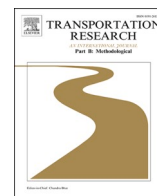




Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Platoon-centered control for eco-driving at signalized intersection built upon hybrid MPC system, online learning and distributed optimization part I: Modeling and solution algorithm design

Hanyu Zhang, Lili Du^{*}

Department of Civil and Coastal Engineering, University of Florida, Gainesville, FL 32608, United States

ARTICLE INFO

Keywords:

Connected and autonomous vehicle
Platoon-centered control
Eco-driving strategy
Model predictive control
Hybrid MPC system
Distributed optimization

ABSTRACT

Inspired by connected and autonomous vehicle (CAV) technologies, extensive studies have developed open-loop vehicle-level trajectory planning or speed advisory to promote eco-driving at traffic intersections. But few studies work on platoon-level closed-loop trajectory control, which can better sustain stream traffic smoothness and efficiency. Motivated by this research gap, this study developed a system optimal platoon-centered control for eco-driving (PCC-eDriving), which can guide a platoon mixed with connected and autonomous vehicles (CAVs) and human-driven vehicles (HDVs) to smoothly approach, split as needed, and then sequentially pass signalized intersections, while reducing or even avoiding sharp deceleration and red idling. The effort is separated to Part I and Part II to prevent a lengthy article. Specifically, Part I of this study modeled the PCC-eDriving as a hybrid Model Predictive Control (MPC) system. It involves three MPC controllers for platoon trajectory control and a mixed-integer nonlinear program (MINLP) for optimal splitting decisions. Each MPC controller is integrated with robust vehicle dynamics and an online adaptive curve learning algorithm to factor control and vehicle driving uncertainties. An active-set-based optimal condition decomposition algorithm (AS-OCD) was developed to efficiently solve the MPC controllers' large-scale optimizers in a distributed manner. The numerical experiments built upon the field and simulated data indicated that the PCC-eDriving could significantly improve traffic smoothness and efficiency while reducing energy consumption and emission at urban signalized intersections. Part II will analyze and prove the sequential feasibility and the Input-to-State stability of the hybrid MPC system, as well as the convergence of the AS-OCD solution approach to theoretically sustain the performance of the hybrid MPC system.

1. Introduction

It has been observed that the traffic signals often cause vehicle idling and stop-and-go traffic fluctuations and thus lead to tremendous traffic congestion at urban traffic intersections. Inspired by connected and autonomous vehicle (CAV) technologies,

DOI of original article: <https://doi.org/10.1016/j.trb.2023.03.008>.

Abbreviations: CAV, Connected and autonomous vehicle; MPC, Model predictive control; PCC-eDriving, Platoon centered control for eco-driving; HDV, Human drive vehicle.

^{*} Corresponding author.

E-mail address: lilidu@ufl.edu (L. Du).

<https://doi.org/10.1016/j.trb.2023.02.006>

Received 16 December 2021; Received in revised form 30 October 2022; Accepted 7 February 2023

Available online 8 April 2023

0191-2615/© 2023 Elsevier Ltd. All rights reserved.

various eco-driving strategies and algorithms have been developed in recent literature, aiming to improve traffic safety and efficiency and reduce energy consumption and emission at signalized intersections. Specifically, by leveraging V2I communication technologies, some studies developed heuristic or optimization model-based approaches to provide either (i) vehicle-level speed advisory, including constant speed plan (e.g., He et al., 2015; Mandava et al., 2009; Alsabaan et al., 2013; Wan et al., 2016; Simchon and Rabinovici, 2020) and real-time dynamic speed control (e.g., Asadi and Vahidi, 2010; Kamal et al., 2012; Sun et al., 2020; Nie and Farzaneh, 2021), or (ii) platoon-level eco-driving strategy by considering an entire platoon as a system and instructing entire platoon's trajectory plan (Wei et al., 2017; Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Wang et al., 2019; Ma et al., 2021; Chen et al., 2021), responding to predefined signal control schedules. Other studies developed joint optimization to generate proactive schemes for coordinating the vehicle platoon trajectory control and the traffic signal control (Feng et al., 2018; Guo et al., 2019; Niroumand et al., 2020), aiming to improve the traffic smoothness at the intersection. However, most of these studies focused on optimizing the leading vehicle's trajectory rather than the entire platoon's movements due to the modeling and computation complexity. Recent studies (e.g., Lioris et al., 2016) showed that the platoon-level eco-driving strategies outperform vehicle-level speed advisory in improving flow efficiency. Thus, they attracted extensive research interest in recent literature (Zhao et al., 2018; Qin et al., 2021; Zhen et al., 2022) and motivated this study.

Apart from the model-based approaches, researchers recently started to use learning-based methods, especially deep reinforcement learning, to develop eco-driving algorithms (Hu et al., 2018; Shi et al., 2018; Qu et al., 2020; Guo et al., 2021; Peng et al., 2022; Huo and Meckl, 2022) by leveraging their merits in tracking vehicle dynamics and handling complex driving scenarios. However, those learning-based methods consider individual vehicles to be non-cooperative. They learn responsive instruction without rigorously factoring the entire platoon's performance and thus cannot ensure system optimality. On the other hand, it is challenging to develop platoon-level eco-driving strategies by reinforcement learning approach. This is because the dimensions of CAVs' actions and states will grow exponentially when more CAVs are considered as platoon members under a reinforcement learning control. It will further jeopardize the computation efficiency and convergence of the reinforcement learning.

Along with the above thoughts, this study intends to use model-based approaches to develop a novel platoon-level eco-driving scheme. Specifically, we will develop a hybrid MPC control system, which guides the platoon through traffic intersections smartly and smoothly by responding to given traffic signal schedules. The hybrid MPC control system involves several enhanced features which fill in the research gaps in literature and also distinguish our study from existing efforts.

First of all, many existing model-based eco-driving strategies apply open-loop planning schemes to suggest platoon trajectories during a time horizon periodically rather than using feedback based closed-loop control (Wei et al., 2017; Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Wang et al., 2019; Chen et al., 2021). Although showing improved platoon performance in saving energy consumption and reducing emissions, these planning schemes usually cannot adapt to the trajectory deviations in a planning horizon resulting from various uncertain factors. Accordingly, many studies work on a pure CAV platoon (Wei et al., 2017; Faraj et al., 2017; Li et al., 2018; Ma et al., 2021). Even though several recent efforts have started to consider mixed-flow traffic (Zhao et al., 2018; Wang et al., 2019; Chen et al., 2021), they often ignore the driving behavior variations of the human-driven vehicles (HDVs), such as reaction time variation under different traffic environments. On the other hand, most platoon-level eco-driving strategies use deterministic double-integrator vehicle dynamics without factoring in control uncertainty resulting from powertrain delay, aerodynamic drag, etc. Nevertheless, modeling these uncertainties into platoon-level eco-driving strategies will lead to nonconvex optimizers or controllers. It consequently raises the extra difficulty of theoretically analyzing the control property and developing an efficient solution approach for this real-time application. This study intends to partially bridge these gaps by developing a robust model predictive control (MPC), factoring the uncertainties resulting from both CAV and HDV.

We also noticed that majority of platoon-level eco-driving strategies only optimize the leading vehicle's trajectory while implementing the following vehicles with existing car-following control, such as adaptive cruise control (ACC) (Marsden et al., 2001; Xiao and Gao, 2010; Guanetti et al., 2018) or cooperative adaptive cruise control (CACC) (Milanés et al., 2013; Wei et al., 2017; Li et al., 2018; Smith et al., 2020; Ma et al., 2021). These traditional ACC and CACC are vehicle-centered responsive controllers for individual CAVs to determine their own optimal driving decisions. They do not directly optimize the platoon performance and thus cannot mathematically guarantee the system optimality, even though some control features such as string stability can be ensured (Ploeg et al., 2011; Öncü et al., 2014). Given this weakness, this study considers an entire platoon as a system and employs a system optimal MPC to systematically control the trajectories of all CAVs in a platoon in real time. This type of platoon-centered control has demonstrated superior performance in maintaining platoon driving efficiency and smoothness (Gong et al., 2016; Gong and Du, 2018). However, these merits come with new challenges, particularly for this study. Specifically, MPC often involves an optimizer, which must be solved within a control interval (< 1 s) to ensure control continuity. When the MPC is built for the entire platoon, it will introduce a large-scale optimizer that is challenging to be solved promptly, especially when the platoon is long, or the MPC uses a long prediction horizon in this study. In addition, existing platoon-centered MPC control (Gong et al., 2016; Gong and Du, 2018; Shen et al., 2022) applies a constant desired spacing policy to facilitate control stability. It leads to a low-capacity usage and does not fit the urban road scenario. To address these challenges, this study integrates an adaptive desired spacing policy in the MPC and then designs a new customized distributed optimization algorithm to solve MPC efficiently.

Furthermore, platoon-level eco-driving strategies often meet the challenge to properly split a long platoon if it cannot wholly pass the intersection within one green interval. Some existing studies (Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Chen et al., 2021) used heuristic splitting rules to maximize traffic throughputs. Other studies (Guanetti et al., 2018) proposed platoon splitting control integrated with platoon merging control, aiming to have individual vehicles cut in the platoon quickly (Dasgupta et al., 2017; Duret et al., 2020; Pauca et al., 2021; Zhang et al., 2022). Those approaches with a single objective often cause severe traffic fluctuation and jeopardize traffic smoothness. This study, therefore, intends to develop an optimal platoon spitting scheme by co-considering traffic

throughput, smoothness, and energy consumption. More importantly, we noticed that splitting a platoon at an intersection will induce three platoon-control states, respectively for the car-following, splitting, and intersection passing. Accordingly, finding the optimal splitting point and generating trajectory control instructions under each state should cooperate and enable smooth state switching. Most existing literature ignored these control transitions. This study intends to bridge this gap by designing a hybrid MPC system, which harmonizes three MPC controllers and the splitting point searching optimizer so that we can ensure optimal platoon performance while facilitating the smooth state switching process.

The aforementioned research gaps and enhanced features of our MPCs raise new research challenges and then highlight the unique methodology contributions of this study. Overall, this study develops an efficient system optimal platoon-centered control for eco-driving (PCC-eDriving) at a signalized intersection, which instructs a mixed flow platoon to smoothly and efficiently approach, split, and then pass signalized intersections reducing red idling as much as possible. The development of the PCC-eDriving control contributes the following mathematical modeling, theoretical analyses, and algorithm design.

First of all, we designed a hybrid MPC system to enable the PCC-eDriving control. The hybrid MPC system involves three MPC controllers and a mixed-integer nonlinear program (MINLP), which together generate optimal control laws/switching signals for the platoon to approach an intersection, split into sub-platoons, and then sequentially pass the intersection during different green intervals.

Next, this study designed an online adaptive curve learning algorithm and integrated it into each MPC controller so that the PCC-eDriving control can quickly learn and predict the HDVs' real-time driving behaviors. It outperforms existing approaches (Zhang, 1994; Gong and Du, 2018) by using shorter warm-up time and fewer computation resources. Thus, it fits the urban road scenarios better since HDV behavior varies more frequently. Besides, the platoon has a shorter time to learn the HDV behaviors when approaching an urban intersection than running on a long stretch freeway segment.

The MPC controllers employ a long prediction horizon (>30 s) for factoring traffic variation during a traffic signal phase in the PCC-eDriving control. But it leads to large-scale optimization problems and poses tremendous challenges to solution approaches. We designed an active-set based optimal condition decomposition approach (AS-OCD) and proved its convergence performance by taking advantage of the problem features. The AS-OCD algorithm distributes the computation loads to individual CAVs in the platoon. It effectively solves the computation difficulty and sustains the continuity of PCC-eDriving control.

Last, this study conducted numerical experiments to validate the performance of the PCC-eDriving control and the involved approaches. Our experiments showed that the adaptive curve learning algorithm can accurately and quickly predict the HDV driving behaviors and the AS-OCD approach can efficiently solve the MPC controllers online within 0.2 s. Besides, the employment of the adaptive desired spacing policy can significantly improve the road capacity by 44% compared with that using the constant desired spacing policy. Incorporating CAV and HDV uncertainties into the platooning control smoothened the traffic oscillations and reduced energy consumption. Furthermore, we compared the proposed PCC-eDriving control with an existing CACC controller and eco-driving strategy developed by Faraj et al. (2017) in the literature. The results indicated that the PCC-eDriving control can improve the traffic throughputs and smoothness while significantly reducing the energy consumption and emission by 50 and 7%.

To the best of our knowledge, this is the first research to use hybrid MPC system to study the entire process of platoon approaching, splitting and then passing intersections. It significantly contributes the methodology development and practice for the field of eco-driving at intersections.

The rest efforts of this study are presented by the structure as follows. Section 2 formally defines the research problem, assumptions and notations, and Section 3 introduces vehicle dynamics and constraints. Built upon that, we develop the hybrid MPC system in Section 4, which is solved by the AS-OCD solution approach and adaptive curve learning approach developed in Section 5 and Section 6, respectively. The performances of the PCC-eDriving and the associated approaches are validated by the numerical experiments in Section 7, and finally we draw a conclusion for the whole study in Section 8. To be noted, the theoretical properties of the MPC controller and the AS-OCD will be analyzed and proved in Part II.

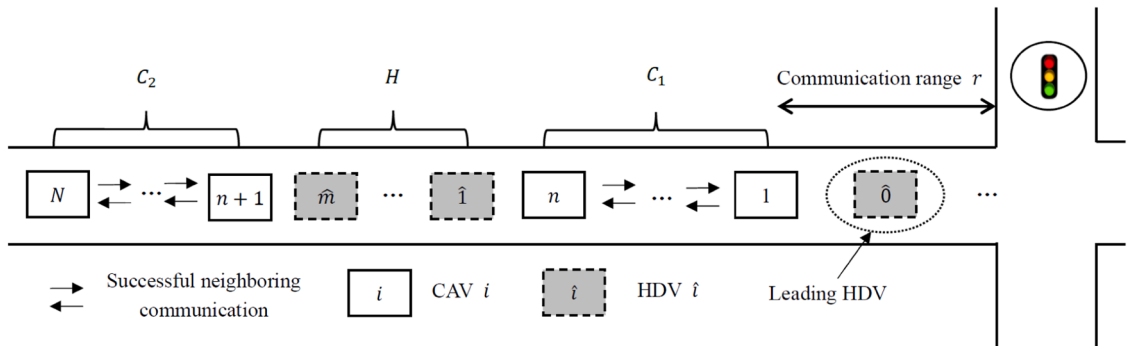


Fig. 1. Sample platoon at the signalized intersection.

2. Problem statement

This study proposes a system optimal platoon-centered control for eco-driving (PCC-eDriving), which guides a mixed flow platoon to efficiently approach and pass signalized intersections with the aim to improve traffic smoothness and throughputs and reduce energy consumption and emission. It considers a sample mixed flow platoon moving toward a signalized intersection on an urban road in Fig. 1. The sample platoon may either follow a leading HDV $\hat{0}$ or drive freely without leading HDV. It contains three sequential platoon segments: a CAV platoon segment C_1 including n -many CAVs labeled as $\{1, \dots, n\}$, a HDV platoon segment H including $\hat{m}$ -many HDVs indexed by $\{\hat{1}, \dots, \hat{m}\}$, and then a CAV platoon segment C_2 with CAVs labeled as $\{n+1, \dots, N\}$. CAV platoon segments C_1 and C_2 form a CAV set $C = C_1 \cup C_2$. A more general mixed flow platoon can be easily generated via repeating this sample platoon.

Within a CAV platoon, we consider a preceding-and-following communication network topology, which reduces the communication delay to the maximum extent. Specifically, we think that the CAVs in the platoon segments C_1 and C_2 are well connected without communication failures and will communicate with their immediately preceding and following CAVs through V2V communication without unbounded delay. Particularly, CAV $i = n$ in CAV platoon segment C_1 can communicate with CAV $i = n+1$ in CAV platoon segment C_2 assuming they are within the communication range. Accordingly, the first leading CAV $i = 1$ will only communicate with its immediately following CAV $i = 2$. Besides, this study considers the movements of the HDVs $\hat{0}$, $\hat{1}$ and $\hat{m}$ can be detected by the adjacent CAVs using on-board sensors.

To mathematically model the proposed study, we use notations x_i, v_i, u_i for $\forall i \in I_C = \{1, \dots, N\}$ and $\hat{x}_i, \hat{v}_i, \hat{u}_i$ for $\forall \hat{i} \in I_H = \{\hat{0}, \hat{1}, \dots, \hat{m}\}$ to represent the longitudinal position, speed, and acceleration of the i_{th} CAV and the $\hat{i}_{th}$ HDV, respectively. The platoon control is conducted at discrete time steps (indexed by $k \in \mathbb{Z}_+ := \{0, 1, 2, \dots\}$) with a control time interval $\tau > 0$. Control inputs u_i ($i \in I_C$) keep constant during an interval τ . For notational simplicity, we use $x_i(k), v_i(k), u_i(k)$ to substitute $x_i(\tau k), v_i(\tau k), u_i(\tau k)$ hereafter.

When the leading platoon segment C_1 reaches the communication zone, namely r distance from the traffic signal, it will acquire and receive the real-time traffic signal information. This information includes green and red phase intervals T_g/T_r as well as the remaining time of the current phase, either a green interval denoted by $\tilde{T}_g$ or a red interval denoted by $\tilde{T}_r$. Moreover, we represent traffic signal intervals by $T_g = \tau k_g, T_r = \tau k_r, \tilde{T}_g = \tau \tilde{k}_g, \tilde{T}_r = \tau \tilde{k}_r$, where $k_g/\tilde{k}_g$ and $k_r/\tilde{k}_r$ describe the number of control time steps covered by the green and red intervals, respectively. To be noted, a recent report ‘‘Adaptive Signal Control Technology’’ issued by DOT FHWA indicates that the traffic signal will be updated at most every several minutes. Given the PCC-eDriving control typically guides the platoon to pass the intersection in a few seconds, we assume that these traffic signal information $T_g, T_r, \tilde{T}_g, \tilde{T}_r$ will keep constant during the PCC-eDriving.

In reality, a long platoon under PCC-eDriving control may need to split into several sub-platoons to sequentially pass the intersection during different consecutive traffic cycles. We use hybrid MPC system including three states q_0, q_1, q_2 , and two switching signals σ_0, σ_1 to mathematically describe this process. Specifically, the mixed flow platoon A on urban road is originally under car-following state q_0 approaching the signalized intersection. When it enters the communication zone and obtains traffic signal information, a switching signal σ_0 optimizer is used to determine the optimal platoon splitting point. Following this splitting decision, platoon A under the car-following state q_0 in Fig. 2 splits into two sub-platoons A_1 and A_2 . The leading sub-platoon A_1 under intersection passing state q_1 is expected to pass the intersection within the current cycle’s green interval, whereas the latter sub-platoon A_2 under splitting state q_2 consists of the rest vehicles that cannot pass the intersection in the current traffic signal cycle under our prediction. As the

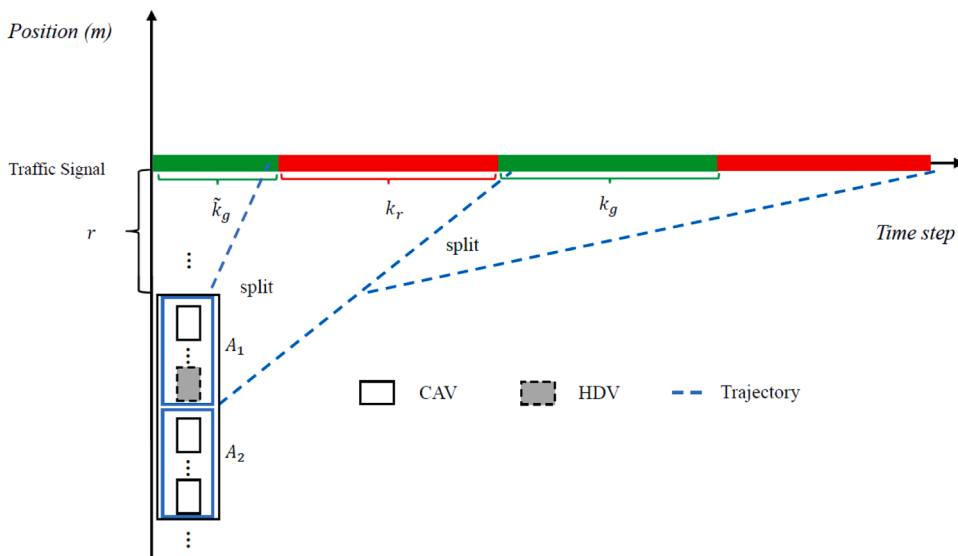


Fig. 2. Platoon splitting scheme.

leading sub-platoon A_1 under state q_1 passes the intersection, the switching signal σ_1 is triggered and sub-platoon A_1 restores car-following state q_0 . When the sub-platoon A_2 reaches the communication zone of the traffic signal, the above-mentioned procedure repeats. Specifically, the optimal platoon splitting point will be determined again and sub-platoon A_2 will further split into two sub-platoons if it cannot pass the intersection entirely in one green interval. Keep using this strategy, the proposed PCC-eDriving control modeled as a hybrid MPC system can be applied to any long platoon with any traffic signal setting. Besides, the mixed flow platoon can pass multiple signalized intersections by following the PCC-eDriving control for each traffic signal.

This PCC-eDriving control and the hybrid MPC system bring in technical challenges from mathematical modeling, theoretical analyses, and solution approach development. It thus motivates the novel methodology developed in this study. We formally present them in the following sections. To be noted, although this study considers one lane scenario to develop the PCC-eDriving control, it can be adapted to the traffic setup with multiple lanes by integrating other advance platooning control algorithms. More exactly, those advance platooning control algorithms will instruct the vehicles targeting the intersection's different (left-turn, right-turn, and going-through) lanes to switch lanes and form a platoon at a region farther away from the intersection (at least beyond the zone where the PCC-eDriving is activated). After that, the formed platoon on each lane will approach the intersection, start the PCC-eDriving control, and then pass the intersection under the eco-driving control independently. Please note that those advance control algorithms for diverging a platoon into different lanes and the platoon control involving left-turn and right-turn maneuvers are out of the scope of this study. Our other on-going or future work will address them.

3. Vehicle dynamics, constraints

This section introduces vehicle dynamics and constraints of the hybrid MPC system. It first introduces CAV dynamics, factoring the uncertainty at each control time step $k \in \mathbb{Z}_+$. Considering vehicles' powertrain delay and aerodynamic drag are stochastic and time-variant, this study adopts the robust double-integrator model in Eqs. (1) and (2) to describe CAV i 's dynamics at step k .

$$x_i(k+1) = x_i(k) + \tau v_i(k) + \frac{\tau^2}{2}(u_i(k) - \Delta u_i(k)), \quad i \in I_C \quad (1)$$

$$v_i(k+1) = v_i(k) + \tau(u_i(k) - \Delta u_i(k)), \quad i \in I_C \quad (2)$$

In Eqs. (1) and (2), $\Delta u_i(k)$ factors CAV i 's powertrain delay and aerodynamic drag by Eq. (3), which was proposed by Montanaro et al., (2020).

$$\Delta u_i(k) = \varepsilon_i v_i(k) + \eta_i u_i(k) - \eta_i u_i(k-1), \quad i \in I_C \quad (3)$$

In Eq. (3), parameter ε_i is the coefficient of a linearized aerodynamic force and η_i represents the time lag in the powertrain for each CAV $i \in I_C$. We consider ε_i and η_i are time-varying random parameters but bounded as $0 < \underline{\varepsilon}_i \leq \varepsilon_i \leq \bar{\varepsilon}_i$, $0 < \underline{\eta}_i \leq \eta_i \leq \bar{\eta}_i$, where $\underline{\varepsilon}_i$, $\bar{\varepsilon}_i$, $\underline{\eta}_i$ and $\bar{\eta}_i$ are corresponding lower and upper bounds for each CAV $i \in I_C$. Accordingly, the control disturbance $\Delta u_i(k)$ of CAV i is bounded as $\underline{\Delta}_i \leq \Delta u_i(k) \leq \bar{\Delta}_i$. For MPC at every time step, ε_i and η_i are randomly generated as parameters following a given distribution (e.g., uniform distribution).

The movements of HDVs are described by well-known Newell's car-following model (Newell, 2002) in Eq. (4) since it is linear and simple, compared with other complicated models such as Wiedemann (1991) or IDM (Treiber et al., 2000) models. Those advance car-following models present complicated nonlinear or nonconvex mathematical features and will make the MPC optimizer unsolvable during a second (a typical time interval used for CAV platooning control). To facilitate the computation, most existing MPC studies will choose to locally linearize the nonconvex function in the optimizer. This approximation will inevitably cause errors, which in turn dampens the advantages of using these complicated car-following models. Besides, these models involve many parameters and make it hard to calibrate parameters in real time. Considering the trade-off mentioned above, this study chooses to use Newell model, which is linear and quite friendly to the computation. What is more, MPC, as a feedback control, has the capability to correct the prediction error (see our discussions later in Section 4). This study will also develop an online adaptive curve learning approach (see Section 6) to update the displacement parameters in the Newell model in real-time. These instruments together ensure that Newell's model work well in our study (see the experiments results in Sections 7.1 and 7.2.)

$$x_{\hat{i}}(k) = x_{\hat{i}-1}(k - t_{\hat{i}}) - d_{\hat{i}}, \quad \hat{i} \in I_H, \quad (4)$$

where $t_{\hat{i}}$ and $d_{\hat{i}}$ represent the time and distance displacement of the HDV $\hat{i}$. It considers the distance-time trajectory of the following vehicle is essentially the same with the leading vehicle except a time and distance displacement. Given this study only needs to know the trajectory of last vehicle ($\hat{m}$) in the HDV segment H , we describe its movement using the aggregated form of Newell's car-following model in Eq. (5),

$$x_{\hat{m}}(k) = x_n(k - T_{\hat{m}}) - D_{\hat{m}}, \quad (5)$$

where $T_{\hat{m}}$ and $D_{\hat{m}}$ represent the aggregated time step and distance displacement of the HDV segment H . Mathematically, $T_{\hat{m}} = \lceil (t_{\hat{m}} + t_{\hat{m}-1} + \dots + t_1) / \tau \rceil$, $D_{\hat{m}} = d_{\hat{m}} + d_{\hat{m}-1} + \dots + d_1$. Using Eq. (5) with given parameters $T_{\hat{m}}$ and $D_{\hat{m}}$, we can predict the trajectory of HDV $\hat{m}$ using the trajectory of CAV n . Nevertheless, we noticed that HDVs' time and distance displacement $T_{\hat{m}}$ and $D_{\hat{m}}$ are highly heterogenous and may vary under different driving environments. It is challenging but important to capture the time-varying HDV behavior for the

PCC-eDriving control accurately. Our study addresses this technical difficulty from two aspects. First, MPC approach in Section 4 itself is closed-loop feedback control, which can correct the HDV behavior prediction errors by updating the new observed system state at each control time step (see more discussions in Section 4). Next, we will develop an adaptive curve learning approach in Section 6 to learn the values of parameters $T_{\bar{m}}$ and $D_{\bar{m}}$ online for human driver's real-time driving behaviors. Our numerical experiments also indicated that Newell's car-following model combined with the adaptive curve learning approach could predict the HDV driving behaviors well in our MPC system.

Furthermore, we consider the CAV control input, speed and the safe distance need to satisfy the physical constraints shown in Eqs. (6), (7) and (8), respectively.

$$a_{\min,i} \leq u_i(k) \leq a_{\max,i}, \quad i \in I_C \quad (6)$$

$$v_{\min} \leq v_i(k) \leq v_{\max}, \quad i \in I_C \quad (7)$$

$$x_{i-1}(k) - x_i(k) \geq L_i + \delta_1 \tau v_i(k) + \delta_2 \tau (v_i(k) - v_{i-1}(k)), \quad i \in I_C \quad (8)$$

Here, $a_{\max,i}$ and $a_{\min,i}$ in Eq. (6) are used to describe the predefined acceleration/ deceleration bounds for CAV i . Similarly, $v_{\min}$ and $v_{\max}$ in Eq. (7) are the pre-specified bounds of driving speed on urban roads. Regarding the driving safety, an adaptive safe distance constraint is enabled by Eq. (8), which considers the safe time headway and the speed difference between leading and following vehicles to ensure the safety and improve the traffic throughputs. Specifically in Eq. (8), parameter $L_i > 0$ represents CAV i 's length plus minimum car-following distance; $\delta_1 \tau$ with $\delta_1 \geq 1$ is the safe time headway; $\delta_2 \tau (v_i(k) - v_{i-1}(k))$ with $\delta_2 \geq 0$ factors the speed difference and enables a smaller (or larger) safe spacing adaptively when the leading vehicle drives faster (or slower) than the following vehicle (i.e., $v_i(k) > v_{i-1}(k)$ vs. $v_i(k) < v_{i-1}(k)$). The idea of Eq. (8) is consistent to GM's car-following model (Chakroborty and Kikuchi, 1999). We expect it can utilize road capacity better than the linear safe constraints (Wu et al., 2020) as well as the highway quadratic safe constraints (Gong et al., 2016). Our experimental results further confirm this merit. It can also achieve collision-free safety performance by using conservative parameters δ_1 and δ_2 in sacrifice of traffic throughputs though.

According to the adaptive safe distance constraints, this study uses the adaptive desired spacing policy in Eq. (9), which will be used in the objective function of the MPC control.

$$s_i(k) = L_i + \delta_1 \tau v_i(k) + \delta_2 \tau (v_i(k) - v_{i-1}(k)) + \delta, \quad (9)$$

In Eq. (9), $s_i(k)$ represents the desired spacing for CAV i at step k . Positive $\delta > 0$ is introduced to make the desired spacing $s_i(k)$ slightly larger than the safe car-following distance in Eq. (8). This makes the control system hardly trigger the activation of the safe distance constraints in Eq. (8) when the platoon is stabilized. It will facilitate the control robustness and smoothness.

Next, we define the CAV i 's spacing and speed errors $\Delta x_i(k)$, $\Delta v_i(k)$ at step k in Eqs. (10) and (11), respectively according to Ghasemi et al., (2013), Wang and Nijmeijer (2015) and Zuo et al., (2021). Please note that the platoon reaches an equilibrium state when the spacing and speed errors $\Delta x_i(k)$, $\Delta v_i(k)$ become zeros for all the CAVs $i \in I$.

$$\Delta x_i(k) = x_{i-1}(k) - x_i(k) - s_i(k), \quad i \in I_C, \quad (10)$$

$$\Delta v_i(k) = v_{i-1}(k) - v_i(k), \quad i \in I_C, \quad (11)$$

Built upon the spacing and speed errors $\Delta x_i(k)$, $\Delta v_i(k)$ in Eqs. (10) and (11), the platoon control dynamics $z(k)$, $z'(k)$ are summarized in Eqs. (12) and (13).

$$z(k) := (\Delta x_1(k), \dots, \Delta x_N(k))^T \in \mathbb{R}^N, \quad (12)$$

$$z'(k) := (\Delta v_1(k), \dots, \Delta v_N(k))^T \in \mathbb{R}^N. \quad (13)$$

Finally, to ensure the stability of the MPC with constraints, we define a terminal constraint at time step $k + P$ as follows in Eq. (14),

$$\Phi_f := (z(k+P) \in \zeta, z'(k+P) \in \zeta'), \quad (14)$$

where P is the MPC prediction horizon. The terminal constraint in Eq. (14) requires the platoon's spacing and speed errors at final time step $k + P$ of the MPC are confined to small domains $\zeta = \left[-\frac{\tau^2}{2} \bar{\Delta}_i P, -\frac{\tau^2}{2} \underline{\Delta}_i P \right]$ and $\zeta' = [-\tau \bar{\Delta}_i P, -\tau \underline{\Delta}_i P]$, respectively, where $\underline{\Delta}_i$ and $\bar{\Delta}_i$ are the lower and upper bound of the CAV control input uncertainty in Eqs. (3). According to Eqs. (3), (6) and (7), $\underline{\Delta}_i = \varepsilon_i v_{\min} + \eta_i (a_{\min,i} - a_{\max,i}) < 0$ and $\bar{\Delta}_i = \varepsilon_i v_{\max} + \eta_i (a_{\max,i} - a_{\min,i}) > 0$. If there is no control uncertainty (i.e., $\Delta u_i(k) = 0$), then the terminal constraint requires the platoon to reach the steady state at final step $k + P$, namely $\zeta = 0$, $\zeta' = 0$.

4. Hybrid MPC system

This section designs the hybrid MPC system to mathematically enable the PCC-eDriving control. Recall that the PCC-eDriving control seeks to instruct the movement of a mixed flow platoon so that it can approach and then pass signalized intersections smoothly. Below we first justify the properness of using the hybrid MPC system and then present our methodology development.

MPC is a closed-loop modern control method. It predicts the system future behaviors in the prediction horizon (e.g., next P time

steps) and then uses a constrained optimization model to find the optimal control law for next P steps but only implements the first-step control law. This prediction and optimization are repeated at each time step so that the MPC can respond to the control and prediction errors timely and smoothly, such as the CAV control uncertainties and the HDV prediction errors in this study. Therefore, it fits the PCC-eDriving control very well. However, as we mentioned in the introduction and problem statement, using a single MPC model is not enough to describe the entire dynamic process of the PCC-eDriving control. Specifically, the platoon will experience three physical states before, during and after the platoon splitting. We formally define them as follows. (i) The entire platoon approaches the intersection before splitting (state q_0). (ii) The first sub-platoon passes the intersection (state q_1). (iii) The second sub-platoon continues to approach intersection avoiding sharp deceleration and reducing red idling time (state q_2). Movements of the first and second sub-platoons under state q_1 and q_2 together split the original platoon. We note that these three states have different control goals and constraints, and thus cannot be described using a single MPC model. Accordingly, three different MPC controllers MPC- q_0 , MPC- q_1 and MPC- q_2 are designed to carry out the platooning control under these three states q_0 , q_1 and q_2 individually. Apart from it, the platoon movement is a continuous and smooth dynamic process, though involving platoon splitting is a discrete event. Then, a fundamental question is how to connect the controllers under these three states properly so that the entire PCC-eDriving control performs continuously and smoothly. To solve this problem, we adopt and design the hybrid system since it has been proved to be a very promising approach and fits the MPC controllers well according to Bemporad et al., (2002); Borrelli et al., (2017). Using Fig. 3, below we explain the design of the hybrid system for this study. Briefly, the hybrid system involves two switching signals σ_0 and σ_1 to trigger the state switching between the states q_0 , q_1, q_2 and their corresponding MPC controllers.

Without loss of generality, we consider the dynamic process starts from state q_0 that the entire mixed flow platoon A approaches the signalized intersection under the guide of MPC- q_0 . Once it enters the communication zone of the traffic signal, the PCC-eDriving control starts a mixed integer non-linear programming optimizer (MINLP) to search for the optimal platoon splitting point within one control interval (1 s). The completion of this step generates switching signal σ_0 , indicating that the platoon A will split into sub-platoons A_1 and A_2 at the optimal point (e.g., $q_0 \xrightarrow{\sigma_0} q_1, q_2$). Then, the PCC-eDriving implements MPC- q_1 and MPC- q_2 , which, respectively guide the movements of sub-platoons A_1 and A_2 so that A_1 can smoothly pass the intersection during the current cycle's green interval, and A_2 can approach the intersection efficiently. Furthermore, when the sub-platoon A_1 passes the intersection, switching signal σ_1 triggers the switching of the state q_1 back to state q_0 (i.e., sub-platoon A_1 restores MPC- q_0). Later, when the sub-platoon A_2 enters the communication zone, if A_2 can entirely pass the intersection in the current cycle's green interval, the PCC-eDriving control switches from MPC- q_2 to MPC- q_1 for A_2 to pass the intersection efficiently. Otherwise, A_2 will split again (2nd split) to form new A_1 and A_2 ; then we repeat the same process above until the resulted sub-platoon A_2 can entirely pass the intersection. In Fig. 3, we denote the states and the sub-platoons after the κ^{th} split as $q_1(\kappa), q_2(\kappa)$ and $A_1(\kappa), A_2(\kappa)$ and assume the splitting process is done after K splits. Note that in the final K^{th} split, sub-platoon $A_2(K-1)$ can entirely pass the intersection and thus state $q_2(K-1)$ only switches to $q_1(K)$ without further splitting. Normally, the maximum number of splits K is not large (e.g., $K \leq 2$). Besides, we should notice that the switching signal σ_1 is triggered when the sub-platoon A_1 physically passes the intersection and thus no mathematical model is needed for switching signal σ_1 . Please note that the states and switching signals in the hybrid system are highly intertwined so that the corresponding MPC controllers and switching signal optimizer should be well designed to ensure the state switching feasibility and hybrid system continuity. Below we formally introduce the mathematical formulations of MPC- q_0 , MPC- q_1 , MPC- q_2 and the signal MINLP- σ_0 in the sequential Sections 4.1–4.4. Then we analyze and prove the switching feasibility and continuity of the hybrid system in Section 4.5.

4.1. MPC controller for state q_0 (MPC- q_0)

This study considers the platoon is approaching the intersection under a platoon-centered car-following control. Accordingly, we develop the MPC- q_0 to conduct CAV trajectory control at any step $k \in \mathbb{Z}_+$ before the platoon enters the communication zone and triggers the switching signal σ_0 . The MPC- q_0 optimizer includes the objective function defined in Eq. (15) subject to the vehicle dynamics and constraints in Eqs. (1)-(14) developed in Section 3. If there is no vehicle ahead leading the platoon, then the first platoon

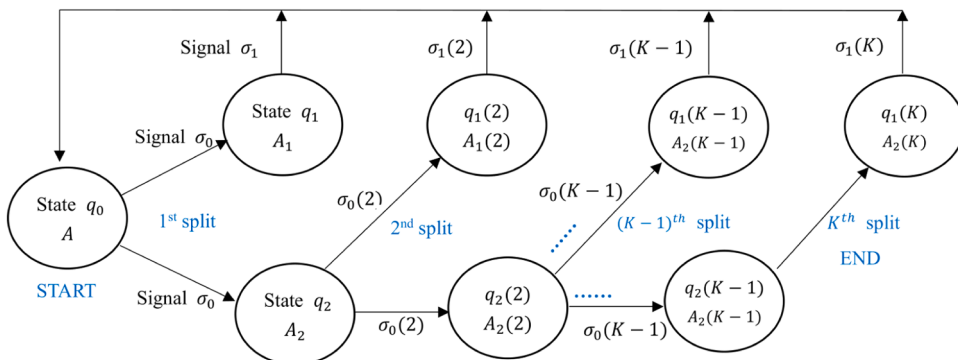


Fig. 3. Hybrid MPC system under normal traffic condition.

CAV $i = 1$ will lead the traffic and keep the desired speed (see [Appendix-I](#) for the desired speed control). The MPC- q_0 predicts the platoon states at any time step $k + p$, for $\forall p = 1, \dots, P$ ($P \in \mathbb{P}$). We simplify step $k + p$ to p hereafter throughout the [Section 4](#) to simplify notation.

MPC- q_0

$$\text{Min } \Gamma(u(p)) = \sum_{p=1}^P \left\{ \frac{1}{2} [z^T(p) Q_z z(p) + (z'(p))^T Q_{z'} z'(p)] + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\} \quad (15)$$

Subject to, for $p \in P$:

Constraints in [Eqs. \(1\)-\(14\)](#), where $u(p) = (u_1(p), \dots, u_N(p))$ is the control input vector at time step $k + p$ for CAVs $i \in I_C$; $Q_z := \text{diag}(\alpha_1, \dots, \alpha_N)$ and $Q_{z'} := \text{diag}(\beta_1, \dots, \beta_N)$ are diagonal matrices; $\alpha_i > 0$ and $\beta_i > 0$ are penalty weights of the spacing and speed errors for each CAV $i \in I_C$.

Next, we will move forward to present the mathematical model for switching signal σ_0 (MINLP- σ_0), MPC- q_1 and MPC- q_2 . Nevertheless, it is noted when the platoon enters the traffic signal communication zone, the current signal phase may be either green or red. These two different traffic signal scenarios will lead to slightly different models and controllers. Therefore, we will first introduce our models under the green scenario in the following [Section 4.2-4.4](#). Then we extend the results to the red scenario in [Remark 2](#).

4.2. Switching signal MINLP- σ_0

This section develops a mixed integer nonlinear programming optimizer (MINLP- σ_0) to find the optimal location to split the platoon A into two sub-platoons A_1 and A_2 while predicting A_1 can pass the intersection in the current green interval. If A_2 cannot pass the intersection in the next green interval, it will further split into two sub-platoons using MINLP- σ_0 again (see details in [Fig. 3](#)). Note that the solution of the MINLP- σ_0 will trigger the switching signal σ_0 and the implementation of MPC- q_1 and MPC- q_2 for the splitting. Therefore, to ensure the control continuity, this MINLP- σ_0 optimizer should factor the platoon's future movements and control during the remaining green interval and the sequential red interval. It should also keep the same constraints as the followed controllers MPC- q_1 and MPC- q_2 to facilitate the switching feasibility. Along with this idea, we develop the MINLP- σ_0 with the thoughts below.

We first introduce binary variable $y_i \in \{0, 1\}$, $\forall i \in \bar{I}_c = \{1, 2, \dots, n, n+1, \dots, N, N+1\}$ to describe the location of the platoon splitting point, which are around CAVs. For example, $y_{i^*} = 1$ represents that the platoon splits immediately ahead of CAV i^* . Please note that the platoon splitting point must locate in front of a CAV because HDVs are not under our control. In addition, to make our control adaptive to the future traffic uncertainties, we regulate the platoon only splits into two sub-platoons every time nearby the intersection and repeat the process until all the platoon vehicles pass the intersection. Then, we have [Eq. \(16\)](#) below to allow only one splitting point in the mixed flow platoon.

$$\sum_{i \in \bar{I}_c} y_i = 1; y_i \in \{0, 1\} \quad (16)$$

Denote i^* as the platoon splitting point, then the last CAV in the sub-platoon A_1 is CAV $i^* - 1$. We thus have [Eq. \(17\)](#) to ensure that the CAV $i^* - 1$ must have passed the intersection at the end of the current green interval at time step $p = \tilde{k}_g$. This constraint will be consistently used by MPC- q_1 . Here the intersection location is set as the longitude coordinate 0. And $M > 0$ is a given large positive number so that [Eq. \(17\)](#) takes effect only for CAV $i^* - 1$.

$$x_{i^*-1}(\tilde{k}_g) \geq -M(1 - y_{i^*}), \quad i \in \bar{I}_c, \quad (17)$$

On the other side, the first CAV in the second sub-platoon A_2 is CAV i^* . [Eq. \(18\)](#) regulates that the CAV i^* (i.e., sub-platoon A_2) cannot pass the intersection until the end of the sequential red interval at time step $p = \tilde{k}_g + k_r$. This constraint will be consistently used by MPC- q_2 .

$$x_{i^*}(\tilde{k}_g + k_r) \leq M(1 - y_{i^*}), \quad i \in \bar{I}_c, \quad (18)$$

We also recognized that the downstream capacity limits the number of vehicles in the sub-platoon A_1 that can pass through the intersection. This observation is captured by [Eq. \(19\)](#), where constant C corresponds to the rest downstream capacity according to current traffic condition.

$$i \cdot y_i \leq C, \quad i \in \bar{I}_c, \quad (19)$$

Next, we consider the future platoon splitting carried by MPC- q_1 and MPC- q_2 will greatly enlarge the inter-vehicle spacing at the splitting point and consequently enlarge the speed difference between two sub-platoons. Accordingly, the MINLP- σ_0 involves this prediction and modifies the measurement of spacing and speed errors by [Eqs. \(20\) and \(21\)](#),

$$\Delta x_i(p) = x_{i-1}(p) - x_i(p) - s_i(p) - y_i \cdot \mathcal{S}, \quad i \in \bar{I}_c, \quad (20)$$

$$\Delta v_i(p) = v_{i-1}(p) - v_i(p) - y_i \cdot \mathcal{S}', \quad i \in \bar{I}_c, \quad (21)$$

where parameters $\mathcal{S}$ and in $\mathcal{S}'$ represent the estimated spacing and speed errors between two sub-platoons A_1 and A_2 . Note that [Eqs.](#)

(20) and (21) are same to the Eqs. (10) and (11) except the spacing and speed errors at the platoon splitting point.

Wrapping up the thoughts above, we summarize the Optimizer MINLP- σ_0 as follows.

MINLP – σ_0

$$\min J(u, y) = J_1(u, y) + \omega_2 J_2(u, y) \quad (22)$$

Subject to, for $p \in P$, $i \in \bar{I}_c$:

Constraints in Eqs. (1)-(9), (12)-(13), (16)-(21), where

$$J_1(u, y) = \sum_{p=1}^{P=\tilde{k}_g+k_r} \left\{ \frac{1}{2} [z^T(p) Q_z z(p) + (z'(p))^T Q_{z'} z'(p)] + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\}$$

$$J_2(u, y) = -\sum_{i \in \bar{I}_c} i \cdot y_i$$

The objective function J in Eq. (22) consider the tradeoff between traffic smoothness and traffic throughputs by tuning weight ω_2 . The first component $J_1(u, y)$ promotes traffic smoothness in the next $P = \tilde{k}_g + k_r$ steps, whereas the second component $J_2(u, y)$ considers maximizing the traffic throughputs. Overall, the MINLP- σ_0 will find the optimal platoon splitting point by predicting the future platoon control and movements during the current green and sequential red intervals, namely the next $P = \tilde{k}_g + k_r$ time steps. It should be noted that the MINLP- σ_0 can always find a feasible solution. The extreme case is the entire platoon A can ($y_{N+1} = 1$) or cannot ($y_1 = 1$) completely pass the intersection during the current green interval. Note that we define a dummy CAV $N+1$ in the CAV set $\bar{I}_c$ to cover the extreme case $y_{N+1} = 1$ when the entire platoon A can completely pass the intersection.

4.3. MPC controller for state q_1 (MPC- q_1)

We next develop the MPC- q_1 for state q_1 . It seeks to instruct the platoon splitting trajectory control of the leading sub-platoon A_1 and guide A_1 to pass the intersection during the current green interval while factoring the platoon system performance and traffic smoothness. Based on the development of the MPC- q_0 and MINLP- σ_0 , MPC- q_1 is given below.

MPC – q_1

$$\text{Min } \Gamma(u) = \sum_{p=1}^P \left\{ \frac{1}{2} [z^T(p) Q_z z(p) + (z'(p))^T Q_{z'} z'(p)] + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\} \quad (23)$$

Subject to, for $i \in \bar{A}_1, p \in P$ where $P = \tilde{k}_g, \tilde{k}_g - 1, \dots, 1$:

Constraints in Eqs. (1)-(13), and

$$x_{i^*-1}(P) \geq 0, \quad (24)$$

where $\bar{A}_1$ represents the CAV set in the sub-platoon A_1 ; the MPC prediction horizon P is a shrinking horizon that decreases at each time step by one as the control proceeds (i.e., $P = \tilde{k}_g, \tilde{k}_g - 1, \dots, 1$). Besides, i^* represents the optimal platoon splitting point determined by the MINLP- σ_0 and thus $i^* - 1$ is the last CAV of A_1 . Accordingly, Eq. (24) ensures that the sub-platoon A_1 passes the intersection within the remaining green interval. It is derived from Eq. (17) in the MINLP- σ_0 .

4.4. MPC controller for state q_2 (MPC- q_2)

This section develops the MPC- q_2 for state q_2 , which instructs sub-platoon A_2 to split from the original platoon A and smoothly approach the intersection for reducing or even avoiding red idling. Once the sub-platoon A_2 reaches the communication zone of the traffic signal, it will trigger the switching signal σ_0 again and calculates the optimal platoon splitting point for itself. If the splitting point is at the end of the sub-platoon A_2 , the entire sub-platoon A_2 can pass the intersection in one green interval. Otherwise, it will further split into two sub-platoons following the same procedure discussed in Fig. 3.

It is noted that the sub-platoon A_2 does not have a leading HDV while it splits from the sub-platoon A_1 . Hence, we consider the first CAV i^* in A_2 as the leading vehicle and generates its eco-driving trajectory as reference for the following vehicles in A_2 . Consequently, our MPC- q_2 consists of two controllers as follows: (i) eco-driving trajectory control for leading CAV i^* ; (ii) following vehicles' eco-driving trajectory optimizer.

MPC- q_2 (i) Eco-driving Trajectory Reference of Leading CAV i^* .

$$\text{Min } F(u_{i^*}) = \sum_{p=1}^P \|u_{i^*}(p-1)\|_2^2 - \omega_3 x_{i^*}(P) \quad (25)$$

Subject to, for $p \in P, P = \tilde{k}_g + k_r, \tilde{k}_g + k_r - 1, \dots, 1$:

CAV i^* dynamics and constraints in Eqs. (1)-(3) and (6)-(8)

$$x_r(P) \leq 0, \quad (26)$$

where P is the shrinking prediction horizon, $P = \tilde{k}_g + k_r, \tilde{k}_g + k_r - 1, \dots, 1$ as the control proceeds. To be noted, if the last HDV in A_1 does not pass the intersection, it will be the leading vehicle for the sub-platoon A_2 and pose a safe car-following constraint to the first CAV i^* in A_2 . Our MPC- q_2 -(i) controller for the leading CAV i^* considered the safe car-following distance constraint in Eq. (8), and thus can handle the scenarios when there exists HDV ahead. Besides, constraint in Eq. (26) illustrates that the leading CAV of A_2 cannot pass the intersection until the red phase runs out. This is derived from Eq. (18) and thus facilitates the switching feasibility (see Section 4.2 in Part II for technical details). The objective function in Eq. (25) co-considers the traffic throughputs and smoothness. On one hand, we intend to improve the traffic throughputs by driving the platoon as close as possible to the intersection (i.e., minimizing the distance between the leading CAV i^* 's and the intersection) at the beginning of the next green interval. On the other hand, we penalize the aggressive acceleration/deceleration for traffic smoothness and reducing energy consumption and emission according to Huang et al. (2018) and USDOE (2022) "USDOE, Driving more efficiently". The two conflicting interests can be tuned by the weight parameter ω_3 in practice.

MPC- q_2 -(ii) Following Vehicles' Trajectory Control.

$$\text{Min } \Gamma(u) = \sum_{p=1}^P \left\{ \frac{1}{2} [z^T(p) Q_z z(p) + (z'(p))^T Q_z' z'(p)] + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\} \quad (27)$$

Subject to, for $i \in \widehat{A}_2 \setminus i^*, p \in P$ where $P = \tilde{k}_g + k_r, \tilde{k}_g + k_r - 1, \dots, 1$:

Constraints in Eqs. (1)-(13), where $\widehat{A}_2$ denotes the CAV set in the sub-platoon A_2 . The MPC- q_2 -(ii) ensures a system optimal car-following control with given leading trajectory provided by the MPC- q_2 -(i). It is similar to MPC- q_0 but with a shrinking prediction horizon P .

Overall, MPC- q_1 and MPC- q_2 follow the optimal splitting decision made by the MINLP- σ_0 . They are well designed to instruct and enable the feasible and smooth platoon splitting trajectory. Below we discuss the extension of this hybrid MPC system from the green scenario to the red scenario in Remark 2.

Remark 1. It should be noted that the sub-platoon A_1 may fail to pass the intersection wholly in the current green interval because some unexpected accidents block the downstream traffic. Our PCC-eDriving model can be extended to handle this extreme scenario. Specifically, MPC- q_1 controller for sub-platoon A_1 can immediately detect this unexpected situation by checking the constraint in Eq. (24), which regulates that the last CAV $i^* - 1$ in A_1 should pass the intersection by the end of the green interval. If Eq. (24) is not satisfied, it means A_1 cannot entirely pass the intersection as expected. To solve this problem, the platoon splitting control (i.e., MINLP- σ_0) can be triggered again to split A_1 into sub-platoons $A_{1,1}$ and $A_{1,2}$ further. This way, the first sub-platoon $A_{1,1}$ can pass the intersection in the current green interval while the sub-platoon $A_{1,2}$ will further split and pass the intersection in the next green intervals using the similar strategy in Fig. 3. This strategy can be repeatedly used as illustrated in Fig. 4 anytime during the PCC-eDriving control when the first sub-platoon finds it cannot pass the intersection within the current green interval.

Remark 2. Extension to Red Scenario: The above hybrid MPC system can be extended to red scenario by doing the following minor changes. Note that under the red scenario, the first sub-platoon A_1 cannot pass the intersection in the current red interval. Then to ensure the traffic safety and efficiency, we add the eco-driving trajectory optimizer (i.e., Eqs. (25) and (26)) for the leading CAV $i = 1$ in A_1 (MPC- q_1). It aims to guide A_1 to smoothly approach the intersection even though it cannot pass it until the current red interval runs out.

Apart from it, we also need to increase the prediction horizon of the MPC- q_1 , MPC- q_2 and the MINLP- σ_0 . More exactly, the MPC- q_1 needs to change the prediction horizon from $P = \tilde{k}_g, \tilde{k}_g - 1, \dots, 1$ under the green scenario to $P = \tilde{k}_r + k_g, \tilde{k}_r + k_g - 1, \dots, 1$ time steps under the red scenario. The similar modification is applied for the MPC- q_2 and MINLP- σ_0 .

5. Distributed optimization algorithm

To implement the hybrid MPC system above, we need to solve the MPC controllers and switching signal optimizer within a control time interval (<1 sec) to sustain the control continuity. The large prediction horizon of these MPC controllers leads to large-scale optimization problems and consequently poses tremendous difficulty in developing efficient numerical solvers. The MINLP- σ_0 optimizer can be efficiently solved by the distributed branch and bound algorithm developed in Androulakis and Floudas (1999). However, for those MPC controllers, this study tested existing well-known distributed optimization algorithms such as distributed primal and dual (Yuan et al., 2011), block coordinate descent (Beck and Tatsuashvili, 2013) and ADMM (Boyd et al., 2011). They cannot converge to a global optimal solution efficiently within one second. This section thus develops a distributed optimization algorithm: active-set based optimal condition decomposition (AS-OCD) to solve the MPC- q_0 , MPC- q_1 and MPC- q_2 efficiently. Below we present the technical

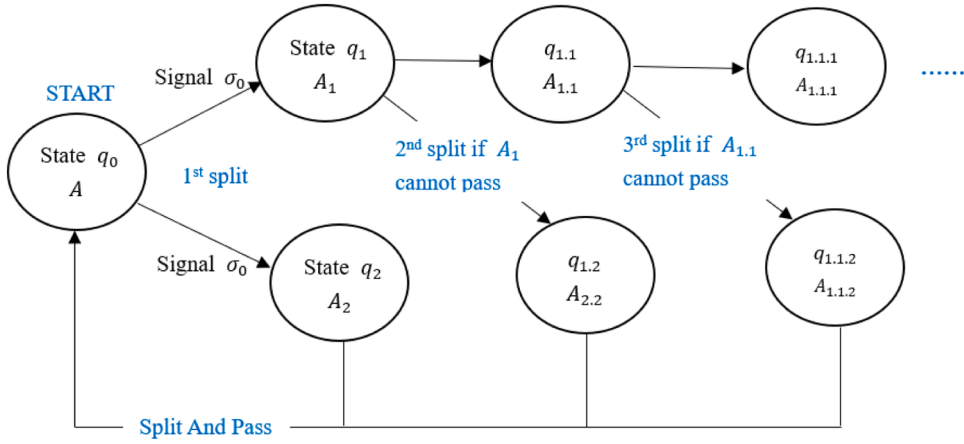


Fig. 4. Repeated Platoon Splitting Control for Accidents.

details.

This study noticed that most of the constraints in the three MPC controllers are inequality constraints and they are inactive¹ under normal traffic conditions. For example, vehicles seldom reach maximum acceleration/deceleration and usually run within speed limits. When the desired spacing and safe distance are designed properly, the safe distance constraints are also inactive at most control time steps. Therefore, physical constraints in Eqs. (6)-(8) are usually inactive. These problem features invoke this study to use active set method (AS) to solve the MPC optimizer efficiently (Nocedal and Wright, 2006). AS method first neglects all the inequality constraints to solve the optimizer and then iteratively adds active constraints back until convergence. It has been widely used particularly in optimization-based control problems (Nak et al. 2023) with extensive redundant inequality constraints in the optimization model. However, the AS method is a centralized algorithm. It does not fit our control problem very well since CAV platoon is a self-organized system subject to frequent platoon topology changes and lack of centralized computation recourse. With this concern, this study further develops the optimal condition decomposition algorithm (OCD) according to Conejo et al. (2002). By integrating AS and OCD approaches into AS-OCD algorithm, we can solve the MPC optimizer efficiently in a distributed manner.

The rest of this section focuses on developing the OCD approach and finally presents the flowchart of the AS-OCD algorithm. To do that, using MPC- q_0 as an example, we first rewrite the MPC controllers in this study by the optimizer F below in Eq. (28) only subject to active set constraints.

$$\text{Optimizer } F : \min \Gamma(u) = \sum_{i=1}^N \left\{ z_i^T Q z_i + (z_i')^T Q' z_i' + \frac{\tau^2}{2} \|u_i\|_2^2 \right\} \quad (28)$$

Subject to l -many $g_{AS}(u_{j-1}, u_j) = 0$, ℓ -many $c_{AS}(u_j) = 0$, where $z_i = (z_i(0), \dots, z_i(P-1))$, $z_i' = (z_i'(0), \dots, z_i'(P-1))$ represent the spacing and speed error control state for CAV i , $\forall i \in I_C$ at each time step during the prediction horizon P ; Similarly, $u_i = (u_i(0), \dots, u_i(P-1))$ represents control input in the active constraints for CAV i , $i \in I_C$. $g_{AS}(u_{j-1}, u_j) = 0$ is an active coupled² constraint such as safe distance constraints in Eq. (8) and $c_{AS}(u_j) = 0$ is an active uncoupled constraint such as acceleration and speed limit constraints in Eqs. (6) and (7). For this optimizer F , we assume there are l -many active coupled constraints $g_{AS}(u_{j-1}, u_j) = 0$ and ℓ -many uncoupled constraints $c_{AS}(u_j) = 0$. Then, the Lagrangian function of the above optimizer F is given by:

$$L_{AS}(u, \eta, \lambda) = \Gamma(u) + \eta^T c_{AS}(u) + \lambda^T g_{AS}(u),$$

with Lagrange multipliers $\eta = [\eta_1, \dots, \eta_\ell]^T$, $\lambda = [\lambda_1, \dots, \lambda_l]^T$.

It is noted that the optimizer F is strictly convex with quadratic objective function subject to linear equality constraints. Then, its Karush-Kuhn-Tucker (KKT) conditions (Boyd et al., 2004) indicate that the global optimal solution of optimizer F can be obtained by solving a system of linear equations. The Newton's method (Nocedal and Wright, 2006) has been widely applied to solve the large-scale system of linear equations efficiently. Mainly, it iteratively computes the searching direction Δu , $-\Delta \eta$, $-\Delta \lambda$ by solving $K \Delta = -f$ in Eq. (29), and then improves the current solution by $u \leftarrow u + \Delta u$, $\eta \leftarrow \eta + \Delta \eta$, $\lambda \leftarrow \lambda + \Delta \lambda$ until converging to an optimal solution.

¹ Given an optimization problem, an inequality constraint $g(x) \geq 0$ is called active at x if $g(x) = 0$ and inactive at x if $g(x) > 0$, whereas equality constraints are always active.

² Coupled constraints refer to the constraints involving more than two agents (two CAVs in this study). Conversely, uncoupled constraints involve only one agent (one CAV).

$$\underbrace{\begin{pmatrix} \nabla^2 L_{AS}(u, \eta, \lambda) & \nabla^T c_{AS}(u) & \nabla^T g_{AS}(u) \\ \nabla c_{AS}(u) & 0 & 0 \\ \nabla g_{AS}(u) & 0 & 0 \end{pmatrix}}_K \underbrace{\begin{pmatrix} \Delta u \\ -\Delta \eta \\ -\Delta \lambda \end{pmatrix}}_{\Delta} = - \underbrace{\begin{pmatrix} \nabla L_{AS}(u, \eta, \lambda) \\ c_{AS}(u) \\ g_{AS}(u) \end{pmatrix}}_f \Leftrightarrow K \Delta = -f \quad (29)$$

Even though Newton's method holds a super linear convergence rate, it cannot be computed in a distributed manner due to the coupling component of $\nabla^2 L_{AS}(u, \sigma, \lambda)$ and $\nabla^T g_{AS}(u)$ in the KKT matrix (denoted as K in Eq. (29)). To adapt the Newton's method to a distributed optimization algorithm, we approximate $\nabla^2 L_{AS}(u, \sigma, \lambda)$ and $\nabla^T g_{AS}(u)$ in the KKT matrix K to block diagonal matrices³ H and A in Eq. (30).

$$H = \text{diag}\left(\frac{\partial^2 L_{AS}(u, \eta, \lambda)}{\partial u_1^2}, \dots, \frac{\partial^2 L_{AS}(u, \eta, \lambda)}{\partial u_N^2}\right); A = \text{diag}\left(\frac{\partial g_1(u)}{\partial u_1}, \dots, \frac{\partial g_N(u)}{\partial u_N}\right) \quad (30)$$

This approximation provides us an approximated KKT matrix $\bar{K}$ in Eq. (31), which are separable in terms of CAV control inputs. Accordingly, we solve the search direction by $\bar{\Delta} = -\bar{K}^{-1}f$. The computation load can be distributed to individual CAVs $\bar{\Delta}_i = -\bar{K}_i^{-1}f_i$ shown in Eq. (31),

$$\underbrace{\begin{pmatrix} \bar{K}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \bar{K}_N \end{pmatrix}}_K \underbrace{\begin{pmatrix} \bar{\Delta}_1 \\ \vdots \\ \bar{\Delta}_N \end{pmatrix}}_{\Delta} = - \underbrace{\begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}}_f \Leftrightarrow \bar{\Delta}_i = -\bar{K}_i^{-1}f_i, \forall i = 1, \dots, N, \quad (31)$$

where $\bar{K}_i$ represents the CAV i 's matrix block in the approximated KKT matrix $\bar{K}$; $\bar{\Delta}_i$ and f_i are CAV i 's search direction and corresponding gradient. The mathematical formulations of $\bar{K}_i$, $\bar{\Delta}_i$, f_i can be seen in Appendix-II. Please note that the approximation of the KKT matrix in this study will not affect the solution optimality and convergence of the OCD algorithms. We will discuss these details in Part II. Besides, note that MPC- q_0 , MPC- q_1 and MPC- q_2 share similar mathematical features (e.g., quadratic objective function, linear constraints, etc.). Thus, this AS-OCD can be applied to solve all three MPC controllers efficiently.

Finally, we present in detail how the AS-OCD algorithm works and distributes the computation loads to different CAVs in Algorithm 1. First of all, each CAV i initializes an empty active set $A_i \in \emptyset$. Then each CAV i will solve its own search direction $\bar{\Delta}_i = -\bar{K}_i^{-1}f_i$ parallelly in a recursive manner. The $\bar{K}_i^{-1}$ and f_i are generated according to the active constraints in the active set A_i . Based upon the calculated search direction, each CAV will update its solution $[u_i \ \eta_i \ \lambda_i]^T \leftarrow [u_i \ \eta_i \ \lambda_i]^T + \bar{\Delta}_i$ where $\bar{\Delta}_i = [u_i \ \eta_i \ \lambda_i]^T$. Then all the CAVs will synchronize to have the overall $\bar{\Delta} = \{\bar{\Delta}_1, \dots, \bar{\Delta}_n\}$. If $\bar{\Delta}$ is small enough, the CAVs will stop solving the search direction $\bar{\Delta}_i$, $\forall i \in I$. Next, each CAV i will check and adds its related infeasible constraints into the active set A_i , while removing the active constraints with negative Lagrange multipliers (e.g., η_i , λ_i). Finally, CAVs will synchronize to have an overall active set $A = \{A_1, \dots, A_n\}$. If the active set A is empty, the current solution is the global optimal solution.

6. Online adaptive curve learning approach

Recall that this study estimates the movements of HDVs by Newell's car-following model in Eq. (5). The two parameters T_m (the aggregated time displacement) and D_m (the aggregated distance displacement) corresponding to human drivers' responses to traffic play a key role in affecting the performance of the MPC controllers. Considering human drivers may have different driving behaviors under different environments, these two parameters are likely to vary accordingly. To make our MPC adaptive to this traffic uncertainty, we prefer accurately estimating these two parameters online at each MPC time step rather than using predefined parameters

Algorithm 1 AS-OCD.

Each CAV i initializes an empty active set $A_i \in \emptyset$.
 Do:
 Repeat until $\bar{\Delta}$ small enough:
 Each CAV i solves $\bar{\Delta}_i = -\bar{K}_i^{-1}f_i$ parallelly, subject to A_i .
 Each CAV i updates solution $\begin{bmatrix} u_i \\ \eta_i \\ \lambda_i \end{bmatrix} \leftarrow \begin{bmatrix} u_i \\ \eta_i \\ \lambda_i \end{bmatrix} + \bar{\Delta}_i$ where $\bar{\Delta}_i = \begin{bmatrix} \Delta u_i \\ \Delta \eta_i \\ \Delta \lambda_i \end{bmatrix}$
 Synchronize $\bar{\Delta} = \{\bar{\Delta}_1, \dots, \bar{\Delta}_n\}$
 Each CAV i checks and adds its infeasible constraints into A_i .
 Each CAV i removes active constraints with negative (η_i, λ_i) from A_i .
 Synchronize $A = \{A_1, \dots, A_n\}$
 Repeat until $A \in \emptyset$.

³ In block matrix A , $g_i(u)$ represents the coupling constraint for CAV i . If $g_i(u)$ is not active, $\frac{\partial g_i(u)}{\partial u_i} = 0$.

observed from historical data. To do that, the curve matching algorithm (Zhang, 1994) is a good candidate method. However, existing study (Gong and Du, 2018) showed that the standard curve matching algorithm requires a relatively long warm-up time and cannot quickly adapt to the frequent traffic variations nearby urban traffic intersections. Hence, it cannot fit this study well.

Motivated by the above view, we develop a new adaptive curve learning algorithm by taking advantage of the unique features of the Newell's car-following model. It seeks to learn and update parameters T_m and D_m at each control step based on the trajectory data, requiring short warm-up time, small computation load and data storage space. Below we illustrate the technical details, including adaptive learning approach and point matching algorithm in the next Sections 6.1 and 6.2, respectively.

6.1. Adaptive learning approach

Newell's car-following model indicates that the trajectory of the following vehicle is essentially the same as the leading vehicle with a time and distance displacement (e.g., T_m , D_m in Fig. 5 (a)). By moving the trajectory data of the HDV $\hat{m}$ at time step $k + 1$ (red dots in Fig. 5 (a)) with optimal temporal and spatial displacement to match the historical trajectory data of the CAV n , we can learn the time and distance displacement $t(k + 1)$, $d(k + 1)$ at the current step $k + 1$. Using the same method, we can find $t(k)$, $d(k)$ at previous step k and then $k - 1, \dots, 0$. However, the learned $t(k)$, $d(k)$ may be inaccurate if the algorithm uses a small set of trajectory data collected by the adjacent CAV. With this concern, the traditional curve matching approach requires large amounts of trajectory data and consequently long warm-up time to guarantee accuracy. To make the learning prompt and accurate, we develop an adaptive learning approach with the main ideas as follows.

First of all, the adaptive learning introduces a time discount factor γ to each trajectory data. Consequently, the curve matching factors most recent trajectory data more, considering they reflect HDVs' current driving characteristics better. Besides, we observed that a larger speed fluctuation reflects more apparent driving behavior variation such as the reaction time delay. Accordingly, this learning approach introduces another weight $w(k)$ to value the importance of each trajectory data for learning the driving behavior variation at step k . Last, to reduce the required computation loads and data storage space, we consider using aggregated $T(k)$, $D(k)$ to describe all the historical time and distance displacement $t(k)$, $d(k)$ learned until time step k . Eqs. (32)-(34) below wrap up the ideas,

$$W(k) = \sum_{j=0}^k w(j), \quad (32)$$

$$T(k) = \sum_{j=0}^k \gamma^{k-j} t(j) w(j) / W(k), \quad (33)$$

$$D(k) = \sum_{j=0}^k \gamma^{k-j} d(j) w(j) / W(k), \quad (34)$$

where $W(k)$ is the aggregated weight of $\{w(0), \dots, w(k)\}$ at step k ; the time discount factor γ satisfies $0 < \gamma \leq 1$. Then, the key question is how to determine values of $w(k)$, $d(k)$ and $t(k)$ accurately. We introduce the point matching algorithm in the next Section 6.2 to determine $d(k)$ and $t(k)$. Below we first present the formulation to determine the weight $w(k)$ in Eq. (35).

$$w(k) = |v_m(k) - v_m(k-1)| \quad (35)$$

Eq. (35) measures the weight of trajectory data at time step k by the absolute velocity change of HDV $\hat{m}$ from step $k - 1$ to k . When

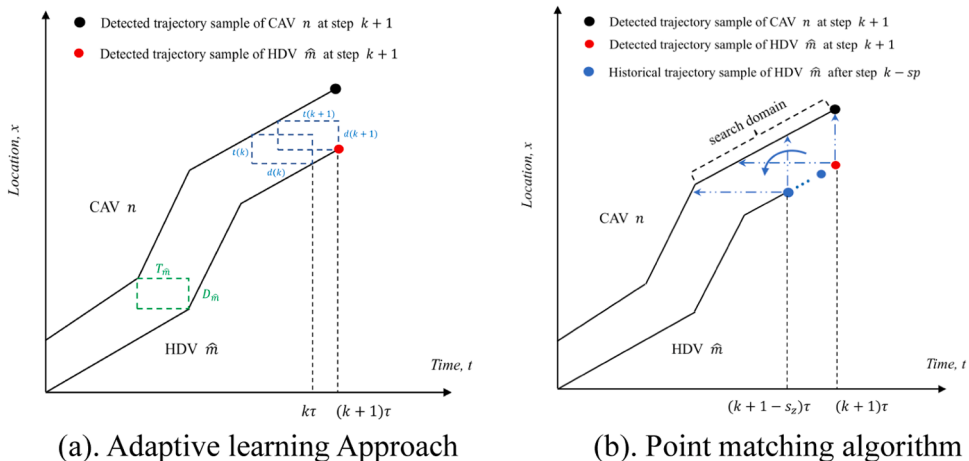


Fig. 5. Online Adaptive Curve Learning Approach.

the speed of HDV $\hat{m}$ does not vary, i.e., $v_{\hat{m}}(k) - v_{\hat{m}}(k-1) = 0$, we have the weight $w(k) = 0$ in the learning. It means that we do not use the trajectory information collected at time step k to update $T(k)$, $D(k)$ because the learned $d(k)$, $t(k)$ do not have much value for update. On the other hand, when the speed of HDV $\hat{m}$ fluctuates apparently at step k , we have $w(k) > 0$ because we consider the learned $d(k)$, $t(k)$ will capture the new driving behavior. This scheme can be explained from the traffic reality. Specifically, we can only clearly observe the following vehicle's reaction when it responds to varying traffic conditions.

However, the learned $T(k)$, $D(k)$ using Eqs. (33)-(34) may have small prediction errors at each step. To avoid error accumulation, this study particularly introduces feedback error gains $e_D(k)$ and $e_T(k)$ obtained from the trajectory prediction errors at step k to further calibrate the time and distance displacement $D(k+1)$ and $T(k+1)$ at the next step.

At last, the learning scheme in Eqs. (33)-(34) is modified into the following Eqs. (36) and (37), which relate $T(k+1)$, $D(k+1)$ to $T(k)$, $D(k)$ and $t(k+1)$, $d(k+1)$.

$$T(k+1) = \frac{W(k)}{W(k) + w(k+1)} \gamma T(k) + \frac{w(k+1)}{W(k) + w(k+1)} t(k+1) + e_D(k) \quad (36)$$

$$D(k+1) = \frac{W(k)}{W(k) + w(k+1)} \gamma D(k) + \frac{w(k+1)}{W(k) + w(k+1)} d(k+1) + e_T(k) \quad (37)$$

As such, our adaptive learning approach in Eqs. (36) and (37) only relies on the most recent trajectory data. It thus can record and track all the historical information with small data storage space and computation load. The error gains $e_D(k)$ and $e_T(k)$ in Eqs. (36) and (37) are calculated based on previous trajectory matching errors. The technical details can be seen in Appendix-III.

6.2. Point matching algorithm

Following the adaptive learning scheme above, we design a point matching algorithm to learn the values of $d(k+1)$, $t(k+1)$ at each time step using recent trajectory data. We use notation $\mathcal{H}_{\hat{m}}(k+1)$ to represent the data set containing newly detected sample trajectory data of HDV $\hat{m}$ at time step $k+1$ (red dot in Fig. 5 (b)) and the recent s_z -many historical trajectory data collected after step $k+1 - s_z$ (blue dots in Fig. 5(b)). Set $\mathcal{H}_{\hat{m}}(k+1)$ includes $|\mathcal{H}_{\hat{m}}| = (s_z + 1)$ -many sample data, represented as $\mathcal{H}_{\hat{m}}(k+1) = \{\rho_{\hat{m}}^l = (x^l, t^l), l = 1, \dots, |\mathcal{H}_{\hat{m}}|\}$. We then define a search domain at time step $k+1$, which includes the sample trajectory data set $\mathcal{C}_n(k+1)$ of CAV n . It includes $|\mathcal{C}_n|$ -many sample data, mathematically $\mathcal{C}_n(k+1) = \{\rho_n^l = (x^l, t^l), l = 1, \dots, |\mathcal{C}_n|\}$. The proposed search domain confines the candidate matching data points, in which we will search for the historical sample data of CAV n that optimally pair with the sample data $\mathcal{H}_{\hat{m}}(k+1)$ of HDV $\hat{m}$.

The point matching algorithm wipes every possible $|\mathcal{H}_{\hat{m}}|$ -many consecutive data points in data set $\mathcal{C}_n(k+1)$. $\mathcal{N} = |\mathcal{C}_n| - |\mathcal{H}_{\hat{m}}|$ represents the number of possible pairs. For any possible pair j , $\forall j = 1, \dots, \mathcal{N}$, a motion vector $\mathcal{J}^j = (-d^j, t^j)$ is searched for the least errors in Eq. (38).

$$\min_{\mathcal{J}^j} (\mathcal{J}^j) = \frac{1}{|\mathcal{H}_{\hat{m}}|} \sum_{l=1}^{|\mathcal{H}_{\hat{m}}|} \|\rho_n^l - \rho_{\hat{m}}^l + \mathcal{J}^j\|^2, \forall j = 1, \dots, \mathcal{N} \quad (38)$$

Obviously, we can have an explicit solution for the unconstrained quadratic optimization problem in Eq. (38), by making $d\mathcal{J}^j(\mathcal{J}^j) / d\mathcal{J}^j = 0$. The explicit optimal solution $\hat{\mathcal{J}}^j$ is given below in Eq. (39).

$$\hat{\mathcal{J}}^j = \frac{1}{|\mathcal{H}_{\hat{m}}|} \sum_{l=1}^{|\mathcal{H}_{\hat{m}}|} (\rho_n^l - \rho_{\hat{m}}^l), \forall j = 1, \dots, \mathcal{N} \quad (39)$$

Then the corresponding optimal objective function is denoted as $\hat{\mathcal{F}}^j$. Accordingly, we will obtain an optimal motion vector $\hat{\mathcal{J}}^j = (-\hat{d}^j, \hat{t}^j)$ with objective value $\hat{\mathcal{F}}^j$ for each possible data pair j , $\forall j = 1, \dots, \mathcal{N}$. We find the best data points pair j^* which minimizes these objective values $\hat{\mathcal{F}}^j(*)$ in Eq. (40).

$$j^* = \operatorname{argmin}\{\hat{\mathcal{F}}^j(*), \forall j = 1, \dots, \mathcal{N}\} \quad (40)$$

Wrapping results above, we obtain the newly learned distance and time displacement $d(k+1)$ and $t(k+1)$ as follows in Eq. (41).

$$d(k+1) = \hat{d}^* ; t(k+1) = \hat{t}^* \quad (41)$$

The pseudo-code of the adaptive curve learning algorithm is given in the Appendix-IV. It is clear that the adaptive curve learning algorithm only needs to store the data $\mathcal{H}_{\hat{m}}(k+1)$, $\mathcal{C}_n(k+1)$ and compute $d(k+1)$, $t(k+1)$ by iterating over $\mathcal{N} = |\mathcal{C}_n| - |\mathcal{H}_{\hat{m}}|$ -many trajectory data points. Hence, the space complexity of this algorithm is $O(|\mathcal{C}_n| + |\mathcal{H}_{\hat{m}}|)$ and the time complexity is $O(|\mathcal{C}_n| - |\mathcal{H}_{\hat{m}}|)$. It should be noted that $|\mathcal{H}_{\hat{m}}|$ represents the number of recent measurement data, usually $|\mathcal{H}_{\hat{m}}| \leq 10$. And $|\mathcal{C}_n|$ is the number of measurement data in the search domain, usually $|\mathcal{C}_n| \leq 30$. The computation efficiency and prediction accuracy of the proposed adaptive curve learning algorithm will be further demonstrated in the simulation experiments in Section 7.

7. Numerical experiments

This section conducts three sets of numerical experiments to verify the performance and merits of our approaches from three aspects. (i) Validate the efficiency of the adaptive curve learning and the AS-OCD algorithms by Experiment-I. (ii) Demonstrate the advantages of the MPC controllers (using MPC- q_0 as example) by involving adaptive desired spacing policy and CAV/HDV uncertainties by Experiment-II. (iii) Validate the merits of the PCC-eDriving control in smoothening traffic, reducing energy consumption and emission by Experiment-III.

Experiment-I and Experiment-II are mainly set up on a sample mixed flow platoon shown in Fig. 1. It is a platoon with 10 vehicles including 7 CAVs ($n = 7$): a leading vehicle $\hat{0}$ followed by 4 CAVs, 3 HDVs and then 4 CAVs. The leading vehicle follows the trajectory collected on Lankershim Boulevard in the Universal City neighborhood of Los Angeles, CA, on June 16, 2005 given in NGSIM dataset. The initial CAV speed and spacing are set as 10 m/s and 15 m. Experiment-III uses a longer mixed flow platoon, which involves an HDV platoon with 3 vehicles sandwiched in between an upstream CAV platoon with 6 vehicles and a downstream CAV platoon with 7 vehicles. CAVs' initial speed and spacing are set as 20 m/s and 30 m. The used physical parameters are given in Table 1.

In Table 1, the spacing and speed penalty weight α_i and β_i are different in MPC controllers and the switching signal MINLP- σ_0 . The MPC controllers assigns different weights to different spacing and speed errors for platoon's good stability performance. However, this setting is not proper for the MINLP- σ_0 , which aims to determine the platoon splitting point. Specifically, each spacing and speed error in the platoon should be assigned equal penalty weights, to make sure every inter-vehicle spacing has equal opportunity to become splitting point.

7.1. Solution approach performance

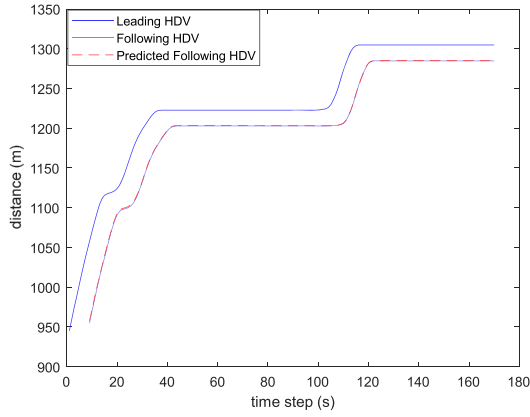
We first validate the computation performance of the adaptive curve learning algorithm, the AS-OCD and the DBB algorithms through Experiment-I. To do that, we, respectively used Newell's car-following model and intelligent driver model (IDM) (Treiber et al., 2000) to simulate the HDV driving behaviors, and then applied online adaptive curve learning approach with Newell's car-following model to predict the HDV trajectories in the MPC. Fig. 6 (a) and (b) present the predicted HDV trajectory by the MPC using Newell's model to simulate field HDV trajectory, while Fig. 6 (c) and (d) present the prediction results using the IDM model to do the simulation.

The results in Fig. 6 (a) and (b) indicate the adaptive curve learning algorithm takes less than 20 s for warm-up. After that, the MPC (combining Newell's model and the adaptive curve learning algorithm) accurately predicted the HDV's trajectory curve, which was simulated by Newell's car-following model (i.e., the predicted trajectory curve is almost coincided with the simulated trajectory in both location and speed). The average spacing prediction error is 0.1255 (m). More importantly, Fig. 6 (c) and (d) demonstrate that the MPC also well predicted the HDV's trajectory curve, which was simulated by IDM model. The average spacing prediction error is 0.3983 (m). Although the prediction accuracy is worse than that in Fig. 6 (a) and (b), these prediction errors will be further corrected by MPC closed-loop control and thus do not harm the control performance very much. Moreover, the average computation time at each time step is less than 0.002 sec. Therefore, we conclude that Newell's model combined with the adaptive curve learning algorithm fits this study very well to predict HDVs' trajectories.

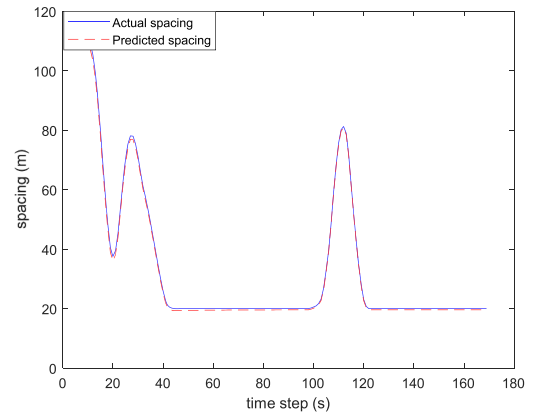
Next, we tested the performance of the AS-OCD and DBB algorithms to solve MPC- q_0 controllers and MINLP- σ_0 by setting the MPC

Table 1
Parameter setting.

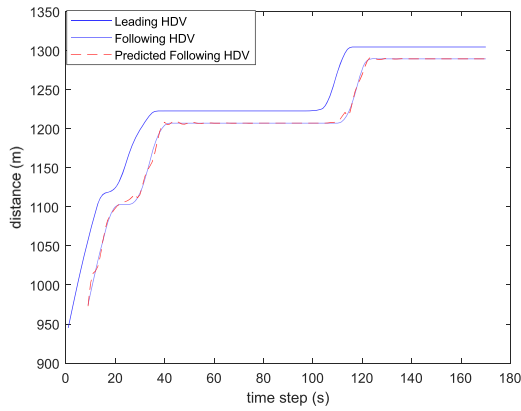
Parameters	Values
Sample time interval τ	1 (s)
Vehicle length L_i	3 (m)
Maximum deceleration $a_{min,i}$	-5 (m/s^2)
Maximum acceleration $a_{max,i}$	4 (m/s^2)
Minimum speed v_{min}	0 (m/s)
Maximum speed v_{max}	22 (m/s)
Aerodynamic coefficient e_i	[0.2, 0.3] (Montanaro et al, 2020)
Powertrain lag σ_i	[0.1, 0.2] (Automobile drag coefficient, 2021)
Safety parameter δ_1	1
Safety parameter δ_2	0.5
Desired spacing parameter δ	5 (m)
Penalty weight ω_1	1
Discount factor γ	0.99
	MPC- q_0 , MPC- q_1 , MPC- q_2
Penalty weight α_i	$0.3*N^2 - 0.6*(N + 1 - i)$ (Gong et al., 2016)
Penalty weight β_i	$0.4*N^2 - 1.2*(N + 1 - i)$ (Gong et al., 2016)
	Switching signal MINLP- σ_0
	$N^2 * p^2$
Penalty weight ω_2	$0.3N^2$
Penalty weight α_i	$0.4N^2$
Penalty weight β_i	$0.4N^2$
Splitting spacing difference $\mathcal{D}$	200 (m)
Splitting speed difference $\mathcal{D}'$	10 (m/s)



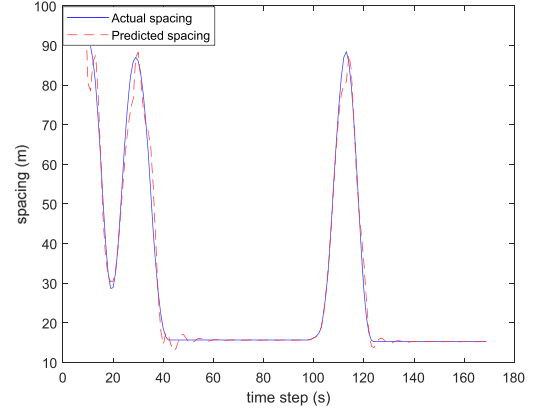
(a). distance – time (Newell - Newell)



(b). spacing – time (Newell-Newell)



(c). distance – time (Newell -IDM)



(d). spacing – time (Newell-IDM)

Fig. 6. Comparison of the predicted and actual trajectory of HDV simulated by Newell and IDM models.

prediction horizons $P = \{20, 30, 40, 50, 60\}$ for different mixed-flow platoons $n = \{7, 10, 13\}$. The mixed flow platoons were formed based on the sample platoon (i.e., 4 CAVs + 3 HDVs + 4 CAVs) in Experiment-I with $n = 7$ CAVs. More CAVs were added to the sample platoon at the end of the platoon to form longer platoons with $n = 10$ and $n = 13$. The results are shown in Table 2. Note that we also tested the longer platoon by adding more HDVs and noticed that it did not affect the computation performance of AS-OCD and DBB algorithms very much. Thus, these results were not presented in Table 2.

The results in Table 2 demonstrate that the computation time increases as the platoon size or prediction horizon P increases. However, the computation time is always less than 0.3 s, which satisfies the computation requirement (< 1 s, the control interval) of the MPC. Therefore, we conclude that the AS-OCD algorithm can efficiently solve the MPC- q_0 controllers with long prediction horizons for different mixed-flow platoons. Similar computation performances were observed when we used the AS-OCD algorithm to solve other MPC controllers such as MPC- q_1 and MPC- q_2 . Besides, Table 2 also shows that the DBB algorithm can solve the MINLP- σ_0 optimizer with platoon size $n = 13$ within 0.3 s, which also meets the computation efficiency demand and can ensure the control smoothness and continuity of the hybrid MPC system.

Table 2

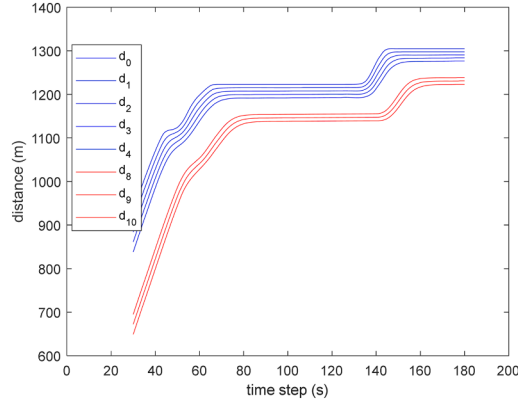
AS-OCD and DBB algorithms computation performance (sec).

AS-OCD algorithm MPC horizon	MPC- q_0 $n = 7$		MPC- q_0 $n = 10$		MPC- q_0 $n = 13$		DBB algorithm MINLP- σ_0 $n = 13$	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
$P = 20$	0.0412	2.82e-05	0.0433	1.82e-05	0.0432	1.61e-05	0.1122	3.36e-05
$P = 30$	0.0601	5.56e-05	0.0630	2.12e-05	0.0992	2.93e-05	0.1338	5.29e-05
$P = 40$	0.0810	5.15e-05	0.0891	1.16e-05	0.1463	7.33e-06	0.1599	1.76e-05
$P = 50$	0.1037	5.07e-05	0.1529	1.04e-05	0.1838	1.90e-05	0.2104	7.09e-05
$P = 60$	0.2122	2.44e-05	0.2464	9.65e-05	0.2521	5.08e-05	0.2688	2.08e-06

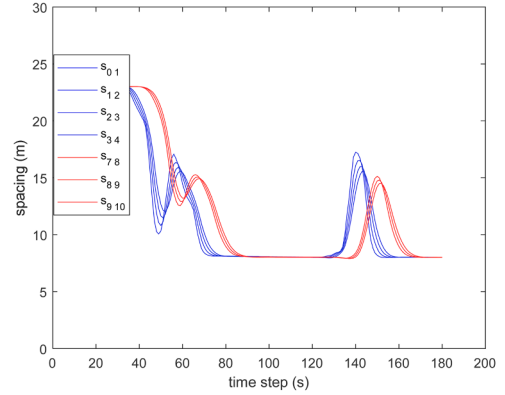
7.2. Traffic flow smoothness and throughputs

Next, this study demonstrates the benefits of improving the platoon stream smoothness, throughputs and efficiency from involving the adaptive desired spacing policy and CAV/HDV uncertainties in the optimizer of the MPC controllers. To do that, we use MPC- q_0 as example and compare its platooning control performance to the CACC control (Chen et al., 2019) and other platooning control. The results are shown in Figs. 7, 8 and 9.

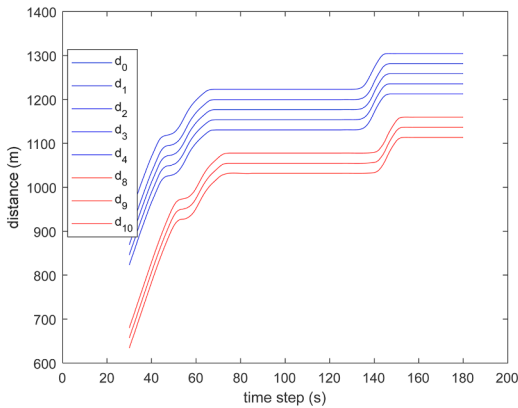
Specifically, Fig. 7 (a_1 , a_2), (b_1, b_2) and (c_1, c_2), respectively illustrate the distance and spacing variations in the experiments under the MPC with adaptive spacing policy (i.e., MPC- q_0), MPC with constant desired spacing policies (i.e., MPC- $\bar{q}_0$) and the CACC. Fig. 7



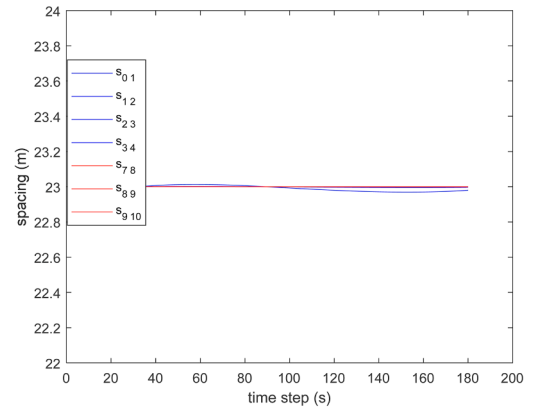
(a_1) distance-time diagram under MPC- q_0



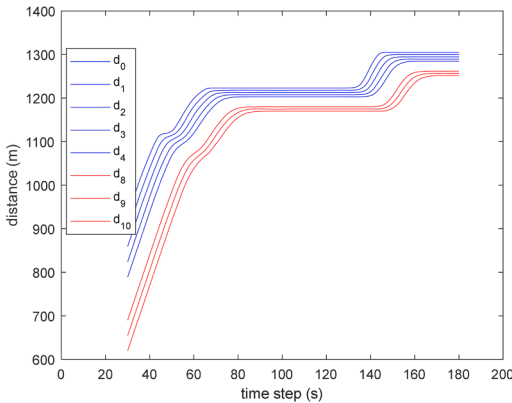
(a_2) spacing-time diagram under MPC- q_0



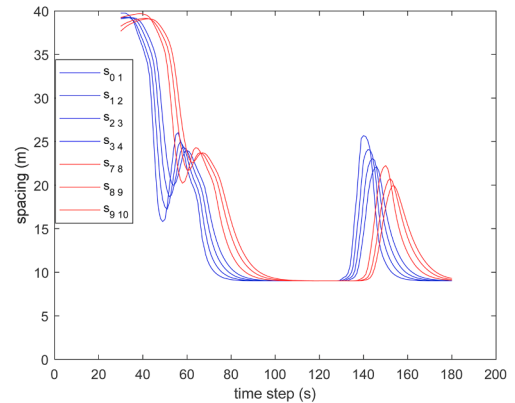
(b_1) distance-time diagram under MPC- $\bar{q}_0$



(b_2) spacing-time diagram under MPC- $\bar{q}_0$

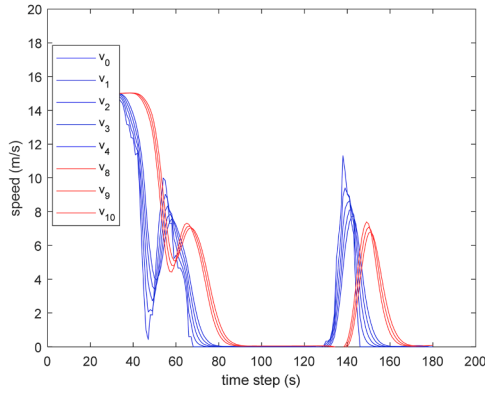
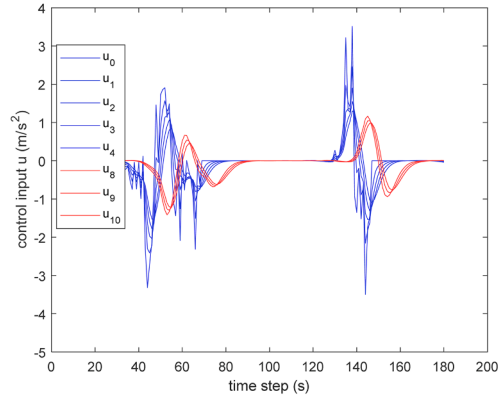
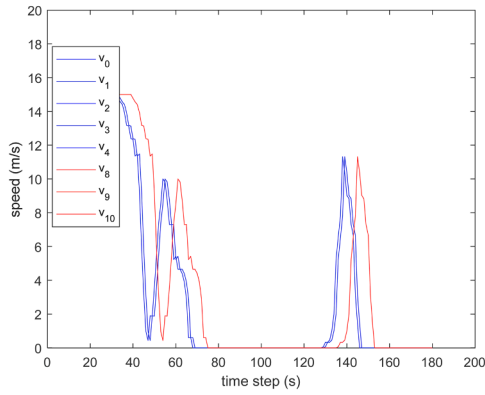
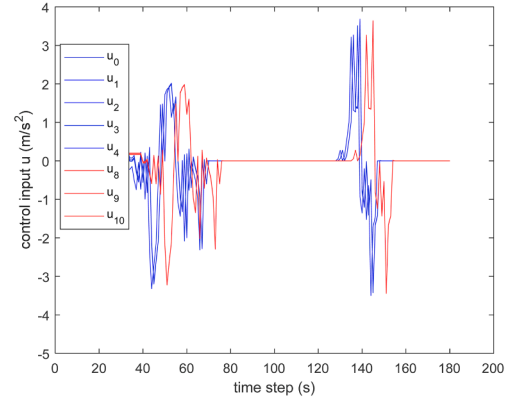
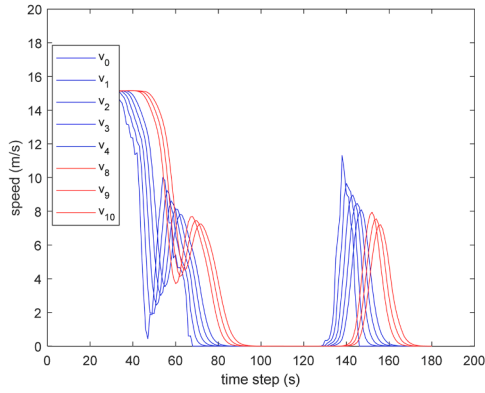
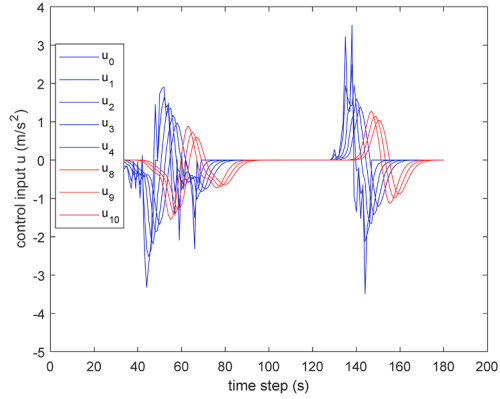


(c_1) distance-time diagram under CACC



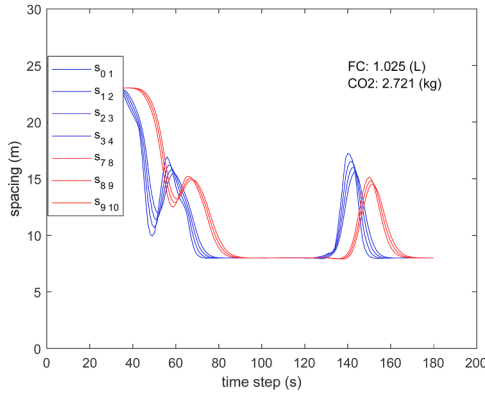
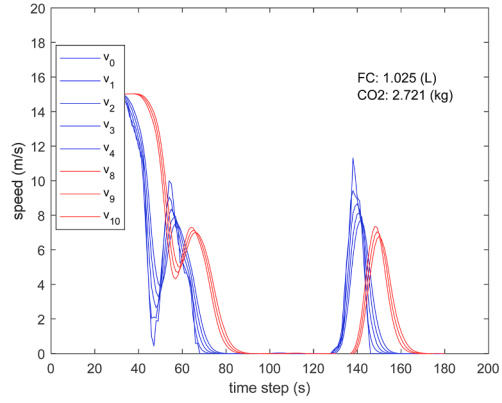
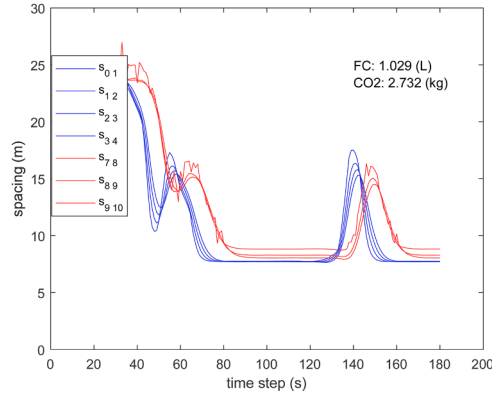
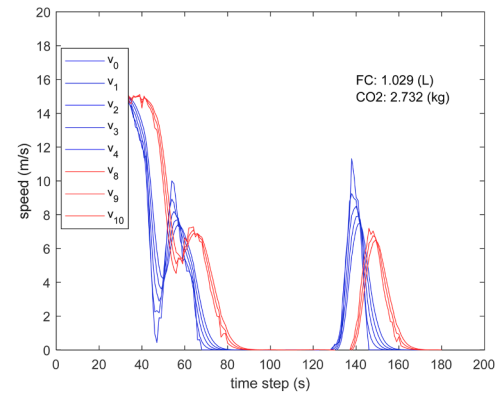
(c_2) spacing-time diagram under CACC

Fig. 7. Distance-time and spacing-time trajectories (MPC- q_0 , MPC- $\bar{q}_0$, CACC).

(a₁) speed-time diagram under MPC- q_0 (a₂) control input-time diagram under MPC- q_0 (b₁) speed-time diagram under MPC- $\bar{q}_0$ (b₂) control input-time diagram under MPC- $\bar{q}_0$ (c₁) speed-time diagram under CACC(c₂) control input-time under CACCFig. 8. Speed-time and control input-time trajectories (MPC- q_0 , MPC- $\bar{q}_0$, CACC).

(a₂, b₂) together indicate that MPC- q_0 allows variant spacing, of which the spacing varies from 8 m to 23 m with the minimum equal to 8 m (including vehicle length) when vehicles stop, whereas the MPC- $\bar{q}_0$ applying constant spacing around 23 m to ensure the safety. As such, Fig. 7 (a₁, b₁) demonstrates that the adaptive desired spacing policy reduces the inter-vehicle spacing compared to the constant desired spacing policy and it improves the road capacity by about 44%. Although the CACC controller in Fig. 7 (c₁, c₂) also enables adaptive spacing, Fig. 7 (a₂, c₂) show that the spacing under CACC control vary more violently and need more time to get stabilized than the MPC- q_0 . Therefore, we conclude that the MPC- q_0 outperforms the CACC in traffic smoothness.

Fig. 8 demonstrates the speed and control input variations under the corresponding control schemes. It is observed that the speed and control input fluctuations under the MPC- q_0 in Fig. 8 (a₁, a₂) are milder than the MPC- $\bar{q}_0$ with constant desired spacing policy in Fig. 8 (b₁, b₂) and the CACC controller in Fig. 8 (c₁, c₂). The mild speed and control input variations can improve driving comfort, and

(a₁) spacing-time diagram under MPC- q_0 -N(a₂) speed-time diagram under MPC- q_0 -N(b₁) spacing-time diagram under MPC- q_0 -I(b₂) speed-time diagram under MPC- q_0 -IFig. 9. Spacing-time and speed-time diagrams, respectively under MPC- q_0 -N, MPC- q_0 -I, MPC- $\hat{q}_0$, MPC- $\tilde{q}_0$.

more importantly lead to energy consumption and emission reductions. The results in Figs. 7 and 8 indicate the MPC- q_0 with adaptive desired spacing policy can effectively mitigate traffic oscillations and improve smoothness and throughputs, compared with CACC control and the MPC using constant desired spacing policy.

We next tested how the online adaptive curve learning helps the MPC to improve traffic smoothness and efficiency. Specifically, Fig. 9 (a₁, a₂) and Fig. 9 (b₁, b₂) illustrate the CAV spacing and speed variations under the MPC- q_0 control, where we, respectively use Newell's car-following model and IDM model to simulate HDV trajectories. To differentiate them, we denote the controller MPC- q_0 -N for Fig. 9 (a₁, a₂) and the controller MPC- q_0 -I for Fig. 9 (b₁, b₂). We can observe that MPC- q_0 -I controller performs as well as MPC- q_0 -N for smoothing CAV's trajectories, except for CAV $i = 8$. We can also observe that the trajectory of CAV $i = 8$ under MPC- q_0 -I fluctuates more violently than other CAVs. A key reason is that CAV $i = 8$ is the first CAV following the HDV platoon segment and suffers from the prediction errors of HDV driving behavior. Nevertheless, the traffic oscillation behind CAV $i = 8$ (e.g., CAV $i = 9, 10$) is mitigated to the minimum extent. Accordingly, the energy consumption under MPC- q_0 -I (shown in Fig. 9 (b₁, b₂)) only increased about 0.4% compared with that under MPC- q_0 -N in Fig. 9 (a₁, a₂). Overall, we confirm the significance of using online adaptive curve learning to smoothen traffic using this MPC.

Fig. 9 (c₁, c₂) and (d₁, d₂) further illustrate the CAV spacing and speed variations under MPC- $\hat{q}_0$ control ignoring CAV control uncertainty and MPC- $\tilde{q}_0$ control without integrating online adaptive learning. Both MPC- $\hat{q}_0$ and MPC- $\tilde{q}_0$ used Newell's car-following model to simulate the HDV driving behaviors. Comparing Fig. 9 (c₁, c₂) and (d₁, d₂) to Fig. 9 (a₁, a₂), we can demonstrate the importance of involving CAV control uncertainties and online adaptive curve learning approach in the controllers. Specifically, vehicle trajectories under MPC- q_0 -N in Fig. 9 (a₁, a₂) are smoother than those under MPC- $\hat{q}_0$ in Fig. 9 (c₁, c₂) and those under MPC- $\tilde{q}_0$ in Fig. 9 (d₁, d₂). Therefore, we conclude that MPC- $\hat{q}_0$ and MPC- $\tilde{q}_0$ suffers from ignoring the errors resulting from the uncertainties of CAV and HDV dynamics. Even though these errors can be corrected by the feedback controller and the traffic fluctuation of the platoon can be

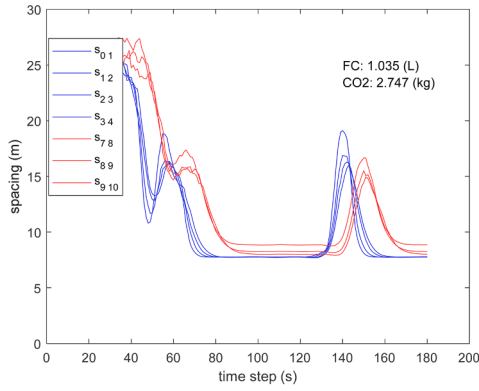
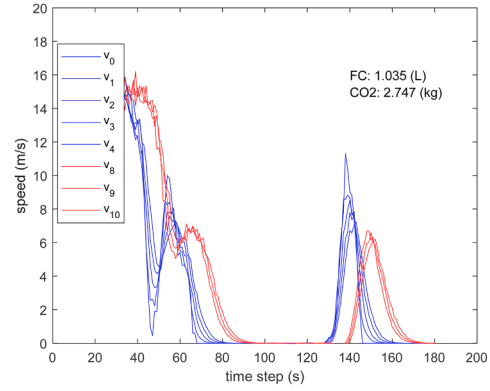
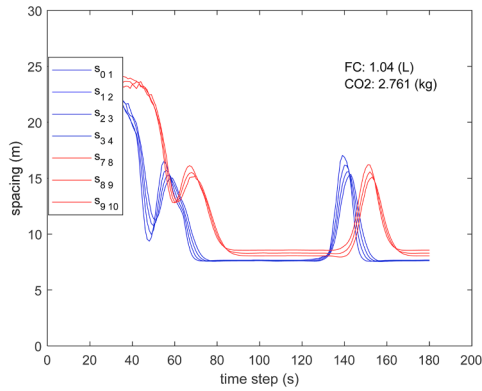
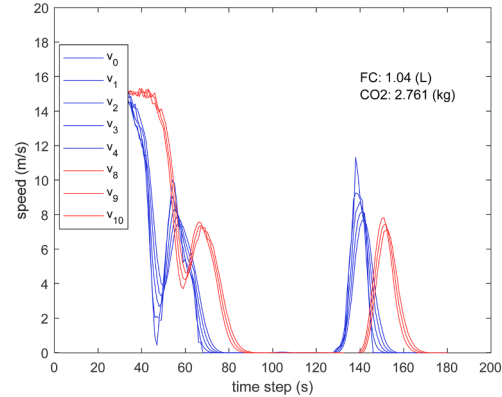
(c₁) spacing- time diagram under MPC- $\hat{q}_0$ (c₂) speed-time diagram under MPC- $\hat{q}_0$ (d₁) spacing-time diagram under MPC- $\tilde{q}_0$ (d₂) speed-time diagram under MPC- $\tilde{q}_0$

Fig. 9. (continued).

bounded around the steady-state (Input-to-State stability) shown in Fig. 9 (c₁, c₂) and (d₁, d₂), it indeed wastes fuel consumption and emission by 1% in Fig. 9 (c₁, c₂) and 1.5% in Fig. 9 (d₁, d₂) as compared with results in Fig. 9 (a₁, a₂) that factor the uncertainties.⁴ Therefore, those observations necessitate factoring in the uncertainties of CAV and HDV dynamics in the MPC.

7.3. Performance of the PCC-eDriving control

This study finally validates the traffic performance of the PCC-eDriving control for guiding a platoon to pass the signalized intersection. The experiments compared the PCC-eDriving control with an existing CACC controller and a PAVSOS controller developed in Faraj et al. (2017) under both green and red scenarios. Fig. 10 (a₁, b₁, c₁) provide the results under green scenario, where we set the remaining green interval as $\tilde{k}_g = 25$ s, when the mixed flow platoon arrives at the communication zone ($r = 300$ m away from the intersection). Fig. 10 (a₂, b₂, c₂) demonstrate the results under the red scenario, where we set there remains red interval as $\tilde{k}_r = 25$ s. Both green and red scenarios use the same traffic signal setting: $k_r = k_g = 40$ s. The experiments use the VT-micro model to estimate the fuel consumptions and emissions (Ahn et al., 2002).

Fig. 10 (a₁, a₂) shows that the PCC-eDriving control can improve the traffic smoothness, save energy consumption and emission compared with the performance of the CACC control in Fig. 10 (b₁, b₂) and the PAVSOS control in Fig. 10 (c₁, c₂) under both green and red scenarios. More exactly, under the green scenario, Fig. 10 (a₁) shows that the platoon under the PCC-eDriving control split into two sub-platoons, and then they sequentially passed the intersection without sharp deceleration. It is noted that the latter sub-platoon decelerated gently and smoothly to avoid red idling and save fuel consumption and emission. The PCC-eDriving control saved approximately 59% fuel consumption and CO₂ emissions for the entire trip, compared with CACC control in Fig. 10 (b₁), which led to sharp deceleration at the intersection and consequently consumed more fuel and produced more CO₂ emission. Compared with the PAVSOS control in Fig. 10 (c₁), our results showed that the PCC-eDriving control saved about 5.4% fuel consumption and CO₂ emission, even though the PAVSOS control also avoided red idling. The PCC-eDriving control outperformed the PAVSOS control since

⁴ The fuel consumption and emission are estimated using VT-micro model developed in Ahn et al., (2002).

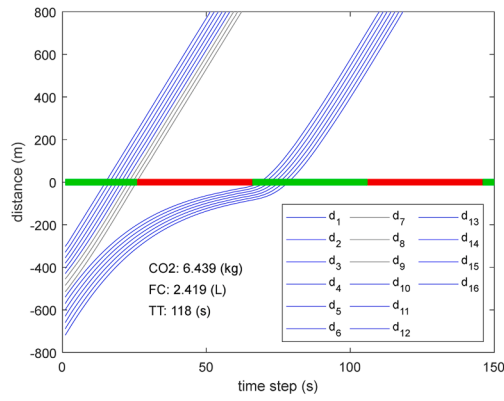
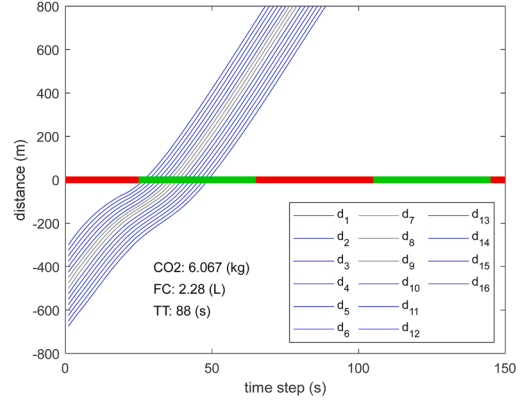
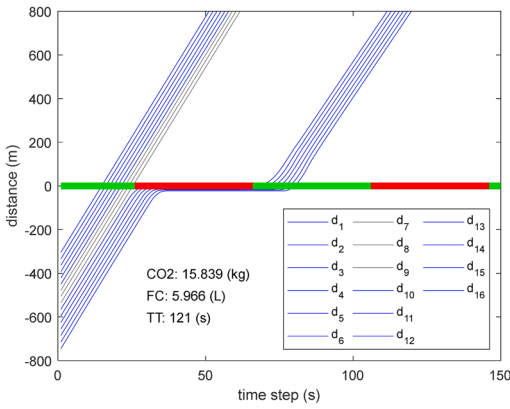
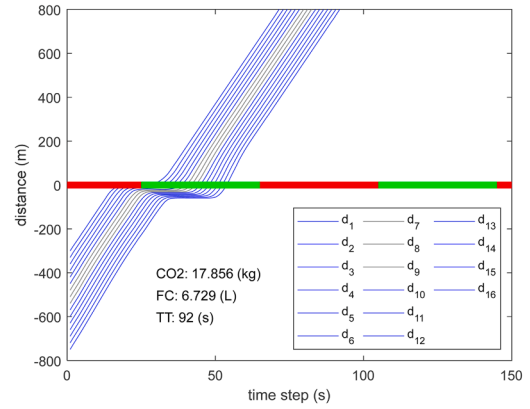
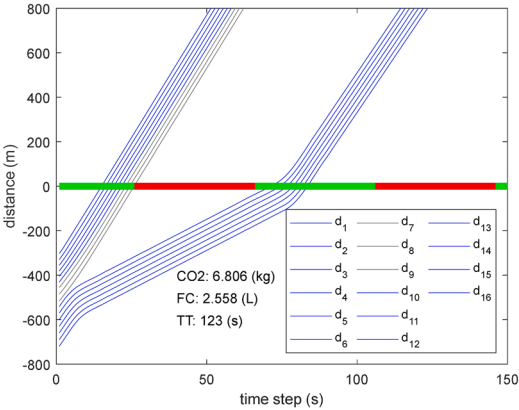
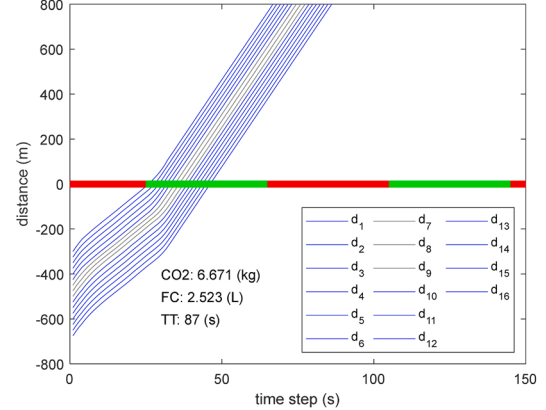
(a₁) PCC-eDriving ($\tilde{k}_g = 25s$)(a₂) PCC-eDriving ($\tilde{k}_r = 25s$)(b₁) CACC ($\tilde{k}_g = 25s$)(b₂) CACC ($\tilde{k}_r = 25s$)(c₁) PAVSOS ($\tilde{k}_g = 25s$)(c₂) PAVSOS ($\tilde{k}_g = 25s$)

Fig. 10. Distance-time trajectory (PCC-eDriving, CACC and PAVSOS).

it generated smoother vehicle trajectories. Under the red scenario, Fig. 10 (a₂) shows that the platoon under the PCC-eDriving control decelerated in advance and passed the intersection in one green interval without sudden deceleration or stop-and-go motions. It saved about 66% and 9.3% fuel consumption and emission, respectively compared with CACC control in Fig. 10 (b₂) and PAVSOS control in Fig. 10 (c₂). It is interesting to observe that the fuel consumption saving under the red scenario is more significant than that under the green scenario. This is because more vehicles in the platoon need to decelerate under the red scenario. Wrapping above, the PCC-eDriving control in this study can significantly improve the traffic smoothness and efficiency compared with existing CACC and eco-driving control.

8. Conclusion

Existing studies mainly use open-loop trajectory planning, speed advisory, or responsive control to develop eco-driving strategies. Although showing improved performance in saving energy and reducing emission, those approaches lack robustness to uncertainty occurring in the traffic environment. On the other hand, closed-loop (feedback) platoon-centered control can sustain better stream traffic smoothness and efficiency, but leads to significant modeling and computation challenges. Motivated by this view, this study develops a system optimal platoon-centered control for eco-driving (PCC-eDriving) to address these gaps, aiming to guide a mixed flow platoon passing the signalized intersections smoothly and efficiently while factoring uncertainties. Mathematically, we design a hybrid MPC system involving three MPC controllers and one mixed integer nonlinear programming optimizer to enable this eco-driving control. It mathematically captures and instructs the dynamic control process for a platoon to approach an intersection, split into sub-platoons and then sequentially pass the intersection during different green intervals. Besides, the MPC controllers of the hybrid system employ adaptive desired spacing policy and online learning algorithm to factor the uncertainties resulting from CAV control and HDV variant driving behavior. An active-set based optimal condition decomposition approach (AS-OCD) is developed to solve large-scale MPC optimizers efficiently in a decentralized manner.

Our simulated experiments show that the online adaptive curve learning approach could accurately predict the HDV driving behavior, and the AS-OCD algorithm can efficiently solve the large-scale MPC optimizers in 0.2 s. Moreover, it is observed that the PCC-eDriving control can significantly improve traffic smoothness and efficiency compared to existing CACC and MPC controllers in literature. Finally, our numerical experiments demonstrate that the PCC-eDriving control could save approximately 50% and 7% fuel consumption and emission compared to an existing CACC control and eco-driving strategy.

There are several interesting future topics motivated by this study. One of them is considering the interaction between the vehicle platooning control and traffic signal control. Accordingly, the traffic signal setup will not be predefined but adaptive to the upcoming traffic flow. This extension may further smoothen traffic and reduce energy consumption at the system level. However, it will bring in new challenges, such as the coordination between the platoon trajectory control and traffic signal control. Furthermore, we can extend this study to consider left or right-turn intents at the intersection. This extension may raise new challenges associated with platoon diverging and latitude trajectory control. They will significantly complicate the mathematical modeling and algorithm design in this study. We propose to address these extensions in our future study.

Author statements

The authors confirm the contributions to the paper as follows. Dr. Du initiated the research idea and led the main methodologies development. Under the supervision of Dr. Du, Ph.D. student Hanyu Zhang contributes to the development of the technical details in PCC-eDriving with hybrid MPC system modeling, control feasibility and stability proofs, distributed optimization solution approach, curve learning algorithm and numerical experiments.

Acknowledgments

This research is partially funded by National Science Foundation, award CMMI 1901994.

Appendix

Appendix-I

We present the reference speed controller for the leading CAV $i = 1$ as follows in Eq. (42):

$$\mathbf{Min} \Gamma(u_1) = \sum_{p=1}^P \left\{ \tau \|v_1(p) - v_r\|_Q^2 + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\} \quad (42)$$

Subject to for $i = 1, \quad p \in P$:

Constraints in Equations (1) – (3) and (6) – (7)

where v_r represents the reference speed for the leading CAV $i = 1$. Q is the positive penalty diagonal matrix for the speed errors.

Appendix-II

The mathematical formulations of $\bar{K}_i$, $\bar{\Delta}_i$, f_i are presented as follows according to Eqs. (29) and (31).

$$\bar{K}_i = \begin{bmatrix} \frac{\partial^2 L_{AS}(u, \eta, \lambda)}{\partial u_i^2} & \frac{\partial^T c_i(u_i)}{\partial u_i} & \frac{\partial^T g_i(u_{i-1}, u_i)}{\partial u_i} \\ \frac{\partial c_i(u_i)}{\partial u_i} & 0 & 0 \\ \frac{\partial g_i(u_{i-1}, u_i)}{\partial u_i} & 0 & 0 \end{bmatrix}; \bar{\Delta}_i = \begin{bmatrix} \Delta u_i \\ -\Delta \eta_i \\ -\Delta \lambda_i \end{bmatrix}; f_i = \begin{bmatrix} \frac{\partial L_{AS}(u, \eta, \lambda)}{\partial u_i} \\ c_i(u_i) \\ g_i(u_{i-1}, u_i) \end{bmatrix}$$

Appendix-III

The error gains $e_D(k)$ and $e_T(k)$ in Eqs. (36) and (37) are calculated based on trajectory matching errors $e(k)$ using the following Eqs. (43)–(45),

$$e(k) = \tilde{H}(k) - H(k), \quad (43)$$

$$e_D(k) = a_D e(k), \quad (44)$$

$$e_T(k) = \frac{a_T e(k)}{\max(v_{\hat{m}}(k), 1)}, \quad (45)$$

where a_D and a_T are gain coefficients for distance error gain $e_D(k)$ and time error gain $e_T(k)$, respectively; $v_{\hat{m}}(k)$ is the speed of HDV $\hat{m}$ at step k ; $H(k)$ is the actual trajectory point of a HDV $\hat{m}$ at step k ; $\tilde{H}(k)$ is the predicted trajectory point of a HDV $\hat{m}$ at step k using the historical trajectory of the CAV n with the distance and time displacement $D(k)$ and $T(k)$ at step k ; $e(k)$ represents the trajectory matching error of the following HDV $\hat{m}$ at step k . Below we explain the technical details to develop Eqs. (43)–(45).

We consider two trajectories $H(k)$ for HDV $\hat{m}$ and $C(k)$ for CAV n . Then we pick a point A: $(k, H(k))$ on $H(k)$. According to the learned displacement $T(k)$ and $D(k)$ at time step k , we can match the point of $(k, H(k))$ on HDV $\hat{m}$ trajectory to the point of B: $(k - T(k), C(k - T(k)))$ on CAV n trajectory. From the matching point B, we can calculate a predicted trajectory point for the HDV $\hat{m}$ at time step k at point C: $(k, C(k - T(k)) - D(k) = \tilde{H}(k))$. It is very likely that points A and C won't match exactly and generate the trajectory matching error $e(k)$ in Eq. (43). According to the error $e(k)$, we further develop the error gain $e_D(k)$ and $e_T(k)$ considering that $e(k)$ can be either positive or negative. Using the case $e(k) > 0$ as an example, we explain our main idea as follows. The same idea can be applied for the case $e(k) < 0$. To make $\tilde{H}(k)$ consistent to $H(k)$, we can increase either the predicted distance displacement $D(k)$ by $e(k)$, or the predicted time displacement $T(k)$ by $e(k)/v_{\hat{m}}(k)$. Using this logic, we can derive Eqs. (44) and (45), where we substitute $v_{\hat{m}}(k)$ with $\max(v_{\hat{m}}(k), 1)$ to handle extreme scenarios when $v_{\hat{m}}(k) = 0$. Since we do not know what proportion we should increase $D(k)$ and $T(k)$ just based on the prediction errors at step k , gain coefficient a_D and a_T are designed small (e.g., $10^{-3} \sim 10^{-2}$) to calibrate $D(k)$ and $T(k)$ bit by bit and ensure the stability. Introducing this error gain is particularly useful to avoid the prediction error accumulation.

Appendix-VI

The pseudo-code of the online adaptive curve learning algorithm is presented below.

Initialize $W(k = 0) = 0$ and $D(k = 0), T(k = 0)$ according to the estimated number of HDVs (Hall, 1996) and calibrated parameters of Newell's Car-following model (Punzo and Simonelli, 2005)

Repeat for control time step $k + 1$:

Allocate data $\mathcal{H}_{\hat{m}}(k + 1) = \{x_{\hat{m}}^l = (x^l, t^l)\}, \mathcal{C}_n(k + 1) = \{x_n^l = (x^l, t^l)\}$

Determine $d(k + 1), t(k + 1), w(k + 1)$ according to Eqs. (35), (38)–(41).

Update $D(k + 1), T(k + 1), W(k + 1)$ according to Eqs. (32), (35), (36).

References

- Ahn, K., Rakha, H., Trani, A., Van Aerde, M., 2002. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *J. Transp. Eng.* 128 (2), 182–190.
- Androulakis, I.P., Floudas, C.A., 1999. Distributed branch and bound algorithms for global optimization. *Parallel Processing of Discrete Problems*. Springer, New York, NY, pp. 1–35.
- Alsabaan, M., Naik, K., Khalifa, T., 2013. Optimization of fuel cost and emissions using v2v communications. *IEEE Trans. Intell. Transp. Syst.* 14 (3), 1449–1461.
- Asadi, B., Vahidi, A., 2010. Predictive cruise control: utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Trans. Control Syst. Technol.* 19 (3), 707–714.
- Automobile Drag Coefficient, Wikipedia, the free encyclopedia. (n.d.), viewed 11 October 2021, Retrieved from https://en.wikipedia.org/wiki/Automobile_drag_coefficient.
- Beck, A., Tetruashvili, L., 2013. On the convergence of block coordinate descent type methods. *SIAM J. Optim.* 23 (4), 2037–2060.
- Bemporad, A., Heemels, W.M.H., De Schutter, B., 2002. On hybrid systems and closed-loop MPC systems. *IEEE Trans. Autom. Control* 47 (5), 863–869.

- Borrelli, F., Bemporad, A., Morari, M., 2017. Predictive Control For Linear and Hybrid Systems. Cambridge University Press.
- Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* 3 (1), 1–122.
- Chakraborty, P., Kikuchi, S., 1999. Evaluation of the general motors based car-following models and a proposed fuzzy inference model. *Transp. Res. Part C Emerg. Technol.* 7 (4), 209–235.
- Chen, C., Wang, J., Xu, Q., Wang, J., Li, K., 2021. Mixed platoon control of automated and human-driven vehicles at a signalized intersection: dynamical analysis and optimal control. *Transp. Res. Part C Emerg. Technol.* 127, 103138.
- Chen, J., Zhou, Y., Liang, H., 2019. Effects of ACC and CACC vehicles on traffic flow based on an improved variable time headway spacing strategy. *IET Intel. Transport Syst.* 13 (9), 1365–1373.
- Conejo, A.J., Nogales, F.J., Prieto, F.J., 2002. A decomposition procedure based on approximate Newton directions. *Math. Program.* 93 (3), 495–515.
- Dasgupta, S., Raghuraman, V., Choudhury, A., Teja, T.N., Dauwels, J., 2017. Merging and splitting maneuver of platoons by means of a novel PID controller. In: *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8.
- Duret, A., Wang, M., Ladino, A., 2020. A hierarchical approach for splitting truck platoons near network discontinuities. *Transp. Res. Part B Methodol.* 132, 285–302.
- Faraj, M., Sancar, F.E., Fidan, B., 2017. Platoon-based autonomous vehicle speed optimization near signalized intersections. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1299–1304.
- Feng, Y., Yu, C., Liu, H.X., 2018. Spatiotemporal intersection control in a connected and automated vehicle environment. *Transp. Res. Part C Emerg. Technol.* 89, 364–383.
- Ghasemi, A., Kazemi, R., Azadi, S., 2013. Stable decentralized control of a platoon of vehicles with heterogeneous information feedback. *IEEE Trans. Veh. Technol.* 62 (9), 4299–4308.
- Gong, S., Shen, J., Du, L., 2016. Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transp. Res. Part B Methodol.* 94, 314–334.
- Gong, S., Du, L., 2018. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transp. Res. Part B Methodol.* 116, 25–61.
- Guanetti, J., Kim, Y., Borrelli, F., 2018. Control of connected and automated vehicles: state of the art and future challenges. *Annu. Rev. Control* 45, 18–40.
- Guo, Y., Ma, J., Xiong, C., Li, X., Zhou, F., Hao, W., 2019. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: combined dynamic programming and shooting heuristic approach. *Transp. Res. Part C Emerg. Technol.* 98, 54–72.
- Guo, Q., Angah, O., Liu, Z., Ban, X.J., 2021. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transp. Res. Part C Emerg. Technol.* 124, 102980.
- Hall, F.L., 1996. Traffic stream characteristics. *Traffic flow theory. US federal highway administration*, 36.
- He, X., Liu, H.X., Liu, X., 2015. Optimal vehicle speed trajectory on a signalized arterial with consideration of queue. *Transp. Res. Part C Emerg. Technol.* 61, 106–120.
- Hu, Y., Li, W., Xu, K., Zahid, T., Qin, F., Li, C., 2018. Energy management strategy for a hybrid electric vehicle based on deep reinforcement learning. *Appl. Sci.* 8 (2), 187.
- Huang, Y., Ng, E.C., Zhou, J.L., Surawski, N.C., Chan, E.F., Hong, G., 2018. Eco-driving technology for sustainable road transport: a review. *Renew. Sustain. Energy Rev.* 93, 596–609.
- Huo, D., Meckl, P., 2022. Power management of a plug-in hybrid electric vehicle using neural networks with comparison to other approaches. *Energies* 15 (15), 5735.
- Kamal, M.A.S., Mukai, M., Murata, J., Kawabe, T., 2012. Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Trans. Control Syst. Technol.* 21 (3), 831–841.
- Li, L.H., Gan, J., Li, W.Q., 2018. A separation strategy for connected and automated vehicles: utilizing traffic light information for reducing idling at red lights and improving fuel economy. *J. Adv. Transp.* 2018.
- Lioris, J., Pedarsani, R., Tascikaraoglu, F.Y., Varaiya, P., 2016. Doubling throughput in urban roads by platooning. *IFAC-PapersOnLine* 49 (3), 49–54.
- Ma, F., Yang, Y., Wang, J., Li, X., Wu, G., Zhao, Y., Wu, L., Aksun-Guvenc, B., Guvenc, L., 2021. Eco-driving-based cooperative adaptive cruise control of connected vehicles platoon at signalized intersections. *Transp. Res. Part D Transport Environ.* 92, 102746.
- Mandava, S., Boriboonsomsin, K., Barth, M., 2009, October. Arterial velocity planning based on traffic signal information under light traffic conditions. *IEEE*, pp. 1–6.
- Marsden, G., McDonald, M., Brackstone, M., 2001. Towards an understanding of adaptive cruise control. *Transp. Res. Part C Emerg. Technol.* 9 (1), 33–51.
- Milánés, V., Shladover, S.E., Spring, J., Nowakowski, C., Kawazoe, H., Nakamura, M., 2013. Cooperative adaptive cruise control in real traffic situations. *IEEE Trans. Intell. Transp. Syst.* 15 (1), 296–305.
- Montanaro, U., Wroblewski, M., Dixit, S., Creighton, S., Pragalathan, S., Sorniotti, A., 2020. Linearising longitudinal vehicle dynamics through adaptive control techniques for platooning applications. *Int. J. Powertrains*.
- Nak, H., Akkaya, S., Yumuk, E., 2023. Active set method based model predictive control for a ball and beam system. In: *Proceedings of the 10th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE, pp. 871–875.
- Newell, G.F., 2002. A simplified car-following theory: a lower order model. *Transp. Res. Part B Methodol.* 36 (3), 195–205.
- Nie, Z., Farzaneh, H., 2021. Role of model predictive control for enhancing eco-driving of electric vehicles in urban transport system of Japan. *Sustainability* 13 (16), 9173.
- Niroumand, R., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2020. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. *Transp. Res. Part C: Emerg. Technol.* 116, 102659.
- Nocedal, J., Wright, S., 2006. Numerical Optimization. Springer Science & Business Media.
- Öncü, S., Ploeg, J., Van de Wouw, N., Nijmeijer, H., 2014. Cooperative adaptive cruise control: network-aware analysis of string stability. *IEEE Trans. Intell. Transp. Syst.* 15 (4), 1527–1537.
- Pauca, O., Maxim, A., Caruntu, C.F., 2021. Trajectory planner based on third-order polynomials applied for platoon merging and splitting. In: *Proceedings of the 29th Mediterranean Conference on Control and Automation (MED)*. IEEE, pp. 83–88.
- Peng, Q., Huo, D., Hall, C.M., 2022. A comparison of neural network-based strategies for diesel engine air handling control. In: *Proceedings of the American Control Conference (ACC)*. IEEE, pp. 3031–3037.
- Ploeg, J., Scheepers, B.T., Van Nunen, E., Van de Wouw, N., Nijmeijer, H., 2011. Design and experimental evaluation of cooperative adaptive cruise control. In: *Proceedings of the 14th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, pp. 260–265.
- Punzo, V., Simonelli, F., 2005. Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transp. Res. Rec.* 1934 (1), 53–63.
- Qin, R., Lu, Y., Guan, J., Ji, C., 2021. Eco-driving speed optimization model of urban intelligent connected vehicle platoon considering driver's comfort level. In: *Proceedings of the 2nd international conference on electronics, communications and information technology (CECIT)*. IEEE, pp. 532–537.
- Qu, X., Yu, Y., Zhou, M., Lin, C.T., Wang, X., 2020. Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: a reinforcement learning based approach. *Appl. Energy* 257, 114030.
- Shen, J., Kammara, E.K.H., Du, L., 2022. Fully distributed optimization-based CAV platooning control under linear vehicle dynamics. *Transp. Sci.* 56 (2), 381–403.
- Shi, J., Qiao, F., Li, Q., Yu, L., Hu, Y., 2018. Application and evaluation of the reinforcement learning approach to eco-driving at intersections under infrastructure-to-vehicle communications. *Transp. Res. Rec.* 2672 (25), 89–98.
- Smith, S.W., Kim, Y., Guanetti, J., Li, R., Firoozi, R., Wootton, B., Kurzhanskiy, A.A., Borrelli, F., Horowitz, R., Arcak, M., 2020. Improving urban traffic throughput with vehicle platooning: theory and experiments. *IEEE Access* 8, 141208–141223.
- Simchon, L., Rabinovici, R., 2020. Real-time implementation of green light optimal speed advisory for electric vehicles. *Vehicles* 2 (1), 35–54.
- Sun, C., Guanetti, J., Borrelli, F., Moura, S.J., 2020. Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet Things J.* 7 (5), 3759–3773.
- Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* 62 (2), 1805.

- USDOE, Driving more efficiently, <<https://www.fueleconomy.gov/feg/driveHabits.jsp>>, [Accessed 22 May 2022].
- Wan, N., Vahidi, A., Luckow, A., 2016. Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. *Transp. Res. Part C Emerg. Technol.* 69, 548–563.
- Wang, Z., Wu, G., Barth, M.J., 2019. Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment. *IEEE Trans. Intell. Transp. Syst.* 21 (5), 2029–2038.
- Wang, C., Nijmeijer, H., 2015. String stable heterogeneous vehicle platoon using cooperative adaptive cruise control. In: *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, pp. 1977–1982.
- Wei, Y., Avci, C., Liu, J., Belezamo, B., Aydın, N., Li, P.T., Zhou, X., 2017. Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. *Transp. Res. Part B Methodol.* 106, 102–129.
- Wiedemann, R., 1991. Modelling of RTI-Elements on multi-lane roads. In: *Proceedings of the Drive Conference (1991: Brussels, Belgium)* (Vol. 2).
- Wu, C., Xu, Z., Liu, Y., Fu, C., Li, K., Hu, M., 2020. Spacing policies for adaptive cruise control: a survey. *IEEE Access* 8, 50149–50162.
- Xiao, L., Gao, F., 2010. A comprehensive review of the development of adaptive cruise control systems. *Veh. Syst. Dyn.* 48 (10), 1167–1192.
- Yuan, D., Xu, S., Zhao, H., 2011. Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms. *IEEE Trans. Syst. Man Cybern. Part B* 41 (6), 1715–1724 (Cybernetics).
- Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vis.* 13 (2), 119–152.
- Zhang, H., Du, L., Shen, J., 2022. Hybrid MPC system for platoon based cooperative lane change control using machine learning aided distributed optimization. *Transp. Res. Part B Methodol.* 159, 104–142.
- Zhao, W., Ngoduy, D., Shepherd, S., Liu, R., Papageorgiou, M., 2018. A platoon based cooperative eco-driving model for mixed automated and human-driven vehicles at a signalised intersection. *Transp. Res. Part C Emerg. Technol.* 95, 802–821.
- Zhen, H., Mosharafian, S., Yang, J.J. and Velni, J.M., 2022. Eco-driving trajectory planning of a heterogeneous platoon in urban environments. *arXiv preprint arXiv: 2205.09618*.
- Zuo, L., Meng, D., Zhang, J., 2021. Optimal spacing policy for vehicle platoon control with road-friction coefficient. *J. Adv. Transp.* 2021.