Centralized Coordination of Autonomous Vehicles at Intersections

Chamideh, Seyedezahra; Tärneberg, William; Kihl, Maria

Published in:
2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)

DOI:
10.23919/SoftCOM50211.2020.9238311

2020

Centralized Coordination of Autonomous Vehicles at Intersections

SeyedeZahra Chamideh
Department of Electrical and Information Technology
Lund University, Sweden
Email: Seyedezahra.chamideh@eit.lth.se

William Tärneberg
Department of Electrical and Information Technology
Lund University, Sweden
Email: william.tarneberg@eit.lth.se

Maria Kihl
Department of Electrical and Information Technology
Lund University, Sweden
Email: maria.kihl@eit.lth.se

***Abstract*—Recent advances in autonomous vehicles present new opportunities in Intelligent transportation systems (ITS) to address urban transport challenges. Therefore, urban traffic scenarios, and in particular intersections as a bottleneck of transportation network, has received significant attention. In this paper we investigate intelligent traffic control mechanisms for autonomous vehicles at intersections as a replacement of traditional intersection control. An edge cloud controller is used to deliver services that provide traffic safety and efficiency to vehicles. Two well-cited optimization algorithms for cooperative vehicles are compared with realistic simulations in SUMO. We investigated the safety and possibility of implementing the proposed algorithms in the real world. This side by side comparison helps to gain insight into the strengths and limitations of these types of algorithms.**

I. Introduction

Intelligent Transportation Systems and in particular autonomous vehicles (AVs) will likely have significant effect on future traffic management systems. As automated vehicles become more common, the traffic control strategies, for example intersection management (IM), have to be improved in order to increase the driving safety. Road intersections are currently managed by using traffic lights, which often result in many vehicles unnecessarily braking and can significantly increase travel times. By leveraging the capacities of AVs, it is possible to remove traditional intersection managers and rely on coordination among the involved vehicles at an intersection.

In cooperative intersection control, there is usually an intersection control unit that can exchange information with the vehicles. In this paper, we assume that all vehicles crossing the intersection can be manipulated by the control unit through two-way communication. Cooperative intersection control could, besides providing safe crossings for the vehicles, optimize overall costs such as travel times, traffic throughput and fuel consumption.

This work was partially supported by the Excellence Center at Linköping-Lund on Information Technology (ELLIIT). Maria Kihl and William Tärneberg are also partially supported by the SEC4FACTORY project, funded by the Swedish Foundation for Strategic Research (SSF), the 5G PERFECTA Celtic Next project funded by Sweden's Innovation Agency (VINNOVA), and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Also, the authors are part of the Nordic University Hub on Industrial IoT (HI2OT) funded by NordForsk.

The idea of Autonomous Intersection Management (AIM) was first proposed by Drenser and Stone [1]. They designed a central decision maker, which manages time-space reservations in the intersection to avoid collisions. After that, a number of papers has been published on methods to design the best intersection management system as well as several metrics to evaluate such systems. For example, a vehicle scheduling problem was proposed in [2]. In their algorithm, vehicle agents are allowed to determine control actions among a set of controlled inputs. Another example is [3], where the authors proposed a decomposition scheme that gives an approximation solution to an optimal control problem. Further, a convex modelling for optimal control of autonomous vehicles at intersections was provided in [4]. Their proposed method includes problem transformation from time to space domain. Also, in [5], a communication strategy was proposed that minimized the use of communication resources for the intersection management.

In Kamal *et al.* [6] a Model Predictive Control (MPC) problem is formulated that generates feasible trajectories for autonomous vehicles. Their control algorithm optimises the control inputs of the vehicles in a given time horizon to minimise the risk of cross-collisions. They assume a constant number of vehicles approaching to the intersection and input traffic into all the sections is set at equal rate.

In Zohdy *et al.* [7], a system for intersection control is developed that optimizes vehicle trajectories within an intersection zone. In their strategy, vehicles pass the critical area with a time difference to avoid collision. They assume that vehicles approach the intersection zone with their maximum possible velocity and the proposed mechanism reduce this speed if it is required.

The overall aim of our research work is to develop a robust centralised controller in a scenario where vehicles are cooperative and connected to an edge cloud based on a 5G infrastructure. The controller should generate optimal route decisions for all vehicles based on the intersection state, and then communicate these decisions to the vehicles that follows the decisions when crossing the intersection. Obviously, an autonomous intersection control algorithm must guarantee total safety for passengers, that is, no collision can occur, which means that the collision probability must be zero for all possible scenarios. Further, the execution time of an algorithm must be limited, in order for it to be feasible to run in real-time.

In this paper, we compare the previously mentioned algorithms [6] [7] by implementing them in the realistic simulation environment SUMO [8]. We have chosen these two algorithms because they have different optimization objectives for the same type of intersection control, and they are well cited in the literature. Both algorithms have been shown to work well in numerical simulations and theoretical analysis. In this paper, we evaluate the safety and possibility of implementing the proposed algorithms in the real world. Our investigation shows the performance of the two algorithms compared with traditional signalised method. Our main conclusion is that these algorithms can only be used for low traffic densities, since the collision probability rather quickly becomes larger than zero when the traffic increases. Also, the execution times of the algorithms makes them rather infeasible for realistic traffic scenarios.

II. Targeted System

In urban transportation network, intersections are a bottleneck in generating traffic congestion. Traffic flow pattern in an intersection depends on its geometry, location and possible movement to and from its various lanes. Congestion wastes a massive amount of time, fuel and creates more uncertainty for traveller. Coordinated intersection traffic management is an important component of the intelligent transportation system. It enables a vehicle to communicate with roadside equipment or other vehicles, and help to improve the road traffic safety and efficiency.

We consider the problem of autonomous vehicle coordination at a crossroad intersection without traffic lights, as depicted in Fig. 1, The system is composed of vehicles equipped with On Board Units (OBUs), which may employ a wide range of sensors types, and an Intersection Coordination Unit (ICU) that will be deployed at the intersection and act as a computing resource pool. The OBUs and the ICU communicates via some type of high bandwidth radio communication, for example using 5G. However, the focus in this paper is on the control part, not the communication part, and therefore, the communication technology is not specified, and we just assume a wireless communication link with negligible packet loss and delay.

We consider a typical crossroad intersection where vehicles are allowed to make left and right in addition to through movement. We split the intersection area into three zones: the entrance zone, the critical zone and the exit zone. The entrance zone represents the area where vehicles approaches the intersection boundary. In the critical area, there is a risk of lateral vehicle collisions. The exit zone includes vehicles leaving the intersection.

Each vehicle periodically sends its status information to the ICU, as part of the ITS facilities layer [9]. The ICU will periodically orchestrate vehicles, aggregate data, and provide control based on the system's objectives. Also, the ICU will determine whether there is any danger according to the driving status of the vehicles. Collisions between two vehicles are prevented by controlling the speed of the vehicles. All vehicles are assumed to always follow the ICU's decisions.

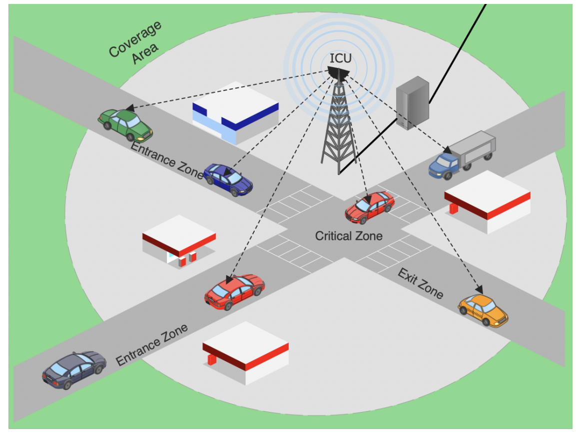

Fig. 1. Targeted system and intersection layout

III. Investigated Algorithms

In this paper, we investigate the performance of two AIM algorithms, the *Model Predictive Control* Algorithm [6] and the *Delay Minimization* Algorithm [7], by implementing them in the realistic simulation environment SUMO [8]. In this section, we first describe our general system model and then present high level descriptions of the two algorithms.

A. System Model

In this section we present the system model that is used by the AIM algorithms for intersection control. A collection of $\mathcal{N} = \{1, 2, \dots, N\}$ autonomous vehicles approaches a coordination area of an intersection with L lanes. For each $i \in \mathcal{N}$ a predetermined path is given and perfectly followed.

The vehicle dynamics are described as a second order integrator, where the vehicle is modeled as a point on the path coordinates. Given vehicle i, $p_{i,t}$ is defined to show the position (distance from the beginning of the critical zone) at time t and $v_{i,t} = \dot{p}_{i,t}$ is the speed of vehicle and $u_{i,t} = \ddot{p}_{i,t}$ the acceleration [10]. The longitudinal motion of each vehicle, $x_{i,t} = [p_{i,t}, v_{i,t}]^\intercal$, can be controlled by its acceleration. We assume the control input is updated in discrete time τ. The discrete time state model of vehicle i is given in equation (1).

It is assumed that the vehicles will follow the acceleration decided by ICU through communication link describe in section II. each vehicle will transmit its basic driving information, including current position, velocity and destination once entered into the intersection zone to initialised the problem at the ICU.

$$x_{i,t+1} = \begin{bmatrix} 1 & -\tau \\ 0 & 1 \end{bmatrix} x_{i,t} + \begin{bmatrix} -\frac{1}{2}\tau^2 \\ \tau \end{bmatrix} u_{i,t} \tag{1}$$

We consider a limited speed and acceleration: $v_{i,t} \in V_i = [v_{min}, v_{max}]$ and $u_{i,t} \in U_i = [u_{min}, u_{max}]$. However, maximum and minimum acceleration of each vehicle depends on its current velocity and speed limitation.

B. Model Predictive Control algorithm

In this section, we give a high-level description of the AIM algorithm for intersection control proposed in [6]. This algorithm will in the rest of the paper be called the *Model Predictive Control (MPC) algorithm*. The algorithm defines an optimization problem that minimize the risk of collision between a pair of vehicles at their possible conflict point during a finite time horizon. Therefore, the algorithm introduces a risk function $\mathcal{F}_{i,j}^{t}$ that is used to determine whether a vehicle pair (i, j) poses a potential risk of collision at time t. The risk function is given in equation (2) below:

$$\mathcal{F}_{i,j}^{t} = \delta_{i,j} exp\{-\alpha_i(p_{i,t} + C_{ij})^2 - \alpha_j(p_{j,t} + C_{ji})^2\} \tag{2}$$

Here, α_i and α_j are positive constants that depend on the two vehicles' sizes. $\delta_{i,j}$ is a binary variable that states whether the pair vehicles (i, j) have the potential to collide or not. C_{ij} and C_{ji} are the distances from the conflict point of the pair vehicles (i, j) to the beginning of the critical zone at intersection for vehicle i and j respectively.

In order to avoid any rear-ends collisions, a minimum separation distance between two vehicle on the same lane, d_{min}, is defined. Since the vehicles are modeled as a point on their path, In the real world d_{min} is the minimum distance between the centre point of two vehicles. The following constraint in equation (3) is defined to prevent rear-ends collisions between vehicles i and j at time t:

$$|p_{i,t} - p_{j,t}| \geq d_{min} \tag{3}$$

The variable $p_{i,t}$ will always have a positive value in the entrance zone. For ensuring that no collisions occur between two vehicles form different approaching lanes inside the critical zone, a linear inequality constraint is defined as in equation (4) below:

$$p_{i,t} + C_{ij} + p_{j,t} + C_{ji} \geq R_{min} \tag{4}$$

Here, R_{min} is a constant that denotes the minimum separation distance between the centers of two vehicles from different approaching path.

A Model Predictive Control (MPC) framework is used to minimize the system's cost over predefined time horizon. An MPC problem with the time horizon of T steps allows the system be optimized in current time slot, while keeping next $T - 1$ time slots in account [11], [12]. For this purpose at each time step an optimization problem is solved to drive the optimal control input for the system by predicting the system state over the defined time horizon.

The objective of the optimization is to achieve a smooth and comfortable flow of vehicles where the vehicles cross the intersection with almost constant and high speed, while minimizing the risk of collisions and energy consumption. Therefore, The system cost, J is defined as in (5). One of the term in cost function attempts to minimise error between the speed of vehicle i and its desired speed, v_d^i, Minimizing the acceleration, $u_{i,t}$ and the last term related to collision avoidance risk function.

$$\begin{aligned} J = & \sum_{t=0}^{T-1} \sum_{i=1}^{N} w_{v_i}(v_{i,t+1} - v_d^i)^2 \\ & + \sum_{t=0}^{T-1} \sum_{i=1}^{N} w_{u_i}(u_{i,t})^2 + \sum_{t=0}^{T-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} w_f \mathcal{F}_{i,j}^{t} \end{aligned} \tag{5}$$

Here, w_{v_i} ,w_{u_i} and w_f are weight coefficients and J is the problem objective to be minimised, subject to the given current states of the vehicles as defined in (1) and constraints as defined in (3) and (4). The speed and acceleration is bounded as described in section III-A.

C. Delay minimization algorithm

In this section, we give a high-level description of the AIM algorithm for intersection control proposed in [7]. The algorithm will in the rest of the paper be called the *Delay minimization algorithm*. In this algorithm, vehicles are assumed arrive at entrance zone with their maximum allowed speed. The ideal profile entails traveling the entire intersection zone without deceleration. This means that in the absence of obstacles, a vehicle should be able to cross the intersection at the same maximum speed. In order to avoid collisions, The algorithm adjusts the vehicles' speed, so that all vehicles can cross the intersection at their respective maximum movement speed without colliding with other vehicles.

Decisions of arrival times of each vehicle to critical zone are made by the optimization module. The objective of the algorithm is to find the optimal deceleration to minimize the total traveling time for all vehicles inside the intersection by considering the safety criteria. The minimum time for a vehicle to travel between the beginning of entrance zone and the beginning line of critical zone without deceleration is called the optimum time, denoted OT_i for vehicle i. The algorithm tries to minimise the extra delay (D_i) that is added to the optimum time in case of necessary deceleration.

In order to avoid any rear-ends collision, a minimum separation headway time of H_{min} between two vehicle on the same lane is defined.

$$|(OT_i + D_i) - (OT_j + D_j)| \geq H_{min} \tag{6}$$

To ensure that no collisions occur between a pair of vehicles form different approaching lanes inside the critical zone, the vehicles have to pass their possible conflict point with a minimum time separation $\Delta\tau$. Therefore, a linear inequality constraint is defined as in equation (7) below:

$$|(OT_i + D_i + \tau_{ij}) - (OT_j + D_j + \tau_{ji})| \geq \Delta\tau\delta_{i,j} \tag{7}$$

Here, $\delta_{i,j}$ is a binary variable that states whether the pair vehicles (i, j) have the potential to collide or not. τ_{ij} and τ_{ji} are defined as the travelling times from the conflict point of pair (i, j) to the beginning of the lane i and j respectively.

The system cost, J is defined as the sum of the required delay for all vehicles inside the intersection zone to avoid collision. J is given by equation (8).

$$J = \sum_{i=1}^{N1} D_i \tag{8}$$

Here $\mathcal{N}_1 = \{1, 2, \dots, N1\}$ is the set of vehicles that approaches the intersection zone in the current time step.

Also, $\mathcal{N}_0 = \{1, 2, \dots, N0\}$ is the set of arriving vehicles at the intersection during the past time steps that are still in the entrance zone. The total number of vehicles inside the entrance zone is $N = N1+N0$, since only vehicles in $\mathcal{N}_1$ are optimised in the current time step, The reserved time for each conflict point for vehicles in $\mathcal{N}_0$ from the previous time step is used as a new constraint for the following time step. This assumption will add the following constraint to the problem.

$$\begin{aligned} OT_i+D_i + \tau_{mn} \geq \\ max[(OT_j + D_j + \tau_{mn}), (OT_k + D_k + \tau_{mn}))] \end{aligned} \tag{9}$$

For all $i \in \mathcal{N}_1$ and $j, k \in \mathcal{N}_0$, τ_{mn} is an arbitrary conflict point. The algorithm defines an optimization problem to minimize the system cost, J at each time step by considering the current states of the vehicle as described in equation (1) and constraints as defined in equations (6) and (7).

IV. Evaluation

In this section, we describe our simulation environment and experiments.

A. Evaluation Environment

To evaluate and compare control methods, a realistic simulation program based on Simulation of Urban Mobility (SUMO) [8] has been developed. SUMO is an open source, highly portable, microscopic and continuous traffic simulation that gives the user control over all aspects of the network, such as vehicle type, driver behaviour, intersection control, and statistical data collection. In our work, we have modified SUMO by allowing each vehicle's speed to be manipulated by a central controller (i.e. the ICU) instead of using their default microscopic flow algorithms in SUMO.

B. Experiment

In this work, we consider a four-way intersection with two lanes in each way. Each lane is 3.5m wide with a maximum speed limit of 20m/s, i.e about 70 km/h. We assume that the intersection area can be modelled as a circle with radius 150m. TableI summarize the simulation parameters and specifications that we used for each algorithm.

The two intersection control algorithms were evaluated for different traffic flow rates. Traffic flow rate is defined as the rate at which vehicles pass a given point on the roadway, and it is normally given in terms of vehicles per hour. Based on the collected data from drivers using a navigation service in China [13] and , the the deployed intersections for a transportation project Michigan [14], we can divide the flow rate range in three different volumes. The peak hour flow rate for a typical

TABLE I
SIMULATION PARAMETERS

MPC	
v_d	16 m/s
u_{max}	5 m/s^2
u_{min}	-6 m/s^2
v_{max}	23 m/s
v_{min}	3 m/s
T	12 s
R_{min}	7 m
d_{min}	7 m

Delay minimization	
V_{max}	16 m/s
u_{max}	0 m/s^2
u_{min}	-6 m/s^2
H_{min}	2 s
$\Delta\tau$	4 s

intersection in an urban area is between 450 vehicles/hour and 600 vehicles/hour, and we define this as a *high volume traffic*. Further, a traffic rate between 150 vehicles/hour and 450 vehicles/hour is defined as a *medium volume traffic* in an urban area. Finally, a traffic rate of less than 150 vehicles/hour is defined as a *low volume traffic* in an urban area.

The maximum possible rate of vehicles crossing the counter point, that is the maximum capacity of the intersection, is defined as the *saturation flow rate*. In saturated intersection all vehicles move one after each other with minimum safe distance. In this paper, we assume that each vehicle has an average length of 4 meters. A safe gap distance of 2.5 meters is required between a pair of vehicles. Therefore, the maximum possible number of vehicles inside the intersection area is 92 vehicles that will be reached at saturation flow rate.

In this paper, we use the following performance metrics when we evaluate the two intersection control algorithms:

- Average speed of vehicles inside the intersection zone.
- Average number of vehicles inside the intersection zone.
- Collision probability.
- Execution time.

Also, all results will be compared with a standard signalised (traffic light) with 90 second green phase and 90 second red phase intersection management method.

The first two performance metrics are the most used metrics in the literature when evaluating the performance of an intersection control algorithm. Therefore, when considering the average speed and the average number of vehicles in the intersection, we expect both algorithms to improve the performance of the system when compared with the signalised method.

However, the objective of our work is to find intersection control algorithms that can be implemented in real world systems. Therefore, the third performance metric, collision probability, will be crucial, since this is the metric that checks the safety condition.

In addition, we expect that the number of objective function calculations and non-linear constraints for the MPC is higher than for the Delay minimisation algorithm, which can result in a problem for real applications due to the required processing times. Therefore, we decided to also show the resulting execution times for the algorithms as a comparison metric.

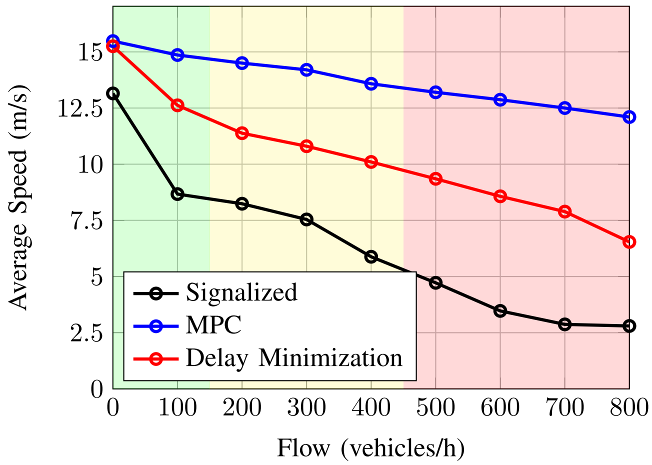

Fig. 2. Average speeds for different flow rates. Flow rates are divided in three different volumes.

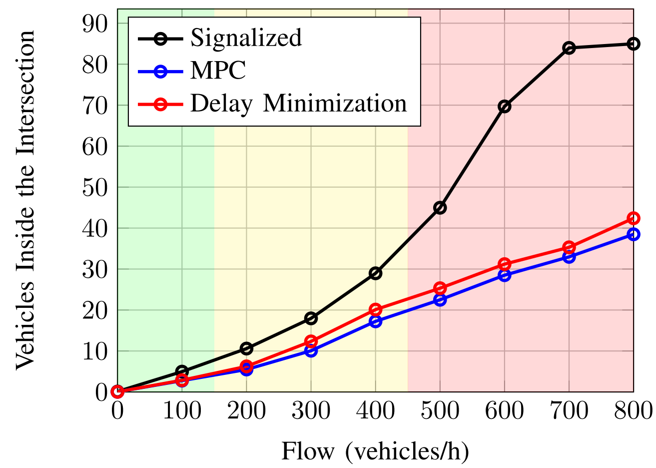

Fig. 3. Average number of vehicles inside the intersection zone (a circle with radius 150m).

V. Results and Discussion

We performed simulations to compare the two algorithms and evaluate the feasibility to deploy them in reality.

A. Average speeds and average number of vehicles

Fig. 2 illustrates the average speed of each vehicle in one time slot for different flow rates. In the MPC algorithm, the acceleration can have either a positive or negative value. In the Delay minimization approach, the acceleration is limited to negative values, since they are always related to the maximum speed. Therefore, as expected, the system average speed for the MPC algorithm is much higher compared to the Delay minimization algorithm and the traditional signalized method.

Fig. 3 shows the average number of vehicles in the intersection area for different flow rates. The signalized intersection will be saturated at a flow rate of about 700 vehicles/hour per lane, which sets an upper limit of the capacity. It is apparent from Fig. 3 that for both the MPC algorithm and and the Delay minimization algorithm, there are much less vehicles inside the intersection compared to the signalized intersection. This shows that both algorithms have the potential to increase the capacity of the intersection. However, the simulations showed that when the number of vehicles in the entrance zone increases to more than 20 vehicles at each time step, the solver may not find a feasible solution to control the vehicles' speeds in proper time. Therefore, we can not reach the maximum claimed capacity of 1600veh/h in [6].

B. Collision probabilities

Our simulation results validated the performance of the two employed methods with respect to traditional signalised approach as reported in [6] [7]. The intersection capacity and traveling times (vehicle speeds) are improved with both the MPC algorithm and the Delay minimisation algorithm.

However, in order to deploy the two proposed methods in the real world, the traffic safety must be evaluated as well, since this will be crucial for operational systems.

Fig. 4 shows the expected collision probabilities for different traffic flows. A signalized intersection is assumed to have zero probability of collisions for all flows, since this is the main reason for deploying traffic lights in intersections.

In the Delay minimization algorithm, the controller avoids collision between two vehicles by reserving two different time slot for crossing the intersection. However, even if this control strategy avoids collision inside the critical zone, it can not guarantee that no collisions occur in the entrance zone.

Fig. 4 shows that there is a probability of 0.2% of collisions for a flow rate of 600 vehicles/hour. This means that 1.2 collisions per hour can be expected when using the Delay minimization algorithm, which of course is not an acceptable traffic safety condition.

On the other hand, the MPC algorithm prevents collisions in the whole intersection area over the given time horizon (12 sec). However, a problem occurs when the arriving flow rate increases to more than 500 vehicles/hour. At this flow rate, it can be expected that new vehicles will enter the intersection zone during the problem prediction time horizon. In the algorithm, the number of vehicles inside the entrance zone is assumed to be constant during the horizon time, and therefore, this dynamic may cause collision in intersection.

Fig. 4 shows that the collision probability increases exponentially with the flow rate for the MPC algorithm, which of course is not an acceptable traffic safety condition.

C. Execution times

Another requirement for operational systems will be that an intersection control algorithm must have a low execution time in order to fulfill the extreme real-time properties required for these types of systems.

Fig. 5 shows the execution times for the two algorithms with 95% confidence intervals. We used a desktop equipped with an Intel Core i7-4790K CPU @4 GHz and DDR3 RAM @1600 MT/s. The computer was configured with an Ubuntu Linux. We set the simulation step time in SUMO to 0.5 seconds.

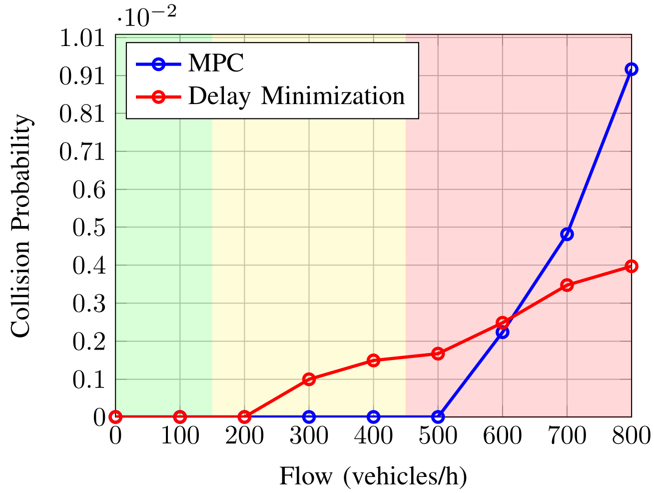

Fig. 4. Simulation results for collision probability

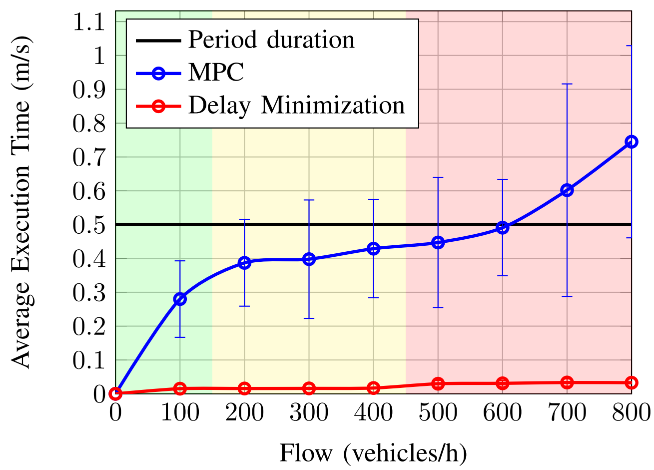

Fig. 5. Execution time for proposed algorithms vs. traffic flow rate

The MPC algorithm is a nonlinear optimisation approaches and it needs to predict the system state in the given prediction time. We expected a higher execution time for the MPC algorithm compared with the Delay minimization algorithm. As can be seen in Fig. 5, this difference is definitely noticeable and for higher flow rates the execution time for the MPC algorithm is longer than the simulation step time, which would mean that the calculations for one time step will not be completed before the next time step begins. Therefore, the controller may not be able to make a control decision during a time step. When the number of vehicles and consequently the size of the problem increases, the decision making process time increases dramatically, and it will require more powerful computers to find the solution in proper time.

VI. Conclusion

The objective of this paper is to evaluate two previously proposed algorithms for an autonomous intersection management [6] [7] in a realistic simulation environment, with the ultimate goal to develop control algorithms for autonomous vehicles that can be deployed in operational systems. It is observed that using these schemes improve the performance of the traditional signalized intersection. However, our simulation shows that the safety conditions are not satisfied in high traffic densities and only can be used for low traffic rate. since the collision probability rather quickly becomes larger than zero when the traffic increases. Also, the execution times of the algorithms in [6] makes the algorithm infeasible for realistic traffic scenarios.

VII. Future Work

The design of real time intersection management systems is a complex task that involves many different steps. Full understanding of all different parts of the design procedure require deep knowledge of theory. This paper has just briefly described the principles of different traffic management methods. In the future, the goal is to design an efficient and scalable control method for managing vehicles at intersections.

References

[1] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[2] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, 2014.

[3] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *2015 American Control Conference (ACC)*, pp. 763–768, IEEE, 2015.

[4] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *2015 54th IEEE conference on decision and control (CDC)*, pp. 4708–4713, IEEE, 2015.

[5] E. Steinmetz, R. Hult, Z. Zou, R. Emardson, F. Brännström, P. Falcone, and H. Wymeersch, "Collision-aware communication for intersection management of automated vehicles," *IEEE access*, vol. 6, pp. 77359–77371, 2018.

[6] M. A. S. Kamal, J.-i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1136–1147, 2014.

[7] I. H. Zohdy and H. A. Rakha, "Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept," *Journal of Intelligent Transportation Systems*, vol. 20, no. 1, pp. 17–32, 2016.

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

[9] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE communications magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[10] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2510–2525, 2018.

[11] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[12] L. Grüne and J. Pannek, "Nonlinear model predictive control," in *Nonlinear Model Predictive Control*, pp. 45–69, Springer, 2017.

[13] J. Zheng and H. X. Liu, "Estimating traffic volumes for signalized intersections using connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 347–362, 2017.

[14] K. Gay, V. Kniss, *et al.*, "Safety pilot model deployment: lessons learned and recommendations for future connected vehicle activities.," tech. rep., United States. Department of Transportation. Intelligent Transportation, 2015.