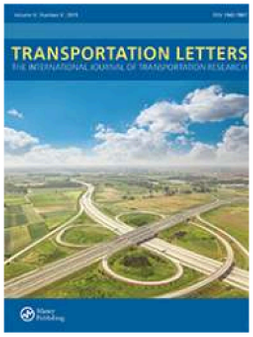

Taylor & Francis
Taylor & Francis Group

Transportation Letters

The International Journal of Transportation Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ytrl20

Leveraging autonomous vehicles in mixed-autonomy traffic networks with reinforcement learning-controlled intersections

Sahand Mosharafian, Shirin Afzali & Javad Mohammadpour Velni

To cite this article: Sahand Mosharafian, Shirin Afzali & Javad Mohammadpour Velni (2022): Leveraging autonomous vehicles in mixed-autonomy traffic networks with reinforcement learning-controlled intersections, Transportation Letters, DOI: 10.1080/19427867.2022.2146302

To link to this article: https://doi.org/10.1080/19427867.2022.2146302

TRANSPORTATION LETTERS
https://doi.org/10.1080/19427867.2022.2146302

Taylor & Francis
Taylor & Francis Group

Leveraging autonomous vehicles in mixed-autonomy traffic networks with reinforcement learning-controlled intersections

Sahand Mosharafian, Shirin Afzali and Javad Mohammadpour Velni

School of Electrical & Computer Engineering, University of Georgia, Athens, GA, USA

ABSTRACT

Development of new approaches to adaptive traffic signal control has received significant attention; an example is the reinforcement learning (RL), where training and implementation of an RL agent can allow adaptive signal control in real time, considering the agent's past experiences. Furthermore, autonomous vehicle (AV) technology has shown promise to enhancing the traffic mobility at highways and intersections. In this paper, delayed action deep Q-learning is developed for a vehicle network with signalized intersections to control the signal phase. A model predictive control (MPC) scheme is proposed to allow AVs to adapt their speed. Several case studies that consider mixed autonomy are examined aiming at reducing network traffic and fuel consumption in the traffic network with multiple intersections. Simulation studies reveal that even with a few AVs in the network, the waiting time, fuel consumption, and the number of stop-and-go movements are significantly reduced, while the travel time is increased.

KEYWORDS

Reinforcement learning; autonomous vehicles; speed control; mixed autonomy

Introduction

Traffic congestion has become a major issue due to the population growth and increasing demand for using vehicles. Until recently, the main attempts had been toward improving physical infrastructure to reduce traffic although it would not be a cost-effective solution. Therefore, an alternative solution has emerged aiming to improve current infrastructure by developing new traffic control methods and optimizing traffic flow. This solution would be particularly appealing for locations such as downtown areas that do not have the capability of being physically upgraded due to surrounding buildings or that they are already upgraded to their limit (Chowdhury and Sadek 2003; El-Tantawy, Abdulhai, and Abdelgawad 2014).

One source of traffic congestion is intersections with traffic signals which can reduce traffic flow and increase waiting time of the vehicles. Vehicle idling (transition between red light and green light, in which the vehicle needs to wait) increases delays in trips and reduces vehicle fuel efficiency or miles per gallon. Some timing methods have been applied to alleviate the traffic in intersections. However, they are not adapted to the dynamic traffic system which changes during the day. Two traditional traffic signal controllers are pre-timed and actuated. Pre-timed method uses historical traffic data to adjust green signal timing in different times of the day. This method does not consider any information about traffic dynamics and changes. Actuated control method takes current traffic situation at an intersection to turn green light on for lanes with more cars without using any information about traffic in long term (Yau et al. 2017). Hence, interactive solutions for traffic signal control would improve the performance of current infrastructures by taking actual traffic information into account. Such solutions are referred to as adaptive traffic signal control (Mannion, Duggan, and Howley 2016).

Reinforcement learning (RL) consists of an agent which interacts with the environment and receives reward based on the action it takes at different states. The goal of the agent is to maximize the discounted future rewards. Consequently, the agent can learn and improve its policy (control action) by receiving reward based on the action. The distinguishing feature of RL is 'learning by interaction with the environment' (Sutton and Barto 2018). RL has been applied to different problems in the context of intelligent transportation (Mannion, Duggan, and Howley 2016; Qu et al. 2020). RL provides real-time solutions for traffic signal problem by considering it as a model-free system. Various learning methods such as Q-learning (Watkins and Dayan 1992) have been widely studied for improving the traffic congestion by reducing traveling time or waiting time, increasing traffic flow, and so on. RL methods can change the control signal phase duration in addition to their sequences to handle the traffic in an intersection or a network of intersections (Yau et al. 2017).

Various traffic signal control problems using RL have been studied. Some researchers considered isolated intersection case (Genders and Razavi 2016; Zhang et al. 2018), while others considered a network of intersections (Abdoos, Mozayani, and Bazzan 2011; Aziz, Zhu, and Ukkusuri 2018). Furthermore, various algorithms and tools have been applied to the traffic signal control problem, e.g., tabular methods (El-Tantawy, Abdulhai, and Abdelgawad 2014; Steingrover et al. 2005) and neural networks (Ge et al. 2019; Genders and Razavi 2016; Zhang et al. 2018). Speed trajectory control has also been studied in various scenarios. While some studies have considered improving speed control in an intersection without traffic signal (Levin and Boyles 2016; Li et al. 2019; Lin et al. 2017), in this research, we focus on the literature on speed control in a signalized intersection.

Yang, Guler, and Menendez (2016) examined the impact of conventional vehicles (i.e. with no vehicle-to-infrastructure (V2I) communication), connected, and autonomous vehicles (AVs) (both equipped with V2I communication systems) on the delay of an intersection with actuated signal control with the goal of optimizing departure sequence

CONTACT Javad Mohammadpour Velni javadm@uga.edu School of Electrical & Computer Engineering, University of Georgia, Athens, GA 30602, USA

of vehicles and AV trajectories. Simulation results showed improvement in delay. Rakha and Kamalanathsharma (2011) made an attempt to optimize fuel consumption of vehicles using V2I information exchange; in the latter study, the vehicles considered were either autonomous and their speed could be controlled, or the drivers could receive advisory speed based on future traffic signal changes and the size of the queue.

Li et al. (2018) considered cycle-based signal control together with electric vehicle (EV) system with eco-driving. Xu et al. (2017) considered cooperation between vehicles and traffic signal controller to minimize trip time and energy consumption. First, the signal timings were calculated, and then, vehicles (all assumed to be AVs) adjusted their speed based on the signal information. The traffic signal was controlled using dual-ring phase control in He, Head, and Ding (2011). In this actuated phase control method, the timings for the whole cycle were calculated based on current traffic information (short-term). More recent research studies such as Tajalli, Mehrabipour, and Hajbabaie (2020) and Zhao, Liu, and Ngoduy (2019) also considered the same problem, but they assumed that all vehicles are autonomous. Additionally, to the best of our knowledge, *no past work has considered the aforementioned problem, in which traffic signal is controlled using RL methods*. RL methods have shown the capability of controlling traffic in real time and adapting to traffic changes with no need to take vehicle dynamic model into account.

In a fully automated traffic network where all the vehicles are connected and automated, traffic lights are no longer needed since vehicles are able to plan safe and efficient trajectories by leveraging wireless communication (Chavoshi, Genser, and Kouvelas 2021). However, considering a fully automated traffic is an optimistic assumption, and our focus in this paper is to improve the traffic networks including both human-driven vehicles and AVs. In general, scenarios that involve a mixture of AVs and human-driven vehicles (mixed autonomy) bring challenges including controllability issue and partial observability (Wu et al. 2017). Here, we intend to study the impact of AVs in mixed autonomy. In this paper, *our goal is not only to adapt traffic signal phase to upcoming traffic using RL (real-time control) but also to adjust AVs speed to the current traffic signal phase in order to reduce the waiting time of the vehicles in a signalized intersection*. The novelty of our work lies in employing an RL-based traffic signal control and online AV speed adjustment together, while we also indirectly consider the queue in front of the AV before the traffic light. In the literature, vehicle speed control has been examined under the assumption that the length of the traffic phases is known (by considering short-term information only); however, in our work, we study the case where only the minimum duration of a traffic phase is known, but the duration of each phase depends on the traffic and the RL agent. To this end, we investigate the impact of different AV penetration rates on the waiting time and fuel consumption in a network of RL-controlled intersections.

In AV speed control problem, kinematic/dynamic system equations, as well as physical constraints such as actuator saturation limits and speed limit, should be taken into account. These limitations add constraints to the problem of optimizing the AV performance. Model predictive control (MPC) allows optimizing multiple performance indices (e.g., fuel consumption, comfort, etc.) under AV system constraints. In MPC, the future control sequence for a chosen horizon is calculated with the goal of optimizing a cost function (Raffo et al., 2009). In our work, MPC optimizes each AV speed trajectory to reduce waiting time and fuel consumption in a traffic network.

The contributions of this paper are as follows: (1) a real-time speed control method in an intersection controlled by an RL agent is proposed; (2) an MPC-based algorithm for the speed control in an RL-controlled intersection is proposed where the exact phase timings are not available to vehicles a priori, assuming that AVs have access to the current phase and the remaining time until the RL controller makes decision; (3) the impact of the queue in front of the AV is considered in the proposed speed adjustment algorithm; (4) the mixed autonomy under different AV penetration rates is studied and the network waiting time and fuel consumption are evaluated, demonstrating the success and efficacy of the proposed coordinated control method. *It is noted that the proposed approach for the traffic control in mixed autonomy environment is* ***cooperative and hybrid*** *in that it involves both data-driven and model-based decision making for signal control and vehicle speed adjustment, respectively*.

The rest of the paper is organized as follows. Section 2 provides problem statement and AV dynamics and describes our traffic signal control algorithm using deep Q-learning. The proposed speed adjustment method is described in Section 3. Simulation results and discussions are provided in Section 4, and finally, concluding remarks are made in Section 5.

Problem statement and preliminaries

Problem of interest: The main objective of this work is to examine the impact of AVs and their speed adjustment based on traffic signal phase when the AVs are in vicinity of an intersection. The paper examines two interrelated issues as follows: the traffic signal should be controlled in a way to reduce the vehicles delay (waiting time), and also, AV speed should be optimized aiming to reduce the waiting time or even preventing the AVs to stop at intersections. Our goal is to show that even with a very small number of AVs in a mixed fleet of vehicles, the waiting time of the vehicles in the network would be positively affected. To demonstrate this, we study multiple scenarios with different AV penetration rates. In all the scenarios, a four-intersection network is considered where each intersection is controlled independently using RL. The impact of AV speed adjustment on network waiting time and fuel consumption with different AV penetration rates in a mixed autonomy is also examined.

Description of Q-learning and its use for traffic signal control

The proposed approach of this paper employs an RL method (namely, deep Q-learning) to first train agents (in this case traffic signals) and then make appropriate decisions based on the circumstances. Q-learning is a temporal difference (TD) control algorithm (Watkins and Dayan 1992). TD methods do not generally require a model of the environment; instead, they learn by experiencing the consequences of actions they provide. Similar to dynamic programming (DP), TD algorithms can update estimates without waiting for a final outcome (Sutton and Barto 2018). One-step Q-learning update formula is as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right], \tag{1}$$

where A_t is the set of actions at time instant t; S_t is the set of states at t; R_t is the set of rewards at t; $Q(s, a)$ is an estimate of the value of taking action a in state s; α is the learning rate; and γ is the discount-rate parameter for the reward value (Sutton and Barto 2018). In deep Q-learning (Mnih et al. 2015), Q-value is estimated using neural network, where two networks are used: Q-network and target network. Target network weights can be updated with Q-network

parameters after every fixed step, or the weights can be updated using soft target updates. The loss function for the Q-network is

$$L = \mathbb{E}_{(s,a,r,s')}[(r + \gamma \max_{a'} Q(s', a'; \theta^t) - Q(s, a; \theta))^2] \quad (2)$$

where θ are the parameters of Q-network, while θ^t are the target network parameters. To calculate the loss, observations are stored in a replay buffer, and loss is calculated by randomly choosing samples from the replay buffer (called mini-batch). Algorithm 1 shows a deep Q-learning algorithm used for the non-episodic task in our work. One of the strategies for updating $Q(S, A)$ is ε-greedy selection, in which the agent exploits (takes the action that maximizes the Q-value) with the probability of $1 - \varepsilon$ and explores (selects a random action) with the probability of ε. Exploitation results in the maximum expected reward in one step, whereas exploration may result in a greater Q-value in the long term (Sutton and Barto 2018).

Algorithm 1 Deep Q-learning algorithm

1: Initialize neural network parameters θ, empty replay buffer B, S_0, learning rate α, discount rate γ, and soft update parameter τ.
2: **repeat**
3: **for** each step **do**
4: In current state S, take action A (e.g., using ε-greedy) and observe R, S'
5: Store transition (S, A, R, S') in replay buffer B
6: Sample a random mini-batch from B
7: Calculate loss using the mini-batch data based on (2) and update Q-network
8: Update target network using soft target update $\theta^t \leftarrow \tau\theta + (1 - \tau)\theta^t$
9: $S \leftarrow S'$
10: **end for**
11: **until** forever

In our traffic signal control problem, actions are traffic signal phases. We consider two actions: (1) east-west and west-east green and (2) north-south and south-north green. Before any transition from one action to another, there is a yellow traffic light phase. RL agent calculates the action based on ε-greedy action selection. As shown in Figure 1, each entering lane within a specific distance d_i from the intersection is divided into some blocks with the length l_b. If a vehicle exists in the block, corresponding value for the block is the normalized vehicle velocity; otherwise, the value is −1. It is worth mentioning that the smaller the size of the block is, the larger the state space is. Reward, in our problem, is the negative of the waiting time per vehicle (in seconds) for vehicles in entering lanes in a vicinity of the junction (250 m radius is considered in this paper). In our study, the RL agent decision step is $T_{RL} = 10$ s, and it updates its decision every T_{RL} seconds. Thus, each action would remain in effect for at least T_{RL} seconds.

Remark 1. It is noted that in our study, the RL agent does not apply its decision to the traffic signal immediately. Instead, every time RL agent gathers information about current state, and it chooses a new action but applies it to the intersection in the next decision-making event (which is in T_{RL} seconds). Simulation experiments presented in Section 4 show that applying RL agent's decision with delay does not deteriorate the performance of the RL algorithm and even improves the performance (in terms of waiting time and fuel consumption).

Autonomous vehicle system dynamics

For each AV, a linear state-space representation is considered as follows:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad (3)$$

where $x(t)$ and $v(t)$ are AV position and velocity, respectively, and $u(t)$ is the vehicle acceleration (also the control input). By discretizing the given state-space equation with sampling time t_s, vehicle dynamics in discrete time are represented as follows:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ t_s \end{bmatrix} u(k)\,. \quad (4)$$

Furthermore, following constraints are imposed on the system states and control input:

$$0 \le v(k) \le v_{max}, \quad (5a)$$

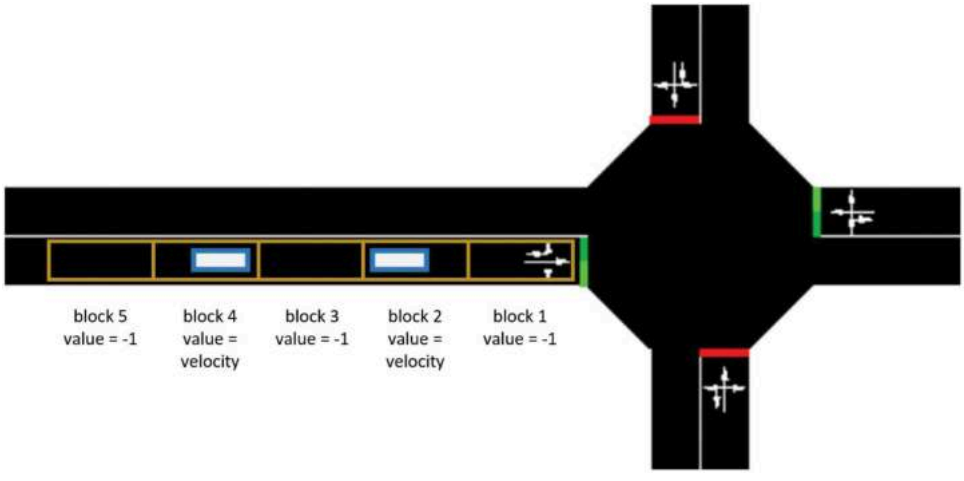

Figure 1. A simple demonstration of state definition in a junction. In this example, an entering lane is divided into five blocks. The corresponding value for blocks with vehicles (shown by white boxes) is the normalized velocity of the vehicle, while the value for the other blocks is −1. An array including the block values for all entering lanes build the state of the junction.

$$u_{min} \leq u(k) \leq u_{max}, \tag{5b}$$

where v_{max} represents road speed limit, and u_{min} and u_{max} are minimum and maximum possible values for system input, respectively.

Proposed approach to AV speed control

When an AV distance from the intersection goes below some predefined value (the value depends on the road speed limit and the minimum green signal period; here it is considered 250 m), the intersection sends the current traffic phase information to the AV through infrastructure-to-vehicle (I2V) communication. Then, the AV (we call it adaptive AV) adjusts its speed based on that information and the information it receives from onboard sensors about its leader vehicle, i.e., the closest vehicle in front of it (if any). If there is a vehicle between the adaptive AV and the intersection, the closest human-driven (non-AV) one to the AV is called the leader vehicle, and the closest AV is called the leader AV. As shown in Figure 2, depending on the existence of a leader vehicle/AV, an adaptive AV experiences one of the following cases:

(1) (case 1) There is no leader vehicle, in which case, the adaptive AV adjusts its speed based on its distance from the intersection, current traffic phase, and the time remaining until the current RL agent makes an action.
(2) (case 2) There is a human-driven leader vehicle, in which case, first the adaptive AV determines if the leader will pass the intersection with its current speed and current signal phase information. If yes, then this case is similar to case 1; otherwise, the AV should use the speed and position of the other vehicle for calculating the possible distance it can travel.
(3) (case 3) There is a leader AV, in which case, first the adaptive AV determines if the leader AV will pass the intersection based on the current leader speed, distance from the intersection, and current traffic signal information. If yes, this case is similar to either case 1 or case 2; otherwise, the adaptive AV does not need to make any calculations and follows the traffic in front of it because the leader AV will adjust its speed.

To check if a leader vehicle can pass the intersection when the traffic light is green, the following inequality conditions can be easily checked (assuming that the leader vehicle moves at a fixed velocity for the next T seconds):

$$V_L^H.T_G > d_L^H \tag{6a}$$

$$V_L^{AV}.T_G > d_L^{AV}, \tag{6b}$$

where T_G is the expected remaining time of the green phase; V_L^H and d_L^H are the leader vehicle's speed and distance from the intersection, respectively; and V_L^{AV} and d_L^{AV} are the leader AV's speed and distance from the intersection, respectively. If (6a) holds, the human-driven leader vehicle will pass the intersection; if (6b) holds, the leader AV will pass the intersection. According to remark 1, when current traffic phase is green, T_G can be calculated as follows:

$$T_G = T + stat.T_{RL}, \tag{7}$$

where T is the remaining time until the traffic controller updates its decision, and *stat* is a binary variable which is one if the current and next phase are the same otherwise zero. It is worth mentioning that according to Remark 1, the current and next traffic phases are available, and the intersection shares both with AVs through V2I communication.

If no leader exists, or the leaders pass the intersection based on (6), then the adaptive AV needs to evaluate whether it can pass the intersection. The AV determines if it can pass the intersection by solving the following MPC problem:

$$\begin{aligned} &\min_{u(k)} \sum_{k=0}^{N_1} [u(k)]^2 + [v(k) - v_{max}]^2 \\ &\text{subject to : system equations } (4), \\ &\text{system constraints } (5), \\ &x(N_1) \geq d_{AV}, \end{aligned} \tag{8}$$

where $N_1 = T_G/t_s$, t_s is the sampling time, and d_{AV} is the adaptive AV's distance from the intersection. Incorporating the vehicle input in the cost function prevents the vehicle from unnecessary acceleration/deceleration, thereby reducing fuel consumption while maximizing vehicle speed improves traffic flow. The last inequality forces the MPC to find a solution (if exists) that guarantees passing the intersection within the prediction horizon N. Consequently, if

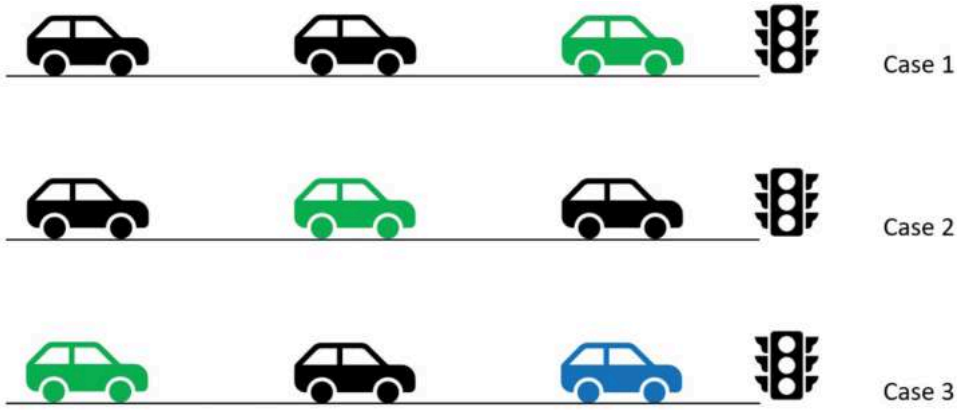

Figure 2. Different cases for an adaptive AV approaching the intersection. The green vehicle is the adaptive AV, the blue one is other AV, and the black one is a human-driven vehicle (non-AV).

Table 1. List of parameters used in the simulation study.

$\tau = 0.0001$	$\overline{V} = 1\ m/s$	$l = 6.5\ m$	$T_{RL} = 10\ s$
$a = 0.00005$	$\underline{V} = 0.1\ m/s$	$l_b = 6.5\ m$	$T_y = 2\ s$
$\gamma = 0.8$	$v_{max} = 15.65\ m/s$	$d_i = 260\ m$	$t_s = 1\ s$

the above optimization problem is feasible, then adaptive AV can pass the intersection. Otherwise, the AV cannot pass during the next T_G seconds; hence, the AV should adjust its speed to avoid unnecessary acceleration/deceleration until it can pass the intersection. Based on the delayed RL agent action execution (see Remark 1), we define T_d as the expected time that the AV should wait until the traffic phase becomes green again, which is calculated as follows:

$$T_d = \begin{cases} T + 2\,T_y + T_{RL}, & \text{if phase is green} \\ T_G + T_y, & \text{if phase is red or yellow} \end{cases} \tag{9}$$

where T_y is the yellow signal duration (in this paper, $T_y = 2s$). The adaptive AV speed trajectory is then calculated by solving the following optimization problem:

$$\begin{gathered} \min_{u(k)} \sum_{k=0}^{N_2} [u(k)]^2 \\ \text{subject to : system equations (4),} \\ \text{system constraints (5),} \\ x(N_2) \le d_{AV}, \\ v(N_2) \le \overline{V}, \\ v(k) \ge \underline{V}, \end{gathered} \tag{10}$$

where $N_2 = T_d/t_s$, and $\underline{V} > 0$ and $\overline{V}$ is a lower bound and higher bound on the AV final speed, respectively. The third constraint guarantees that the AV will not pass the intersection at $k = N_2$. The fourth constraint is considered to avoid arriving at the intersection with high speed so that the AV would be able to stop in the case that traffic phase does not turn green after T_d seconds as expected. The last constraint prevents the adaptive AV from stopping completely within the prediction horizon.

In the case that there is a leader AV which cannot pass the intersection (either the phase is red or green), the adaptive AV does not need to make any calculation based on our earlier discussion for case 3. If the leader is a human-driven vehicle that cannot pass the intersection, then the adaptive AV should calculate the effective distance (its distance from the intersection while considering the position of the leader). The effective distance (d_{eff}) is defined as follows:

$$d_{eff} = d_{AV} - (d_L^H - x) - l \text{ where } x = \min(V_L^H.T_d, d_L^H), \tag{11}$$

where l is the average vehicle length plus the minimum gap between vehicles (here, l is assumed to be 6.5 m). After calculating d_{eff}, AV calculates its speed using (10) by replacing d_{AV} with d_{eff}. Consequently, every AV within a specified perimeter of the intersection changes its speed based on Algorithm 2. Each AV runs Algorithm 2 every second so that each adaptive AV updates its speed based on the most recent information.

Algorithm 2 Proposed AV speed control strategy

```
F_H ← 0
F_AV ← 0
if there is a human-driven (non-AV) leader vehicle then
    V_L^H ← leader vehicle speed
    d_L^H ← leader vehicle distance from the intersection
    if current phase is green and (6a) does not hold then
        F_H ← 1
    else if current phase is not green then
        F_H ← 2
    end if
end if
if there is a leader AV then
    V_L^AV ← leader AV speed
    d_L^AV ← leader AV distance from the intersection
    if current phase is green and (6b) does not hold then
        F_AV ← 1
    else if current phase is not green then
        F_AV ← 2
    end if
end if
if F_AV > 0 then
    follow the traffic
else
    if F_H == 0 then
        if signal is green and (8) is feasible then
            adjust speed based on (8)
        else
            adjust speed using (10)
        end if
    else
        d_AV ← calculate d_eff using (11)
        use (10) to adjust speed
    end if
end if
```

Simulation studies and discussion

Simulation of Urban Mobility (SUMO) (Lopez et al. 2018) is used as the traffic simulation environment. SUMO provides information about possible movements at intersections and right-of-way rules in details. To obtain the waiting time, fuel consumption, location, and speed of vehicles, Traffic Control Interface (TraCI) is used (Lopez et al. 2018), while the deep Q-learning controller is implemented using Keras library in Python (Chollet et al. 2015), and the MPC optimization problem is implemented and solved using CVXPY (Agrawal et al. 2018; Diamond and Boyd 2016). Parameters used in the simulation are listed in Table 1. For the Q-network and target network, neural networks with five hidden layers are used, each of which consists of 1024 units with rectified linear activation functions. The traffic network used in the simulation studies is shown in Figure 3. This network initializes without vehicles inside, and vehicles enter the network from

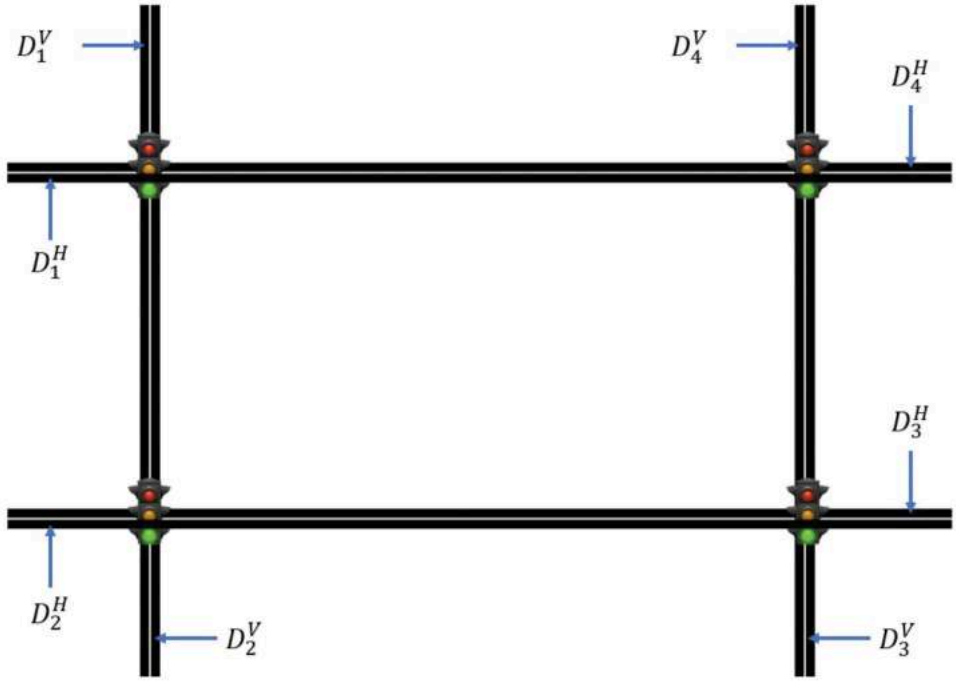

Figure 3. Network of four connected intersections used in our simulation studies. Each intersection is controlled by an independent RL agent. In each of the network's incoming lanes, a detector is installed, namely, D_1^H-D_4^H for horizontal lanes and D_1^V-D_4^V for vertical lanes. The detectors are later used for micro-simulation calibration.

eight entering lanes. Since the RL agent controls the traffic using $\varepsilon-$ greedy and may select actions randomly, and the vehicles are generated randomly in SUMO, the simulations are performed repeatedly and the average over those simulations is calculated. The rest of this section is divided into three parts; first, the simulation studies are done without micro-simulation calibration. Then, micro-simulation calibration is discussed, and finally, simulation studies are conducted using the calibrated environment.

Simulation studies without a micro-simulation calibration

In the first study, the performance of the deep Q-learning algorithm with and without action delay is investigated. The average traffic flow in the vertical lanes is assumed to be 540 veh/h, while in the horizontal lanes, the average flow is 360 veh/h. As shown in Figure 4, the two top subplots represent vehicle waiting time, while the bottom subplots depict the vehicle fuel consumption. Results show that applying the RL decision with delay in the next

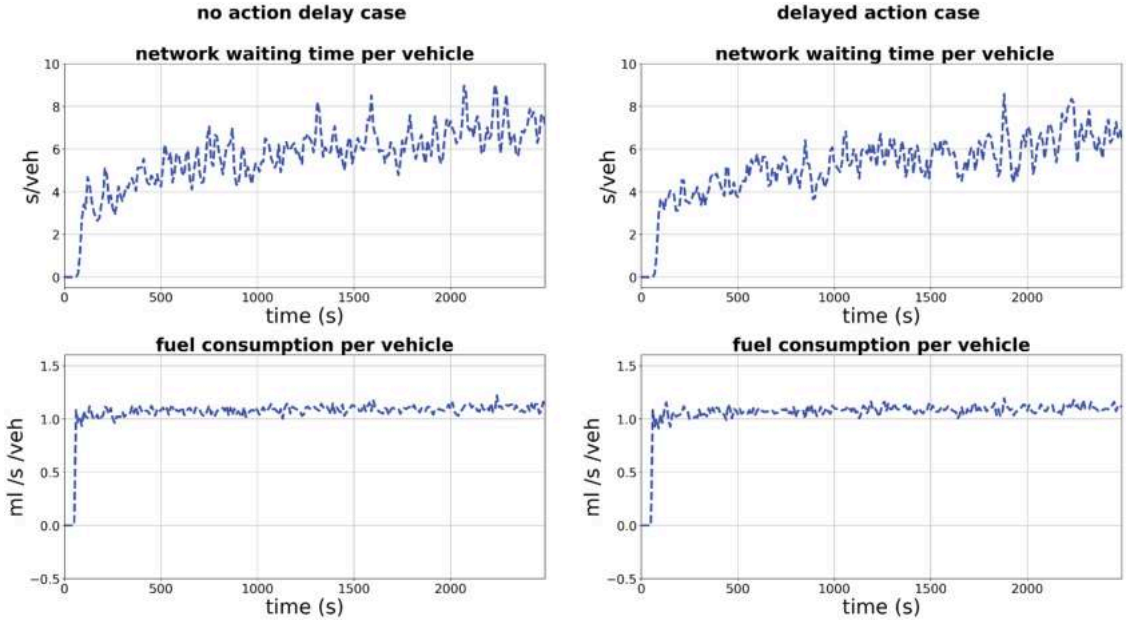

Figure 4. Performance of the RL agent in two different cases. In the first case, whose results are shown in the left two subplots, the agent applies its decision immediately; in the second case shown in the right two subplots, the action is applied in the next decision-making event. The delayed action case performance is slightly better than the case without action delay in terms of mean waiting time and mean fuel consumption.

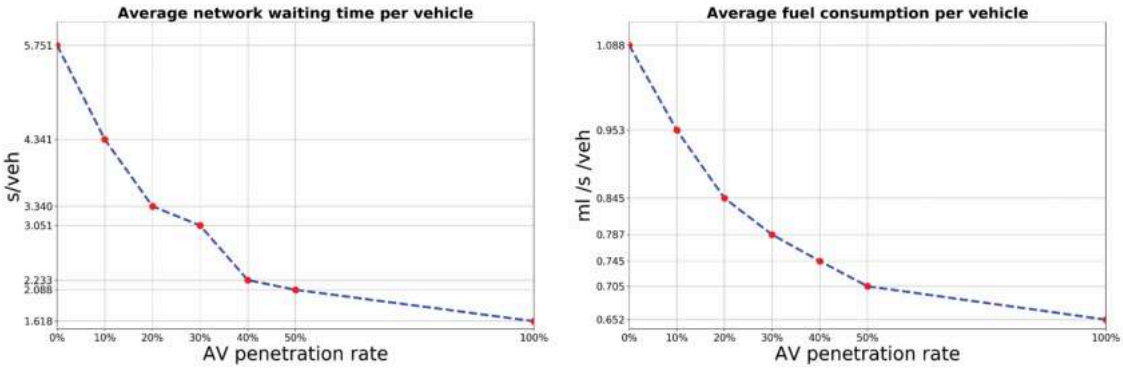

Figure 5. Performance of the proposed method for AV penetration rate of 10% in the left two subplots and 20% in the right two subplots.

decision-making step slightly improves the performance of the RL traffic controller in our study. The mean waiting time and fuel consumption per vehicle from time 500 s to 2500 s for no action delay case are 6.174 s, and 1.091 ml/s, respectively, while for the delayed action case, the mean waiting time is 5.751 s and the fuel consumption is 1.088 ml/s.

Remainder of the simulations investigate the impact of the proposed AV speed adjustment method in the mixed autonomy with different AV penetration rates. It is expected that increasing the AV penetration rate improves the traffic system through reducing waiting time and fuel consumption. Figures 5–7 depict the waiting time and fuel consumption with different AV penetration rates. It is noted that these plots show average results of several runs. The mean waiting time and fuel consumption per vehicle from time 500 s to 2500 s are calculated for each case, and the results are summarized in Figure 8. Comparing the case without AVs with those that include AVs in the network reveals the significantly high impact of our proposed MPC-based speed adjustment on improving the traffic flow. The results demonstrate that even a relatively low number of AVs in the mixed autonomy improves the traffic flow. The case study considering 10% penetration rate reveals that the mean waiting time per vehicle is reduced by 24% and the fuel consumption per vehicle is reduced by 12%. This improvement occurs since AVs are able to shape the behavior of the vehicles following them and hence traffic. When penetration rate is increased to 50%, waiting time and fuel consumption are reduced by 63% and 35%, respectively. In the case that all vehicles in the network are autonomous, waiting time is reduced by 72% and fuel consumption is reduced by 40%. Although a network including only AVs is the ideal case because of the lowest waiting time and fuel consumption, it may not be feasible to achieve such a network. However, even a small number of AVs show promising impact on improving the traffic system. It is noted that as the penetration rate increases, AVs, and hence human-driven vehicles following them, have a high chance of passing the intersection without going to a stop behind the traffic light. According to Figure 7, in the ideal case, vehicle's waiting time gets very close to zero at some intervals.

Next, two sets of simulations are conducted considering two different traffic flows to evaluate the impact of the traffic flow on the performance of the proposed approach. In the first case, the average traffic flow is 50% of the flow used in the previous simulations (results are shown in Figure 9), while the average

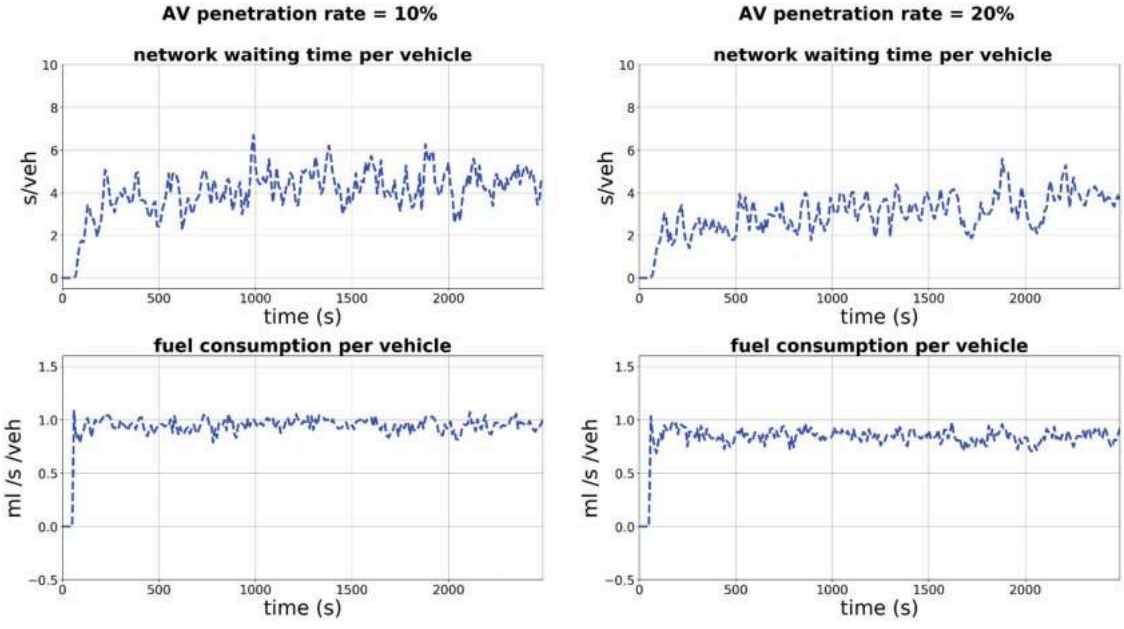

Figure 6. Performance of the proposed method for AV penetration rate of 30% in the left two subplots and 40% in the right two subplots.

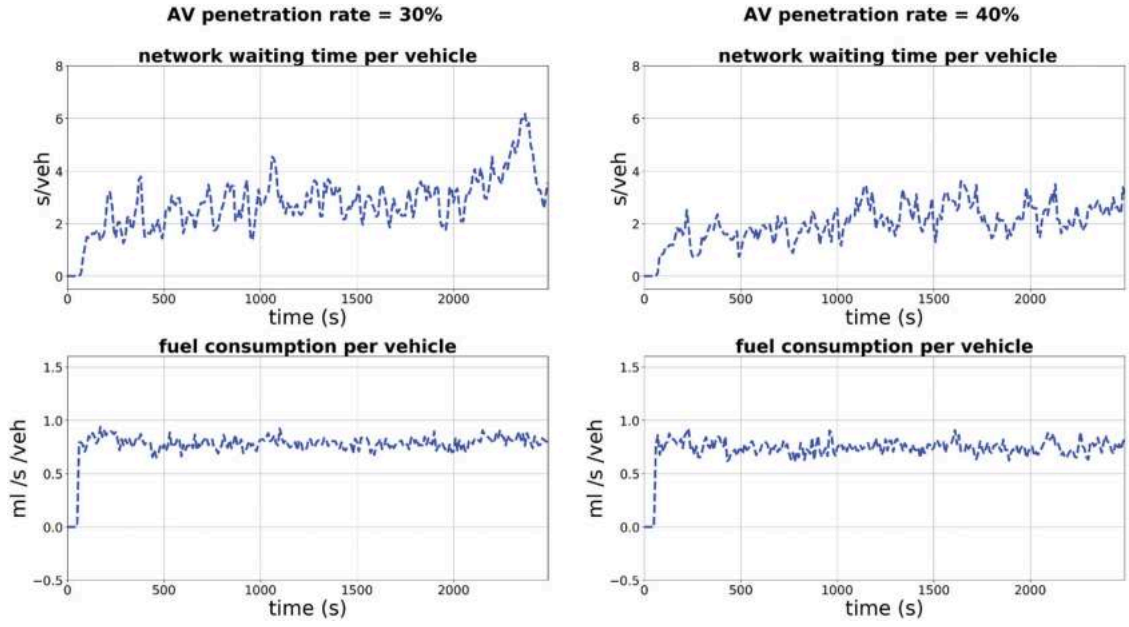

Figure 7. Performance of the proposed method for AV penetration rate of 50% in the left two subplots and 100% in the right two subplots.

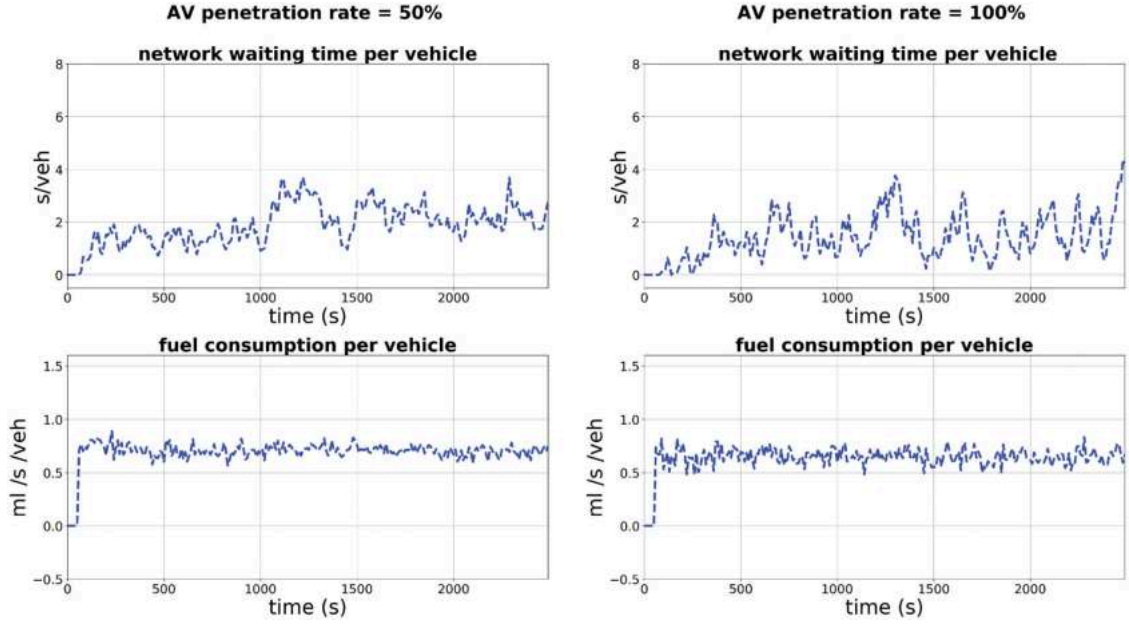

Figure 8. The average waiting time per vehicle and average fuel consumption per vehicle for different AV penetration rates. Results show that increasing the number of AVs in the network improves the traffic flow by decreasing both waiting time and fuel consumption.

traffic flow in the second case is 150% of the flow used in the previous simulations (results are shown in Figure 9). In both cases, comparison is made between two AV penetration rates; 0% and 10%. The mean waiting time and fuel consumption per vehicle from time 500 *s* to 2500 *s* are calculated for each case, and the results are summarized in Figure 10. In the simulation with 50% traffic of the original flow, the 10% penetration rate reduces the waiting time by 15% and the fuel consumption by 8%. When the flow is increased to 150% of the original flow, the 10% penetration rate improves the traffic network by decreasing the waiting time by 32% and the fuel consumption by 18% (see the results shown in Figure 11). The results validate the capability of the proposed MPC-based speed adjustment in improving the traffic system, considering relatively high and low flows in the four-intersection network used in this paper.

Results with micro-simulation calibration

It is noted that calibrating a simulation environment to match the field data is an important step toward simulation models and results validation (Abuamer et al. 2017). In our simulation setup, we expect that after flow calibration, a specific speed profile in each incoming lane is realized. Since our setup is a hypothetical network, instead of using real-world data, hypothetical flow and speed profiles shown in Figure 12 are generated. To calibrate the traffic flow and velocity,

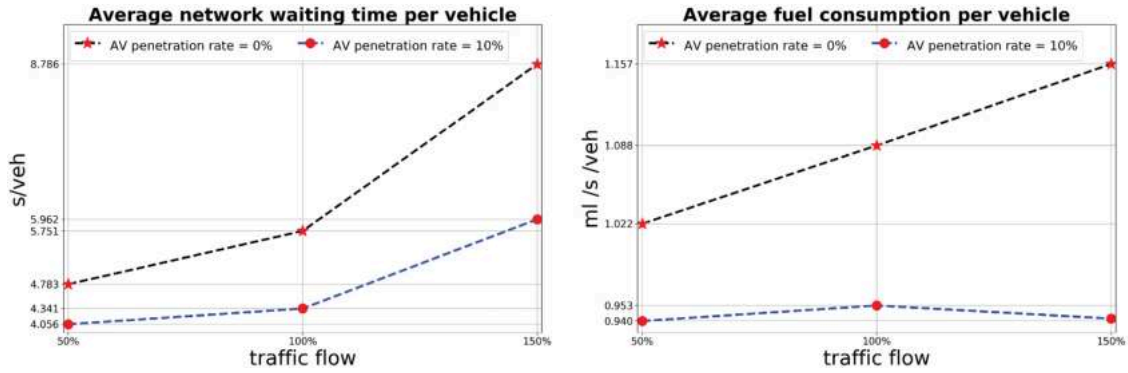

Figure 9. Performance of the proposed method when the average traffic flow is 50% of the average flow considered in the previous simulation studies (Figures 4–8). Results for AV penetration rate of 0% are shown in the left subplots and 10% in the right subplots.

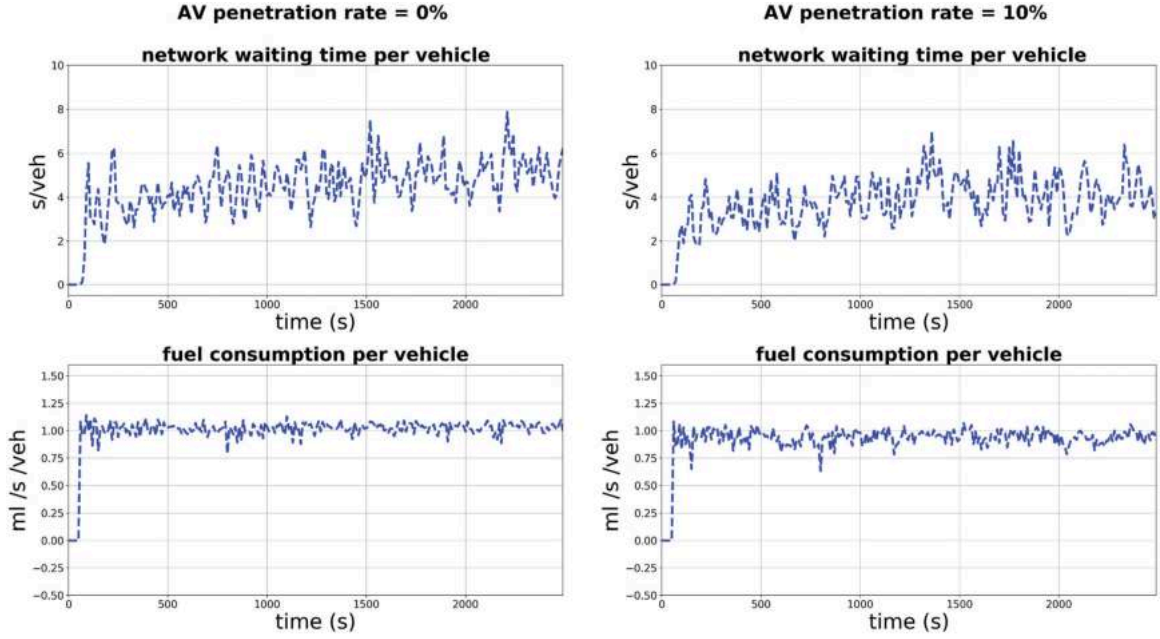

Figure 10. The average waiting time per vehicle and average fuel consumption per vehicle for different vehicle flows. Results show that the proposed AV speed control method is successfully able to reduce the network waiting time and fuel consumption with high and low network average traffic flow (note that 100% traffic flow is the flow in the previous simulation studies (Figures 4–8)).

the SUMO calibrator is employed, which may add or remove vehicles and change the velocity of a lane to reach a specified flow/speed profile. In each of the incoming lanes, one detector is placed (detectors D_1^V-D_4^V, and D_1^H-D_4^H as shown in Figure 3). Similar to Abuamer et al. (2017), Geoffrey E. Heavers (GEH) statistics is used to compare the calibrated simulation results with the desired performance. GEH is calculated as follows:

$$GEH = \sqrt{\frac{2[R(k)-S(k)]^2}{R(k)+S(k)}}, \tag{12}$$

where $R(k)$ is the real-world flow (in our case, the desired flow), while $S(k)$ is the simulation data. GEH statistics for all detectors are summarized in Table 2. Results demonstrate that at least 85% of GEH values are within the error equal or less than five at all lanes' loop detectors. Hence, the hypothetical setup can now represent the hypothetical expected desired flow (Abuamer et al. 2017; Sadat and Celikoglu 2017). The speed profiles in the simulation environment after calibration for D_1^H, as well as D_1^V are depicted in Figure 13.

Simulation studies after calibrating simulation environment

For the rest of our simulation studies, the calibrated network is used to investigate the impact of the proposed AV speed adjustment method in the mixed autonomy environment with different AV penetration rates. Four criteria are used to evaluate the impact of AV penetration rate on the traffic network's performance: average waiting time, average fuel consumption, average travel time, and the number of stop-and-go movements (SAGs). According to Figures 14 and 15, *increasing the number of AVs in the traffic network results in lower waiting time, fuel consumption, and SAG movements (except for the* 50% *penetration rate), but it comes with the cost of increasing the travel time*. Similar to the results in the subsection 'Simulation studies without a micro-simulation calibration,' even with a small number of AVs in the network, the improvement in waiting time and fuel consumption is noticeable. The percentage of the improvements compared to the 0% penetration rate is shown in Figure 16. The case study considering the 10% penetration rate reveals that the average waiting time per vehicle is

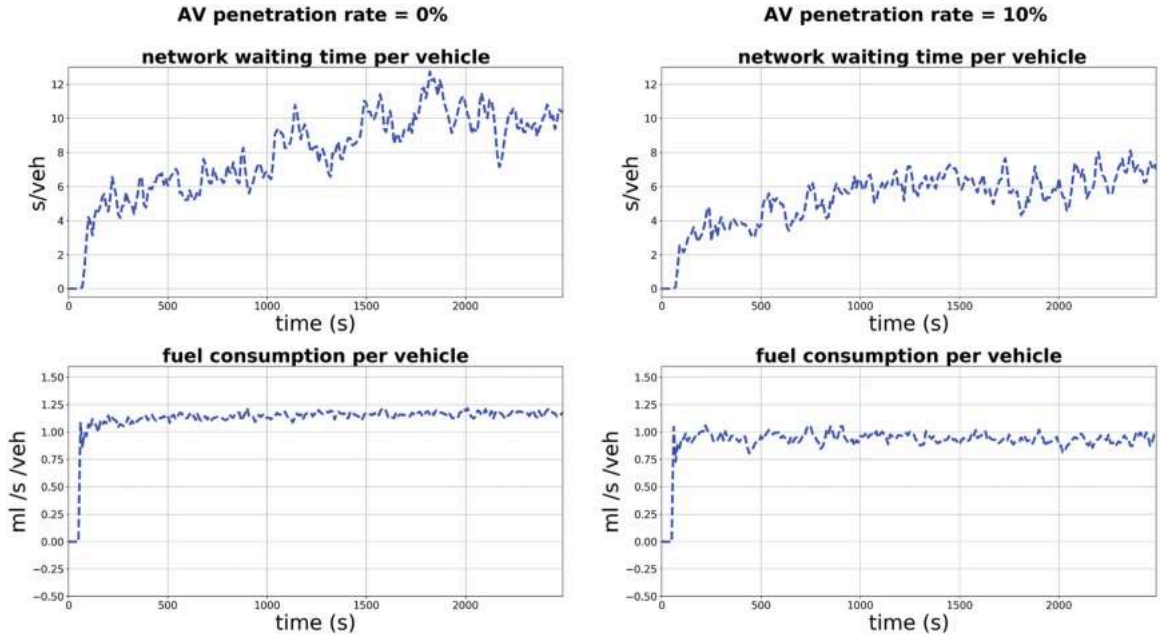

Figure 11. Performance of the proposed method when the average traffic flow is 150% of the average flow considered in the previous simulation studies (Figures 4–8). Results for AV penetration rate of 0% are shown in the left subplots and 10% in the right subplots.

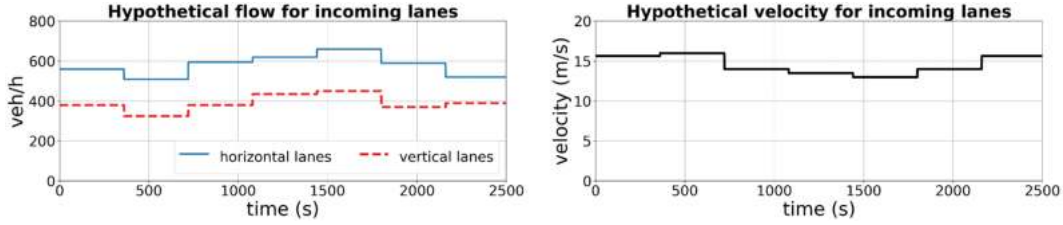

Figure 12. Desired flow and speed profiles for the micro-simulation calibration.

Table 2. GEH statistics summary.

Detector	Mean	Variance	Detector	Mean	Variance
D_1^H	2.36	2.44	D_1^V	2.25	3.09
D_2^H	2.28	2.30	D_2^V	2.71	3.74
D_3^H	2.08	3.23	D_3^V	2.24	2.59
D_4^H	1.88	3.49	D_4^V	2.31	1.53

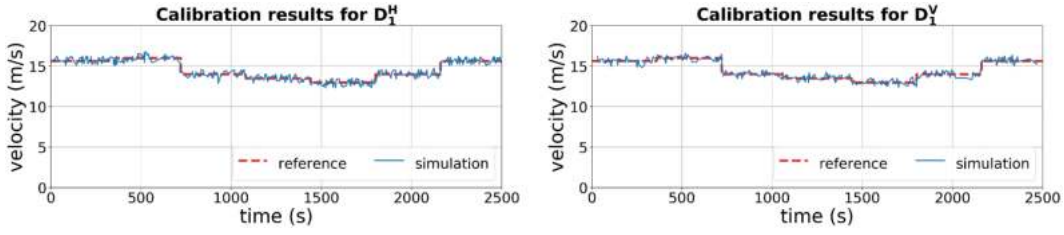

Figure 13. Desired and simulated speed profiles in one of the horizontal and vertical lanes.

reduced by 27%, the average fuel consumption per vehicle is reduced by 8%, and the number of SAGs is reduced by 21%; however, the average travel time is increased by 5%. When the penetration rate is increased to 50%, the waiting time, fuel consumption, and the number of SAGs are reduced by 75%, 26%, and 43%, respectively. However, the average travel time is increased by 26%. In the case that all vehicles in the network are autonomous, the waiting time is reduced by 88%, the fuel consumption is reduced by 39%, the SAG count is reduced by 78%, yet the travel time is increased by 36%.

Although a network including only AVs seems to be the ideal case because of the lowest waiting time, fuel consumption, and SAG counts, the noticeable increase in the travel time makes it less appealing. Besides, even a small number of AVs shows a promising impact on

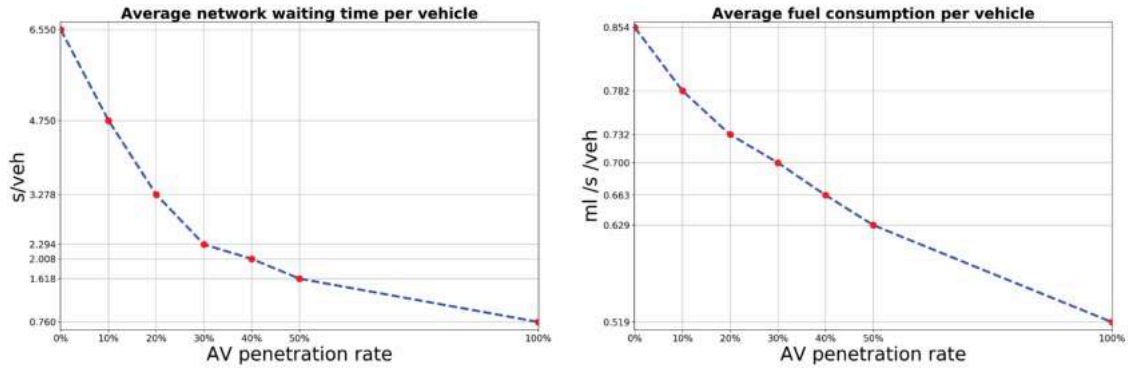

Figure 14. The average waiting time per vehicle and average fuel consumption per vehicle for different AV penetration rates. Results show that increasing the number of AVs in the network improves the traffic flow by decreasing both the waiting time and fuel consumption. It is noted that the results are consistent with the ones presented earlier using the uncalibrated simulation environment (see Figure 5).

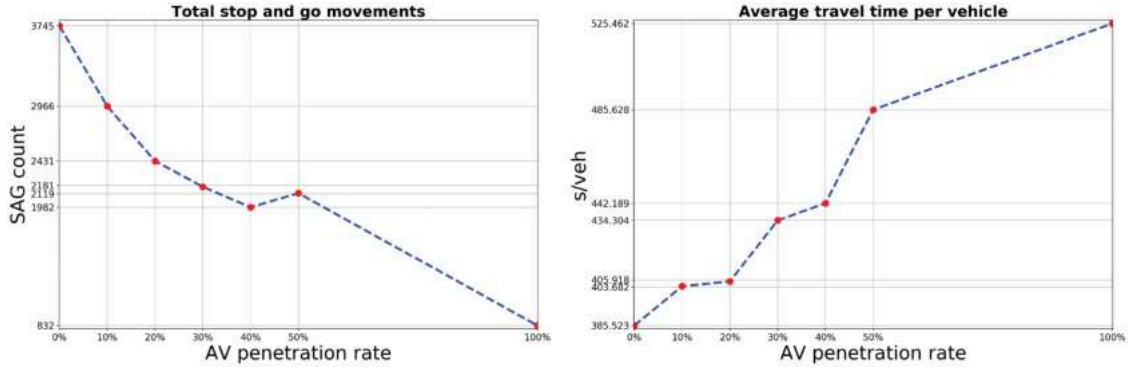

Figure 15. Total stop-and-go (SAG) movements and average travel time per vehicle for different AV penetration rates. Generally, increasing the AV penetration rate results in higher travel time in the network, while the number of SAGs reduces except for the 50% penetration rate.

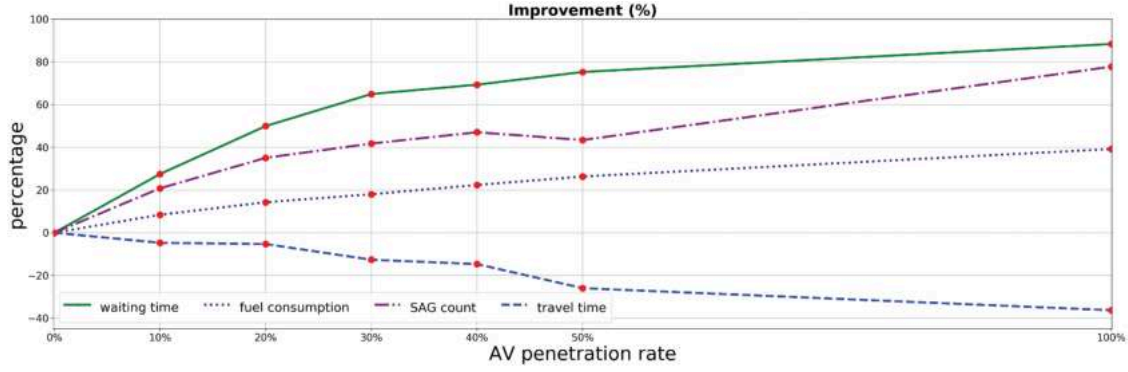

Figure 16. The improvement in waiting time, fuel consumption, SAG movements count, and travel time in percent compared to the case that no AV is present in the traffic network. According to the results, it is concluded that 40% penetration rate is of great importance since going from 40% to 50% results in higher SAG counts and a noticeable increase in travel time.

improving the traffic system. Based on the results in Figures 14–16, we notice that 40% penetration rate is of great importance since going from 40% to 50% penetration rate results in higher SAG counts and a noticeable increase in travel time.

Conclusion

In this work, we examined the impact of an MPC-based AV speed adjustment in signalized intersections in a mixed autonomy environment, where each traffic light is controlled using a deep Q-learning RL algorithm. In the proposed speed adjustment method, AVs plan their trajectory by solving an MPC problem to minimize their acceleration/deceleration and avoid stopping at the intersection as much as possible. Simulation results show that AV speed adjustment can improve the traffic system efficacy. In a network of four connected intersections controlled by independent RL agents, with only a relatively small number of AVs (10% penetration rate), the vehicle waiting time and fuel consumption

are noticeably reduced. In addition, reduction of network waiting time and average fuel consumption becomes more significant as the number of AVs in the network increases.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research was supported by the National Science Foundation under grant [CNS-1931981].

ORCID

Javad Mohammadpour Velni http://orcid.org/0000-0001-8546-221X

References

Abdoos, M., N. Mozayani, and A. L. Bazzan. 2011. "Traffic Light Control in Non-stationary Environments Based on Multi Agent q-learning." In 2011 14th International IEEE conference on intelligent transportation systems (ITSC), 1580–1585. Washington, DC, USA: IEEE.

Abuamer, I. M., M. Sadat, M. A. Silgu, and H. B. Celikoglu. 2017. "Analyzing the Effects of Driver Behavior within an Adaptive Ramp Control Scheme: A Case-study with Alinea." In 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)109–114. Vienna, Austria: IEEE.

Agrawal, A., R. Verschueren, S. Diamond, and S. Boyd. 2018. "A Rewriting System for Convex Optimization Problems." *Journal of Control and Decision* 5: 42–60. doi:10.1080/23307706.2017.1397554.

Aziz, H. A., F. Zhu, and S. V. Ukkusuri. 2018. "Learning-based Traffic Signal Control Algorithms with Neighborhood Information Sharing: An Application for Sustainable Mobility." *Journal of Intelligent Transportation Systems* 22: 40–52. doi:10.1080/15472450.2017.1387546.

Chavoshi, K., A. Genser, and A. Kouvelas. 2021. "A Pairing Algorithm for Conflict-free Crossings of Automated Vehicles at Lightless Intersections." *Electronics* 10: 1702. doi:10.3390/electronics10141702.

Chollet, F., et al. 2015. "Keras documentation." keras. io 33.

Chowdhury, M. A., and A. W. Sadek. 2003. *Fundamentals of Intelligent Transportation Systems Planning*. Norwood, MA, USA: Artech House.

Diamond, S., and S. Boyd. 2016. "CVXPY: A Python-embedded Modeling Language for Convex Optimization." *Journal of Machine Learning Research* 17: 1–5.

El-Tantawy, S., B. Abdulhai, and H. Abdelgawad. 2014. "Design of Reinforcement Learning Parameters for Seamless Application of Adaptive Traffic Signal Control." *Journal of Intelligent Transportation Systems* 18: 227–245. doi:10.1080/15472450.2013.810991.

Genders, W., and S. Razavi. 2016. "Using a Deep Reinforcement Learning Agent for Traffic Signal Control." arXiv preprint arXiv:1611.01142 .

Ge, H., Y. Song, C. Wu, J. Ren, and G. Tan. 2019. "Cooperative Deep q-learning with q-value Transfer for Multi-intersection Signal Control." *IEEE Access* 7: 40797–40809. doi:10.1109/ACCESS.2019.2907618.

He, Q., K. L. Head, and J. Ding. 2011. "Heuristic Algorithm for Priority Traffic Signal Control." *Transportation Research Record* 2259: 1–7. doi:10.3141/2259-01.

Levin, M. W., and S. D. Boyles. 2016. "A Multiclass Cell Transmission Model for Shared Human and Autonomous Vehicle Roads." *Transportation Research Part C: Emerging Technologies* 62: 103–116. doi:10.1016/j.trc.2015.10.005.

Lin, P., J. Liu, P. J. Jin, and B. Ran. 2017. "Autonomous Vehicle-intersection Coordination Method in a Connected Vehicle Environment." *IEEE Intelligent Transportation Systems Magazine* 9: 37–47. doi:10.1109/MITS.2017.2743167.

Li, M., X. Wu, X. He, G. Yu, and Y. Wang. 2018. "An Eco-driving System for Electric Vehicles with Signal Control under v2x Environment." *Transportation Research Part C: Emerging Technologies* 93: 335–350. doi:10.1016/j.trc.2018.06.002.

Li, Z., Q. Wu, H. Yu, C. Chen, G. Zhang, Z. Z. Tian, and P. D. Prevedouros. 2019. "Temporal-spatial Dimension Extension-based Intersection Control Formulation for Connected and Autonomous Vehicle Systems." *Transportation Research Part C: Emerging Technologies* 104: 234–248. doi:10.1016/j.trc.2019.05.003.

Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flo¨tter¨od, R. Hilbrich, L. Lu¨cken, J. Rummel, P. Wagner, and E. Wießner. 2018. "Microscopic Traffic Simulation Using Sumo." In The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE. https://elib.dlr.de/124092/

Mannion, P., J. Duggan, and E. Howley. 2016. "An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control." In *Autonomic Road Transport Support Systems*, 47–66. Switzerland: Springer.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, et al. 2015. "Human-level Control through Deep Reinforcement Learning". *nature* 518: 529–533. doi:10.1038/nature14236.

Qu, Z., Z. Pan, Y. Chen, X. Wang, and H. Li. 2020. "A Distributed Control Method for Urban Networks Using Multi-agent Reinforcement Learning Based on Regional Mixed Strategy Nash-equilibrium." *IEEE Access* 8: 19750–19766. doi:10.1109/ACCESS.2020.2968937.

Raffo, G. V., G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker. 2009. "A Predictive Controller for Autonomous Vehicle Path Tracking." *IEEE Transactions on Intelligent Transportation Systems* 10: 92–102. doi:10.1109/TITS.2008.2011697.

Rakha, H., and R. K. Kamalanathsharma. 2011. "Eco-driving at Signalized Intersections Using v2i Communication." In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 341–346. Washington, DC, USA: IEEE.

Sadat, M., and H. B. Celikoglu. 2017. "Simulation-based Variable Speed Limit Systems Modelling: An Overview and a Case Study on Istanbul Freeways." *Transportation Research Procedia* 22: 607–614. doi:10.1016/j.trpro.2017.03.051.

Steingrover, M., R. Schouten, S. Peelen, E. Nijhuis and B. Bakker. 2005. *Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion*, 216–223. Brussels, Belgium: BNAIC, Citeseer.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT press.

Tajalli, M., M. Mehrabipour, and A. Hajbabaie. 2020. "Network-level Coordinated Speed Optimization and Traffic Light Control for Connected and Automated Vehicles 22 (11).

Watkins, C. J., and P. Dayan. 1992. "Q-learning." *Machine Learning* 8: 279–292. doi:10.1007/BF00992698.

Wu, C., A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen. 2017. "Flow: A Modular Learning Framework for Autonomy in Traffic." arXiv preprint arXiv:1710.05465 .

Xu, B., X. J. Ban, Y. Bian, J. Wang, and K. Li. 2017. "V2i Based Cooperation between Traffic Signal and Approaching Automated Vehicles." 2017 IEEE Intelligent Vehicles Symposium (IV), 1658–1664. Redondo Beach, CA, USA: IEEE.

Yang, K., S. I. Guler, and M. Menendez. 2016. "Isolated Intersection Control for Various Levels of Vehicle Technology: Conventional, Connected, and Automated Vehicles." *Transportation Research Part C: Emerging Technologies* 72: 109–129. doi:10.1016/j.trc.2016.08.009.

Yau, K. L. A., J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk. 2017. "A Survey on Reinforcement Learning Models and Algorithms for Traffic Signal Control." *ACM Computing Surveys (CSUR)* 50: 1–38. doi:10.1145/3068287.

Zhang, R., A. Ishikawa, W. Wang, B. Striner, and O. Tonguz. 2018. "Intelligent Traffic Signal Control: Using Reinforcement Learning with Partial Detection." arXiv preprint arXiv:1807.01628.

Zhao, W., R. Liu, and D. Ngoduy. 2019. "A Bilevel Programming Model for Autonomous Intersection Control and Trajectory Planning." *Transportmetrica A: Transport Science* 1–25.