per Vehicle

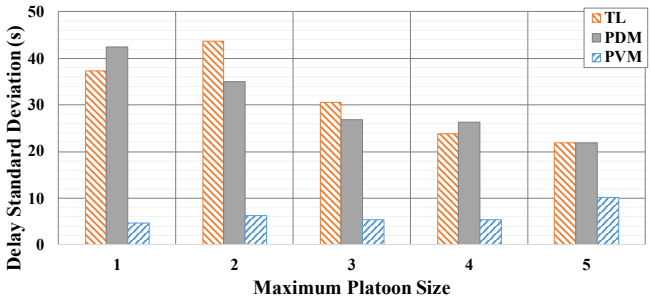

Fig. 5: Delay Standard Deviation

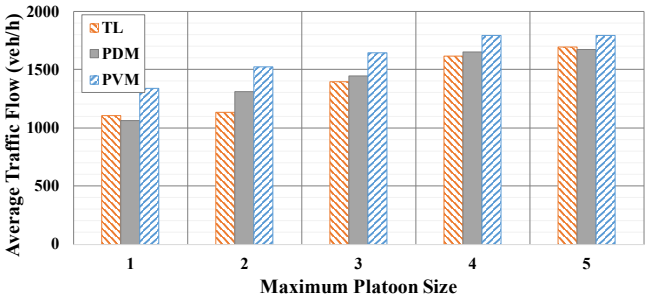

Fig. 6: Intersection Traffic Flow

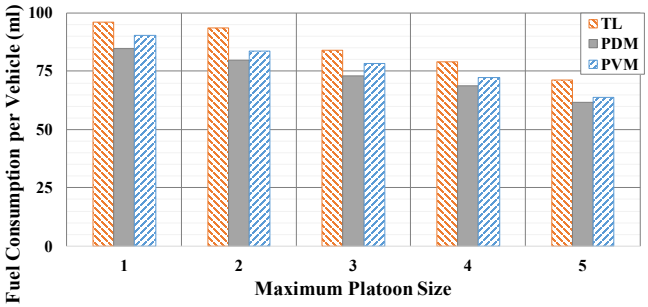

Fig. 7: Fuel Consumption

TABLE III: Aggregated Results

	Traffic Light	PVM	PDM	PVM-Improvement	PDM-Improvement
Delay(s)	43.26	**6.56**	22.71	**6.6×**	1.9×
Capacity(veh/h)	1388	**1617**	1426	**13.8%**	2.7%
FCPV(ml/v)	84	77	**73**	8%	**13%**
STDEV(s)	31.44	**6.37**	30.48	**4.9×**	3%

[6] J. C. Spall and D. C. Chin, "Traffic-responsive signal timing for system-wide traffic control," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 3-4, pp. 153–163, 1997.

[7] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[8] W.-H. Lin and C. Wang, "An enhanced 0-1 mixed-integer lp formulation for traffic signal control," *IEEE Transactions on Intelligent transportation systems*, vol. 5, no. 4, pp. 238–245, 2004.

[9] I. S. Association *et al.*, "Ieee guide for wireless access in vehicular environments (wave) architecture," *IEEE Std*, pp. 1609–0, 2013.

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

[11] M. Bashiri and C. H. Fleming, "A platoon-based intersection management system for autonomous vehicles," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 667–672.

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

[13] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 2004, pp. 530–537.

[14] ——, "Multiagent traffic management: An improved intersection control mechanism," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM, 2005, pp. 471–477.

[15] ——, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[16] G. Sharon, S. D. Boyles, and P. Stone, "Intersection management protocol for mixed autonomous and human-operated vehicles," *Transportation Research Part C: Emerging Technologies (Under submission TRC-D-17-00857)*, 2017.

[17] D. Miculescu and S. Karaman, "Polling-systems-based control of high-performance provably-safe autonomous intersections," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 1417–1423.

[18] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[19] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Intersection management using vehicular networks," SAE Technical Paper, Tech. Rep., 2012.

[20] W. Wu, J. Zhang, A. Luo, and J. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 65–74, 2015.

[21] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[22] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Model predictive control of vehicles on urban roads for improved fuel economy," *IEEE Transactions on control systems technology*, vol. 21, no. 3, pp. 831–841, 2013.

[23] ——, "Ecological vehicle control on roads with up-down slopes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 783–794, 2011.

[24] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on automatic control*, vol. 38, no. 2, pp. 195–207, 1993.

[25] A. Iihoshi, S. Kobayashi, and Y. Furukawa, "Vehicle platoon control system," Feb. 29 2000, uS Patent 6,032,097.

[26] J. Hedrick, D. McMahon, V. Narendran, and D. Swaroop, "Longitudinal vehicle controller design for ivhs systems," in *American Control Conference, 1991*. IEEE, 1991, pp. 3107–3112.

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

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on July 08,2020 at 22:48:48 UTC from IEEE Xplore. Restrictions apply.