Platoon-Centered Control for Eco-driving at Signalized Intersection Built upon Hybrid MPC System, Online Learning and Distributed Optimization

Hanyu Zhang
Department of Civil and Coastal Engineering,
University of Florida, Gainesville, Florida, 32608
Email: hanyu.zhang@ufl.edu
ORCID:0000-0002-5644-612X

Lili Du* (Corresponding author)
Associate professor
Department of Civil and Coastal Engineering,
University of Florida, Gainesville, Florida, 32608
Email: lilidu@ufl.edu
ORCID:0000-0003-1740-1209

Abstract: 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 develops a system optimal platoon-centered control for eco-driving (PCC-eDriving), which is able to guide a mixed flow platoon to smoothly approach, split as needed, and then sequentially pass signalized intersections, while reducing or even avoiding sharp deceleration and red idling. The PCC-eDriving control is mathematically enabled by a hybrid model predictive control (MPC) system, including three MPC controllers for platoon trajectory control and a mixed-integer nonlinear programming (MINLP) optimizer for optimal platoon splitting decision. The sequential and switching feasibility as well as the Input-to-State stability of the hybrid MPC system are analyzed and proved to sustain the system performance theoretically. Besides, we design robust vehicle dynamics and an online adaptive curve learning algorithm for the MPC controllers to factor control and vehicle driving uncertainties. The numerical experiments validate the merits of the proposed PCC-eDriving control and the involved approaches in improving traffic smoothness and efficiency and reducing energy consumption and emission at urban signalized intersections.

Keywords: Connected and autonomous vehicle, platoon-centered control, eco-driving strategy, model predictive control, hybrid MPC system, distributed optimization.

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 congestions at urban traffic intersections. In recent years, inspired by the rapid advancement of connected and autonomous vehicle (CAV) technologies, various eco-driving strategies and algorithms have been developed in the literature, with the aim to improve traffic safety and efficiency as well as reduce energy consumption and emission at signalized intersections. Mainly, existing studies leverage V2I communication technologies to advise vehicle trajectories adaptive to predefined traffic signal schedule at intersections. We categorized these efforts into two groups: (i) vehicle-level speed advisory, which provides constant speed plan (e.g., Mandava et al., 2009; Alsabaan et al., 2013; He et al., 2015; Wan et al., 2016; Simchon and Rabinovici, 2020) or real-time dynamic speed control (e.g., Asadi and Vahidi, 2010; Kamal et al., 2012; He et al., 2015; Sun et al., 2020; Nie and Farzaneh, 2021) for individual vehicles; (ii) platoon-level eco-driving strategy, which develops trajectory plan or control for the entire platoon (Wei et al., 2017; Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Zhang et al., 2019; Wang et al., 2019; Ma et al., 2021; Chen et al., 2021). These studies often use heuristic or optimization model-based approaches to enable the optimal trajectory plan and vehicle control. Some recent studies consider more proactive schemes for improving the traffic smoothness at the intersection by using joint optimization schemes to coordinate the vehicle platoon trajectory control and the traffic signal control (Feng et al., 2018; Guo et al., 2019; Niroumand et al., 2020). Due to the modeling and computation complexity, most of these studies focused on optimizing the leading vehicle's trajectory rather than the entire platoon's movements. The existing studies (e.g., Lioris et al., 2016) showed that the platoon-level eco-driving strategies perform better than vehicle-level speed advisory in terms of system traffic efficiency, and thus attract tremendous research interests. This study considers an entire platoon as a system and intends to develop robust and energy-efficient platoon-centered control so that it can respond to traffic signal control at traffic intersection smartly and smoothly. Thus, we share the same interest in platoon-level eco-driving with predefined traffic signal schedule.

Apart from the model-based approaches mentioned above, other researchers started to apply learning-based approaches, especially reinforcement learning for eco-driving (Hu et al., 2018; Shi et al., 2018; Qu et al., 2020; Guo et al., 2021). Compared with the model-based approaches, the learning-based approaches can track actual vehicle driving dynamics more accurately and handle complex driving scenarios of which the model-based approaches are complicated and difficult. However, those learning-based approaches consider vehicles are non-cooperative and learn responsive instruction for individual vehicle without rigorously considering entire platoon's performance, and thus cannot achieve system optimal solution. On the other hand, it is very challenging to use the reinforcement learning approach for developing platoon-level eco-driving strategy. This is because when CAVs of a platoon are under a reinforcement learning control, the dimension of the discretized action and state space will grow exponentially. It significantly challenges the computation and convergence of the reinforcement learning. Along with this thought, this study tends to use model-based approaches. More specifically, we seek to develop a hybrid model predictive control (MPC) system for platoon-level eco-driving strategy. Nevertheless, we noticed the following research gaps in existing model-based eco-driving studies, which motivate us to develop an advanced eco-driving control by involving several enhanced features.

First of all, most of the existing model-based eco-driving strategies focus on providing vehicle trajectory plan or speed advice (Wei et al., 2017; Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Wang et al., 2019; Chen et al., 2021) rather than using feedback based closed-loop control. Although showing improved performance in saving energy consumption and reducing emission, this type of planning scheme usually cannot adapt to the trajectory deviations in a planning horizon due to various uncertain factors. Accordingly, many studies work on a pure CAV traffic flow (Wei et al., 2017, Faraj et al., 2017, Li et al., 2018; Ma et al., 2021;). Even though several recent efforts 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 eco-driving strategies use deterministic double-integrator vehicle dynamics without factoring control uncertainty resulting from powertrain delay, aerodynamic drag, etc. Nevertheless, modeling these uncertainties into the eco-driving strategies will lead to nonconvex optimizers or controllers. It consequently brings in extra difficulty to the theoretical analyses in control and the solution approach development 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 most of the existing platoon-level eco-driving strategies only optimize the leading vehicle's trajectory while implementing the following vehicles with adaptive cruise control (ACC) or cooperative adaptive cruise control (CACC) (Wei et al., 2017, Li et al., 2018; Ma et al., 2021). These types of responsive controllers cannot theoretically guarantee the platoon system performance, even though some control feature such as string stability can be ensured (Ploeg et al., 2011, Öncü et al., 2014). In view of this weakness, this study considers an entire platoon as a system and employs a system optimal model predictive platoon-centered control, which involves MPC to systematically control the trajectories of all CAVs in a platoon in real time. This type of platoon-centered control has proved superior performance in maintaining platoon driving efficiency and smoothness (Gong and Du, 2018). However, these merits accompany with new challenges particularly for this study. Specifically, the system optimal MPC often involves an optimizer, which must be solved within a control interval (< 1 second) to ensure control continuity. However, the scale of the optimizer becomes large and it is very difficult to solve the optimizer promptly when a long platoon or a large prediction horizon is considered in this study. In addition, existing MPC platoon-centered control (Gong et al., 2016; Gong and Du, 2018; Shen et al., 2021) 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 will develop MPC platoon-centered control with adaptive desired spacing policy and then develop a new customized distributed optimization algorithm to solve MPC efficiently.

Furthermore, a key challenge of the eco-driving strategy is how to properly manage a long platoon's trajectory if it cannot completely pass the intersection within one green interval. Existing studies (Faraj et al., 2017; Li et al., 2018; Zhao et al., 2018; Chen et al., 2021) use simple heuristic rules to spit a platoon for only maximizing traffic throughputs. It has been noticed that those approaches often 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 further noticed that splitting a platoon at an intersection will induce three states of platoon control respectively for the car-following, splitting, and intersection passing. Accordingly, finding the optimal splitting point as well as generating trajectory control instructions under each state should cooperate and enable smooth state switching. However, most existing literature ignored these control details. 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 forementioned research gaps and enhanced points 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 signalized intersection, which instructs a mixed flow platoon to smoothly and efficiently approach and then pass signalized intersections reducing red idling as much as possible. The development of the PCC-eDriving control contributes the following mathematical modeling, 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 programming optimizer (MINLP), which together generate optimal control laws/switching signal for the platoon to approach an intersection, split into sub-platoons, and then sequentially pass intersection during different green intervals.

To ensure the performance of the PCC-eDriving control, we further proved the control feasibility and stability of the hybrid MPC system. To be noted, the MPC controllers factor the control and vehicle (CAV and HDV) driving uncertainty and employ an adaptive desired spacing policy. They make the MPC controllers robust to traffic dynamics on urban roads and improve traffic throughputs but introduce new theoretical challenges regarding control feasibility and stability against disturbances. This study investigated these issues by analyzing the MPC controllers and proving the hybrid MPC system's sequential/switching feasibility and Input-to-State stability.

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 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 sec) for factoring the traffic variation during the 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 second. Besides, the employment of the adaptive desired spacing policy can significantly improve the road capacity by 44% compared with 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 and significantly reduce 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.

Following the introduction, we present the efforts of this study by the structure below. Section 2 formally defines the research problem. Then we provide vehicle dynamics, constraints in section 3. Built upon that, we develop the hybrid MPC system in section 4 and prove the control feasibility and stability in section 5. The AS-OCD solution approach and the adaptive curve learning approach are developed in section 6 and section 7 respectively. Last, this study conducts numerical experiments to validate our approaches in section 8, and finally draws a conclusion in section 9.

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 Figure 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 $\widehat{m}$-many HDVs indexed by $\{\hat{1}, \dots, \widehat{m}\}$, 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 general preceding-and-following communication network topology. Specifically, we consider that the CAVs in the platoon segments C_1 and C_2 are well connected to their immediate preceding and following CAVs through V2V communication. Besides, this study considers the movements of the HDVs $\hat{0}$, $\hat{1}$ and $\widehat{m}$ can be detected by the adjacent CAVs using on-board sensors.

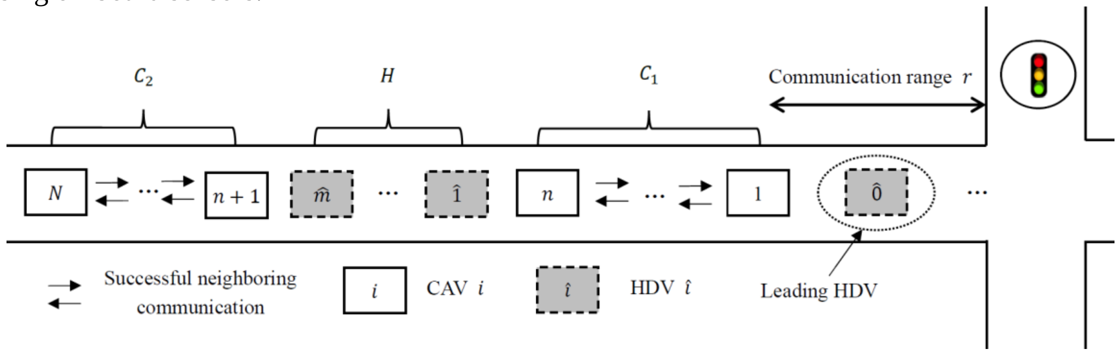


Figure 1. Sample platoon at the signalized intersection

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 $x_{\hat{\imath}}$, $v_{\hat{\imath}}$ $u_{\hat{\imath}}$ for $\forall \hat{\imath} \in I_H = \{\hat{0}, \hat{1}, \dots, \widehat{m}\}$ to represent the longitudinal position, speed, and acceleration of the i_{th} CAV and the $\hat{\imath}_{th}$ 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.

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 Figure 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 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 as illustrated in Figure 2.

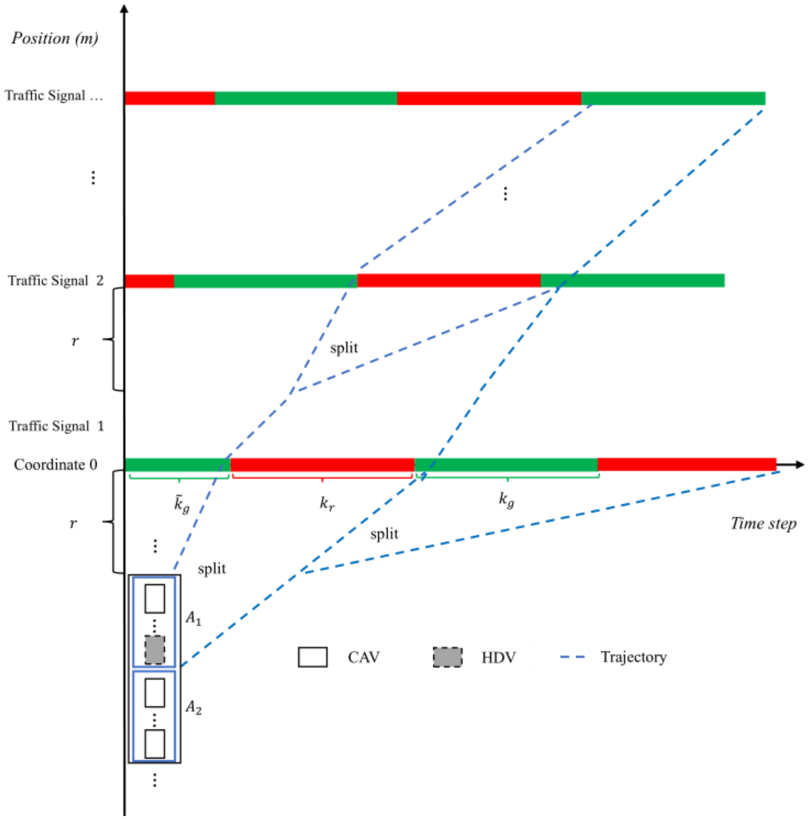


Figure 2. Platoon splitting scheme

This PCC-eDriving control and the hybrid MPC system introduced above bring in technical challenges from the aspects in mathematical modeling, theoretical analyses and solution approach development. It thus motivates novel methodology development in this study. We formally present them in the following sections.

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 Equations (1) and (2) to describe CAV i's dynamics at step k.

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

$$v_i(k+1) = v_i(k) + \tau\big(u_i(k) - \Delta u_i(k)\big), \qquad i \in I_C \tag{2}$$

In Equations (1) and (2), $\Delta u_i(k)$ factors CAV i's powertrain delay and aerodynamic drag by

Equation (3), which was proposed by Montanaro et al., (2020).

$$\Delta u_i(k) = \varepsilon_i v_i(k) + \eta_i u_i(k) - \eta_i u_i(k-1), \qquad i \in I_C \tag{3}$$

In Equation (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 $0 < \underline{\varepsilon_i} \le \varepsilon_i \le \overline{\varepsilon}_i$, $0 < \underline{\eta_i} \le \eta_i \le \overline{\eta}_i$, where $\underline{\varepsilon_i}$, $\overline{\varepsilon}_i$, $\underline{\eta_i}$ and $\overline{\eta}_i$ 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 $\underline{\Delta_i} \le \Delta u_i(k) \le \overline{\Delta}_i$. 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 Newell's car-following model (Newell, 2002) in Equation (4),

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

where $t_{\hat{\imath}}$ and $d_{\hat{\imath}}$ represent the time and distance displacement of the HDV $\hat{\imath}$. 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 ($\widehat{m}$) in the HDV segment H, we describe its movement using the aggregated form of Newell's car-following model in Equation (5),

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

where $T_{\widehat{m}}$ and $D_{\widehat{m}}$ represent the aggregated time step and distance displacement of the HDV segment H. Mathematically, $T_{\widehat{m}} = \lceil (t_{\widehat{m}} + t_{\widehat{m}-1} + \cdots + t_1)/\tau \rceil$, $D_{\widehat{m}} = d_{\widehat{m}} + d_{\widehat{m}-1} + \cdots + d_{\hat{1}}$. Using Equation (5) with given parameters $T_{\widehat{m}}$ and $D_{\widehat{m}}$, we can predict the trajectory of HDV $\widehat{m}$ using the trajectory of CAV n. Nevertheless, we noticed that HDVs' time and distance displacement $T_{\widehat{m}}$ and $D_{\widehat{m}}$ are highly heterogenous and may vary under different driving environments. It is challenging but important to accurately capture the time-varying HDV behavior for the PCC-eDriving control. To address this technical difficulty, this study develops an adaptive curve learning approach in Section 7 to learn and update the values of parameters $T_{\widehat{m}}$ and $D_{\widehat{m}}$ online.

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

$$a_{min,i} \le u_i(k) \le a_{max,i}, \qquad i \in I_C \tag{6}$$

$$v_{min} \le v_i(k) \le v_{max}, \qquad i \in I_C \tag{7}$$

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

Here, $a_{max,i}$ and $a_{min,i}$ in Equation (6) are used to describe the predefined acceleration/ deceleration bounds for CAV i. Similarly, v_{min} and v_{max} in Equation (7) are the pre-specified bounds of driving speed on urban roads. Regarding the driving safety, an adaptive safe distance constraint is enabled by Equation (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 Equation (8), parameter $L_i > 0$ represents CAV i's length plus minimum car-following distance; $\delta_1 \tau$ with $\delta_1 \ge 1$ is the safe time headway; $\delta_2 \tau \big(v_i(k) - v_{i-1}(k)\big)$ with $\delta_2 \ge 0$ 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 Equation (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 Equation (9), which will be used in the objective function of the MPC control.

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

In Equation (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 Equation (8). This makes the control system hardly trigger the activation of the safe distance constraints in Equation (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 Equations (10) and (11) respectively as follows.

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

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

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

$$z(k) \coloneqq \big(\Delta x_1(k), \dots, \Delta x_N(k)\big)^T \in \mathbb{R}^N, \tag{12}$$

$$z'(k) \coloneqq \big(\Delta v_1(k), \dots, \Delta v_N(k)\big)^T \in \mathbb{R}^N. \tag{13}$$

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

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

where P is the MPC prediction horizon. The terminal constraint in Equation (14) requires the platoon's spacing and speed errors at final time step $k + P$ of the MPC are confined to small domains $\zeta = \left[-\frac{\tau^2}{2}\overline{\Delta}_i P, -\frac{\tau^2}{2}\underline{\Delta}_i P\right]$ and $\zeta' = [-\tau\overline{\Delta}_i P, -\tau\underline{\Delta}_i P]$ respectively, where $\underline{\Delta}_i$ and $\overline{\Delta}_i$ are the lower and upper bound of the CAV control input uncertainty in Equations (3). According to Equations (3), (6) and (7), $\underline{\Delta}_i = \varepsilon_i v_{min} + \eta_i\big(a_{min.i} - a_{max,i}\big) < 0$ and $\overline{\Delta}_i = \varepsilon_i v_{max} + \eta_i\big(a_{max.i} - a_{min,i}\big) > 0$. 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 in 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 errors timely and smoothly. 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 as 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 Figure 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 second). 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., $q_0 \overset{\sigma_0}{\rightarrow} q_1, q_2$). 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 Figure 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 5.2.

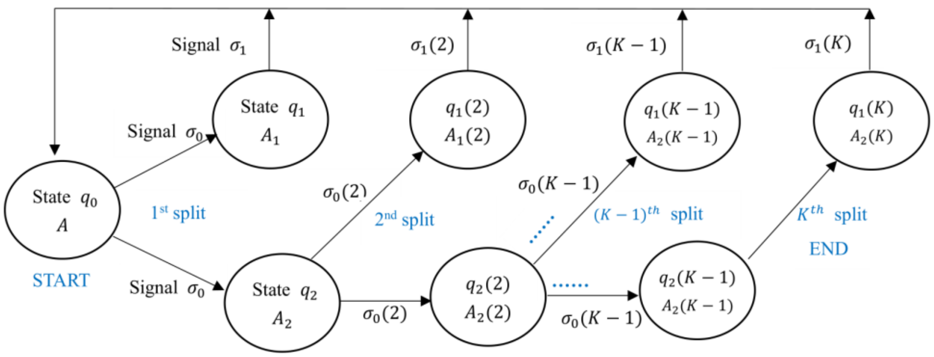


Figure 3. Hybrid MPC system

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 Equation (15) subject to the vehicle dynamics and constraints in Equations (1)-(14) developed in section 3. If there is no vehicle ahead leading the platoon, then the first platoon 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 $\forall\, p = 1, \dots, P$ ($p \in P$). We simplify step $k + p$ to p hereafter throughout the section 4 to simplify notation.

MPC-q_0

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

Subject to, for $p \in P$:

Constraints in Equations (1)-(14),

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

Note that this platoon-centered car-following control is different from the existing approach in Gong et al., (2016); Gong and Du, (2018); Shen et al., (2021). It explicitly factors the CAV control uncertainties by Equation (3), HDV driving variations by Equations (4) and (5) as well as adaptive safe distance constraints and desired spacing policy by Equations (8) and (9). These enhanced features capture more traffic and control reality, aiming to improve the traffic system performance under urban environments, but introduce extra challenges in the MPC sequential feasibility and control stability. We will address these challenges in section 5.

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 1**.

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 Figure 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 $y_i \in \{0,1\}$, $\forall i \in \overline{I_c} = \{1,2,\dots,n,n+1,\dots N,N+1\}$ 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 Equation (16) below to allow only one splitting point in the mixed flow platoon.

$$\sum\nolimits_{i\in\overline{I_c}} y_i = 1; y_i \in \{0,1\} \tag{16}$$

Denote i^* as the platoon splitting point, then the last CAV in the sub-platoon A_1 is CAV i^*-1. We thus have Equation (17) to ensure 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 Equation (17) takes effect only for CAV i^*-1.

$$x_{i-1}(\tilde{k}_g) \geq -M(1-y_i), \quad i \in \overline{I_c}, \tag{17}$$

On the other side, the first CAV in the second sub-platoon A_2 is CAV i^*. Then Equation (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.

$$x_i(\tilde{k}_g+k_r) \leq M(1-y_i), \quad i \in \overline{I_c}\,, \tag{18}$$

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 Equations (19) and (20),

$$\Delta x_i(p) = x_{i-1}(p) - x_i(p) - s_i(p) - y_i * \mathcal{D}, i \in \overline{I_c}, \tag{19}$$

$$\Delta v_i(p) = v_{i-1}(p) - v_i(p) - y_i * \mathcal{D}', i \in \overline{I_c}, \tag{20}$$

where parameters $\mathcal{D}$ and in $\mathcal{D}'$ represent the estimated spacing and speed errors between two sub-platoons A_1 and A_2. Note that Equations (19) and (20) are same to the Equations (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

$$\textbf{Min}\ \ J(u,y) = J_1(u,y) + \omega_2 J_2(u,y) \tag{21}$$

Subject to, for $p \in P, i \in \overline{I_c}$:

Constraints in Equations (1)-(9), (12)- (13), (16)-(20),

where

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

$$J_2(u,y) = -\sum_{i\in\overline{I_c}} i * y_i$$

The objective function J in Equation (21) 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 will guide the followed MPC-q_1 and MPC-q_2 control and help to ensure the feasibility of the state switching $q_0 \overset{\sigma_0}{\rightarrow} (q_1, q_2)$. Note that the MINLP-$\sigma_0$ 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 $\overline{I_c}$ 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

$$\textbf{Min}\ \Gamma(u) = \sum_{p=1}^{P} \left\{\frac{1}{2}\left[z^T(p)Q_z z(p) + \left(z'(p)\right)^T Q_{z'} z'(p)\right] + \frac{\tau^2}{2}\omega_1 \|u(p-1)\|_2^2\right\} \tag{22}$$

Subject to, for $i \in \widehat{A_1}, p \in P$ where $P = \tilde{k}_g, \tilde{k}_g - 1, \dots, 1$:

Constraints in Equations (1)-(13), and

$$x_{i^*-1}(P) \geq 0, \tag{23}$$

where $\widehat{A_1}$ 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, Equation (23) ensures that the sub-platoon A_1 passes the intersection within the remaining green interval. It is derived from

Equation (17) in the MINLP-σ_0. To keep control continuity and facilitate switching feasibility, MPC-q_1 also shares the same constraints in Equations (1)-(13) with MPC-q_0. We will discuss these technical details in section 5.2.2.

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 Figure 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^*.

$$\textbf{Min } \mathrm{F}(u_{i^*}) = \sum_{p=1}^{P} \|u_{i^*}(p-1)\|_2^2 - \omega_3 x_{i^*}(P) \tag{24}$$

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 Equations (1)-(3) and (6)-(8)

$$x_{i^*}(P) \le 0, \tag{25}$$

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. Constraint in Equation (25) illustrates that the leading CAV of A_2 cannot pass the intersection until the red phase runs out. This is derived from Equation (18) and thus facilitates the switching feasibility (see Section 5.2 for technical details). The objective function in Equation (24) 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. The two conflicting interests can be toned by the weight parameter ω_3 in practice.

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

$$\textbf{Min } \Gamma(u) = \sum_{p=1}^{P} \left\{ \frac{1}{2}\left[z^T(p) Q_z z(p) + \left(z'(p)\right)^T Q_{z'} z'(p) \right] + \frac{\tau^2}{2}\omega_1 \|u(p-1)\|_2^2 \right\} \tag{26}$$

Subject to, for $i \in \widehat{A_2} \backslash i^*, p \in P$ where $P = \tilde{k}_g + k_r,\ \tilde{k}_g + k_r - 1, \dots, 1$:

Constraints in Equations (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 1**.

Remark 1. 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., Equations (24) and (25)) 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. Control Feasibility and Stability

This section investigates three important theoretical properties of the hybrid MPC system: sequential feasibility, switching feasibility and stability. They together theoretically ensure the hybrid MPC system can generate feasible and stable control law to guide the platoon going through the intersection efficiently and smoothly.

5.1. MPC sequential feasibility

As the MPC is implemented recursively at each time step $0, 1, \dots, k-1, k,$ a fundamental theoretical question is whether the MPC can find a feasible control law at each time step k (i.e., whether the constraint set of the MPC optimizer is non-empty at each time step k), given the platoon system starts from an initial feasible condition at $k = 0$. The system is called sequential (recursive) feasible (Löfberg, 2012) if the answer to this question is affirmative. The hybrid MPC system in this study has MPC-q_0, MPC-q_1 and MPC-q_2 controllers. MPC-q_0 has constraints in Equations (1)-(14), which are further shared with MPC-q_1 and MPC-q_2 except the terminal constraint in Equation (14). Hence, this study first proves the sequential feasibility of MPC-q_0. Then, we further discuss the sequential feasibility of MPC-q_1 and MPC-q_2 as well as the switching feasibility of the hybrid system between different MPCs in section 5.2. To prove the sequential feasibility of the MPC-q_0, we first classify Equations (1)-(14) into the following three sets:

(i) $\mathcal{S}_1(u(k))$: constraint set in Equations (1)-(3) and (6)-(8) for capturing the CAV dynamics, acceleration, speed and safety constraints at step $k \in \mathbb{Z}_+$.

(ii) $\mathcal{S}_2(u(k), Z(k+P))$: the HDV movements in Equations (4)-(5) and the terminal constraint in Equation (14) at step $k \in \mathbb{Z}_+$.

(iii) $\mathcal{S}_3(u(k))$: the control dynamics in Equations (9)-(13) at step $k \in \mathbb{Z}_+$.

It should be noticed that the third constraint set $\mathcal{S}_3(u(k))$ are control dynamic formulations. They are always feasible if the first constraint set $\mathcal{S}_1(u(k))$ is feasible. For the second constraint set $\mathcal{S}_2(u(k), Z(k+P))$, Equations (4)-(5) are equality constraints to curve the HDV trajectory. Thus, they are always feasible in math and also stay feasible in practice if accurate time and distance displacements are estimated. Apart from it, the terminal constraint in Equation (14) is only active at the final time step $k + P$ of the MPC. Its sequential feasibility is ensured if the nominal MPC system is asymptotical stable (see proof in detail in section 5.3) according to Mayne et al., (2000). Consequently, to prove the sequential feasibility of the MPC-q_0, this study will mainly analyze and prove the sequential feasibility of the first constraint set $\mathcal{S}_1(u(k))$ by **Lemma 1**.

Lemma 1. For $k \in \mathbb{Z}_+ := \{0,1,2,\dots\}$ and $i \in I_C$, if $\mathcal{S}_1(u_i(k))$ is feasible, then there exists $\delta_1 \geq 1$ and $\delta_2 \geq 0$ that make $\mathcal{S}_1\big(u_i(k+1)\big)$ feasible and compact. In addition, the non-empty feasible control input profile $\mathbb{S}_1(u_i(k))$ for platoon vehicle i at step k is given below:

$$u_i(k) \in \mathbb{S}_1\big(u_i(k)\big) = \left[\max\left\{a_{min,i}, \underline{a_{i,v}}\right\}, \min\left\{a_{max,i}, \overline{a_{i,v}}, \overline{a_{i,d}}\right\}\right], \tag{27}$$

where

$$\underline{a_{i,v}} = \frac{v_{min} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} \leq 0$$
$$\overline{a_{i,v}} = \frac{v_{max} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} \geq 0$$
$$\overline{a_{i,d}} = \varepsilon_i v_i(k) + \eta_i u_i(k) - \eta_i u_i(k-1) + \frac{g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big)}{\tau^2\left(\delta_1 + \delta_2 + \frac{1}{2}\right)} + \frac{\left(\delta_2 - \frac{1}{2}\right)\underline{u_{i-1}(k)}}{\delta_1 + \delta_2 + \frac{1}{2}}$$
$$g_i(k) = x_{i-1}(k) - x_i(k) - \Big(L_i + \delta_1\tau v_i(k) + \delta_2\tau\big(v_i(k) - v_{i-1}(k)\big)\Big)$$

Proof: To prove the sequential feasibility of the constraint set $\mathcal{S}_1\big(u_i(k)\big)$ constituted of Equations (1)-(3) and (6)-(8), we need to find non-empty control input profile $\mathbb{S}_1(u_i(k))$ at step k for $k \in \mathbb{Z}_+$ that makes the constraint set $\mathcal{S}_1\big(u_i(k+1)\big)$ feasible, given that $\mathcal{S}_1\big(u_i(k)\big)$ is feasible. Namely, with feasible state at any step k, the MPC can have a feasible control input at step k leading to a feasible state at step $k+1$. Below we provide the technical details.

We first reformulate the speed limit constraint at step $k+1$ according to Equations (2), (3) and (7). And then we find its corresponding feasible control input set $u_i(k) \in \left[\underline{a_{i,v}}, \overline{a_{i,v}}\right]$ as follows in Equation (28).

$$\begin{aligned} & v_{min} \leq v_i(k+1) \leq v_{max} \\ & \quad \Leftrightarrow v_{min} \leq v_i(k) + \tau\big(u_i(k) - \Delta u_i(k)\big) = v_{max} \\ & \quad \Leftrightarrow v_{min} \leq (1-\tau\varepsilon_i)v_i(k) + \tau(1-\eta_i)u_i(k) + \tau\eta_i u_i(k-1) \leq v_{max} \\ & \quad \Leftrightarrow u_i(k) \in \left[\underline{a_{i,v}}, \overline{a_{i,v}}\right], \end{aligned} \tag{28}$$

where the lower bound $\underline{a_{i,v}}$ and upper bound $\overline{a_{i,v}}$ are given in Equation (29).

$$\begin{aligned} \underline{a_{i,v}} &= \frac{v_{min} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} \\ \overline{a_{i,v}} &= \frac{v_{max} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} \end{aligned} \tag{29}$$

Similarly, we reformulate the safe distance constraints at step $k+1$ according to Equations (1)-(3) and (8). For discussion convenience, we use $g_i(k)$ to represent the safe distance constraint in Equation (8) for CAV i at step k and denote $\underline{u_i(k)} = u_i(k) - \Delta u_i(k)$. Then, we present the mathematical derivations below in Equation (30).

$$\begin{aligned} g_i(k+1) &= x_{i-1}(k+1) - x_i(k+1) - \Big(L_i + \delta_1\tau v_i(k+1) + \delta_2\tau\big(v_i(k+1) - v_{i-1}(k+1)\big)\Big) \\ &= x_{i-1}(k) - x_i(k) + \tau\big(v_{i-1}(k) - v_i(k)\big) + \frac{\tau^2}{2}\left(\underline{u_{i-1}(k)} - \underline{u_i(k)}\right) \\ &\quad - \Big(L_i + \delta_1\tau v_i(k) + \delta_2\tau\big(v_i(k) - v_{i-1}(k)\big)\Big) - \delta_1\tau^2\underline{u_i(k)} - \delta_2\tau^2\left(\underline{u_i(k)} - \underline{u_{i-1}(k)}\right) \\ &= g_i(\mathrm{k}) + \tau\big(v_{i-1}(k) - v_i(k)\big) + \tau^2\left(\delta_2 + \frac{1}{2}\right)\underline{u_{i-1}(k)} - \tau^2\left(\delta_1 + \delta_2 + \frac{1}{2}\right)\underline{u_i(k)} \\ &= g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big) + \tau^2\left(\delta_2 - \frac{1}{2}\right)\underline{u_{i-1}(k)} - \tau^2\left(\delta_1 + \delta_2 + \frac{1}{2}\right)\underline{u_i(k)} \end{aligned} \tag{30}$$

Note that we assume $\mathcal{S}_1(u_i(k))$ is feasible, it makes the safe distance constraints in Equation (8) are feasible at time step k. Mathematically, $g_i(k) \geq 0$. To make the safe distance constraints keep feasible at next time step $k+1$ (i.e., $g_i(k+1) \geq 0$), we should have the following control input requirement ($u_i(k) \leq \overline{a_{i,d}}$) in Equation (31) based upon Equations (3) and (30).

$$\begin{aligned} u_i(k) \leq \overline{a_{i,d}} &= \varepsilon_i v_i(k) + \eta_i u_i(k) - \eta_i u_i(k-1) \\ &\quad + \frac{g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big)}{\tau^2\left(\delta_1 + \delta_2 + \frac{1}{2}\right)} + \frac{\left(\delta_2 - \frac{1}{2}\right)\underline{u_{i-1}(k)}}{\delta_1 + \delta_2 + \frac{1}{2}} \end{aligned} \tag{31}$$

Recall that we define $\delta_1 \geq 1$, $\delta_2 \geq 0$ for Equation (8).Without loss of generalizability, we pick $\delta_1 \geq \max\left\{\frac{v_{min} - v_{max}}{\tau(a_{min,i} - \varepsilon v_{min})} - 1, 1\right\}$, $\delta_2 = \frac{1}{2}$ to show the sequential feasibility[1]. By plugging in $\delta_2 = \frac{1}{2}$, we can remove the term with $\underline{u_{i-1}(k)}$ and simplify Equation (31) to Equation (32) below.

$$\overline{a_{i,d}} = \varepsilon_i v_i(k) + \eta_i u_i(k) - \eta_i u_i(k-1) + \frac{g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big)}{\tau^2(\delta_1 + 1)} \tag{32}$$

Wrapping the Equations (6), (28), (31) and (32), we have the following solution set $\mathbb{S}_1\big(u_i(k)\big)$ in Equation (33) that makes the constraints set $\mathcal{S}_1(u_i(k+1))$ feasible given that $\mathcal{S}_1(u_i(k))$ is feasible.

$$u_i(k) \in \mathbb{S}_1\big(u_i(k)\big) = \left[\max\left\{a_{min,i}, \underline{a_{i,v}}\right\}, \min\{a_{max,i}, \overline{a_{i,v}}, \overline{a_{i,d}}\}\right] \tag{33}$$

We next show $\mathbb{S}_1\big(u_i(k)\big)$ in Equation (33) is non-empty. To do that, it suffices to show $\max\left\{a_{min,i}, \underline{a_{i,v}}\right\} \leq \min\left\{a_{max,i}, \overline{a_{i,v}}, \overline{a_{i,d}}\right\}$. More specifically, we need to prove the following six inequalities hold (i) $a_{max,i} \geq a_{min,i}$, (ii) $a_{max,i} \geq \underline{a_{i,v}}$, (iii) $\overline{a_{i,v}} \geq a_{min,i}$, (iv) $\overline{a_{i,v}} \geq \underline{a_{i,v}}$, (v) $\overline{a_{i,d}} \geq a_{min,i}$, (vi) $\overline{a_{i,d}} \geq \underline{a_{i,v}}$. It is obvious that (i) $a_{max,i} \geq a_{min,i}$ and (iv) $\overline{a_{i,v}} \geq \underline{a_{i,v}}$ hold according to Equations (6) and (29). Below we sequentially show the inequalities (ii), (iii), (v) and (vi) are satisfied in the Equations (34)-(37) when $\delta_1 \geq \max\left\{\frac{v_{min} - v_{max}}{\tau(a_{min,i} - \varepsilon v_{min})} - 1, 1\right\} \geq \frac{v_{min} - v_{max}}{\tau(a_{min,i} - \varepsilon v_{min})} - 1$ and $\delta_2 = \frac{1}{2}$.

Specifically, we confirm inequality (ii) $a_{max,i} \geq \underline{a_{i,v}}$ holds by the derivations given in Equation (34).

$$\begin{aligned} a_{max,i} - \underline{a_{i,v}} &= a_{max,i} - \frac{v_{min} - (1 - \tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1 - \eta_i)} \\ &\geq a_{max,i} + \frac{-v_{min} + (1 - \tau\varepsilon_i)v_{min} + \tau\eta_i a_{min,i}}{\tau(1 - \eta_i)} \\ &= a_{max,i} - \frac{\varepsilon_i v_{min} - \eta_i a_{min,i}}{(1 - \eta_i)} > 0 \end{aligned} \tag{34}$$

[1] Please note that the selection of the parameters here is to ensure feasibility rigorously. It is not necessarily the best choice for the implementation.

We prove inequality (iii) $\overline{a_{i,v}} \geq a_{min,i}$ by the mathematical process in Equation (35).

$$\begin{aligned}\overline{a_{i,v}} - a_{min,i} &= \frac{v_{max} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} - a_{min,i} \\ &\geq \frac{v_{max} - (1-\tau\varepsilon_i)v_{max} - \tau\eta_i a_{max,i}}{\tau(1-\eta_i)} - a_{min,i} \\ &= \frac{\varepsilon_i v_{max} - \eta_i a_{max,i}}{(1-\eta_i)} - a_{min,i} > 0 \end{aligned} \tag{35}$$

To ensure inequality (v) $\overline{a_{i,d}} \geq a_{min,i}$, we develop the mathematical process in Equation (36).

$$\begin{aligned}\overline{a_{i,d}} - a_{min,i} &= \frac{\varepsilon_i v_i(k) - \eta_i u_i(k-1)}{(1-\eta_i)} + \frac{g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big)}{\tau^2(\delta_1+1)(1-\eta_i)} - a_{min,i} \\ &\geq \frac{1}{(1-\eta_i)}\left[\frac{v_{i-1}(k+1) - v_i(k)}{\tau(\delta_1+1)} - (1-\eta_i)a_{min,i} \right. \\ &\quad \left. + \big(\varepsilon_i v_i(k) - \eta_i u_i(k-1)\big)\right] + \frac{g_i(k)}{\tau^2(\delta_1+1)(1-\eta_i)} \\ &\geq \frac{1}{(1-\eta_i)}\left[\frac{v_{min} - v_{max}}{\tau(\delta_1+1)} - (1-\eta_i)a_{min,i} + \big(\varepsilon_i v_{min} - \eta_i a_{min,i}\big)\right] \\ &= \frac{1}{(1-\eta_i)}\left[\frac{v_{min} - v_{max}}{\tau(\delta_1+1)} - \big(a_{min,i} - \varepsilon_i v_{min}\big)\right] \end{aligned} \tag{36}$$

By choosing a feasible $\delta_1 \geq \frac{v_{min}-v_{max}}{\tau(a_{min,i}-\varepsilon v_{min})} - 1$, we have $\frac{1}{(1-\eta_i)}\left[\frac{v_{min}-v_{max}}{\tau(\delta_1+1)} - \big(a_{min,i}-\varepsilon_i v_{min}\big)\right] \geq 0$. Consequently, we confirm inequality (v) $\overline{a_{i,d}} - a_{min,i} \geq 0$ in Equation (36).

Last, we confirm inequality (v) $\overline{a_{i,d}} \geq a_{min,i}$ by the derivation below in Equation (37).

$$\begin{aligned}\overline{a_{i,d}} - \underline{a_{i,v}} &= \left[\frac{\varepsilon_i v_i(k) - \eta_i u_i(k-1)}{(1-\eta_i)} + \frac{g_i(k) + \tau\big(v_{i-1}(k+1) - v_i(k)\big)}{\tau^2(\delta_1+1)(1-\eta_i)}\right] \\ &\quad - \frac{v_{min} - (1-\tau\varepsilon_i)v_i(k) - \tau\eta_i u_i(k-1)}{\tau(1-\eta_i)} \\ &> \frac{1}{\tau(1-\eta_i)}\left[\frac{v_{i-1}(k+1) - v_i(k)}{\delta_1+1} - \big(v_{min} - v_i(k)\big)\right] \\ &= \frac{v_{i-1}(k+1) - v_{min} + \delta_1(v_i(k) - v_{min})}{\tau(1-\eta_i)(\delta_1+1)} \geq 0 \end{aligned} \tag{37}$$

Wrapping the results above, we prove the sequential feasibility of the constraints $\mathcal{S}_1\big(u(k)\big)$, with which we conclude the **Lemma 1** is proved. ■

5.2. Hybrid switching feasibility

Switching feasibility guarantees that MPC controllers in a hybrid system can switch feasibly according to the system design. Then to ensure the continuity of the hybrid system, it is crucial to prove the switching feasibility. Namely, the system can keep feasible as the control switches from an MPC controller to another. As shown in Figure 3, the hybrid system involves three types of state switchings (i) $q_0 \overset{\sigma_0}{\rightarrow} q_1, q_2$; (ii) $q_1 \overset{\sigma_1}{\rightarrow} q_0$; (iii) $q_2(\kappa-1) \overset{\sigma_0}{\rightarrow} q_1(\kappa), q_2(\kappa)$. Given the sequential feasibility of MPC-q_0 proved in Section 5.1, we will show the switching feasibility one by one below.

(i). $q_0 \overset{\sigma_0}{\rightarrow} q_1, q_2$ represents the platoon A under the control of MPC-q_0 splits into two sub-platoons A_1 and A_2 by switching signal σ_0, and then operate respectively under the control of MPC-q_1 and MPC-q_2. Note that MPC-q_1 and MPC-q_2 share the same vehicle dynamics and constraints in Equations (1)-(13) with MPC-q_0 except extra constraints in Equations (23) and (25). Therefore, the control switching feasibility is ensured if MPC-q_1 and MPC-q_2 stay sequential feasible with these extra constraints. To demonstrate it, recall that the switching signal MINLP -σ_0 also shares the same vehicle dynamics and constraints in Equations (1)-(13) with MPC-q_0, while Equation (23) in MPC-q_1 and (25) in MPC-q_2 are respectively derived from Equations (17) and (18) in the MINLP-σ_0. That is to say, if MINLP-σ_0 can find a feasible platoon splitting point factoring Equations (17) and (18), then involving constraints in Equations (23) or (25) into MPC-q_0 for switching to MPC-q_1 or MPC-q_2 (i.e., implementing splitting point decision in the MPC control) will ensure that MPC-q_1 and MPC-q_2 can always find feasible trajectory control solution at each step (i.e., the sequential feasibility of MPC-q_1 and MPC-q_2 is confirmed). According to the definition of the platoon splitting point in Equation (16), MINLP-σ_0 can always find a feasible platoon splitting point. Therefore, we confirm the sequential feasibility of MPC-$q_1$and MPC-q_2 and the switching feasibility of $q_0 \overset{\sigma_0}{\rightarrow} q_1, q_2$.

(ii). $q_1 \overset{\sigma_1}{\rightarrow} q_0$ represents the control of the sub-platoon A_1 switches from MPC-q_1 to MPC-q_0 when the switching signal σ_1 is triggered. Note that the MPC-q_0 has an extra terminal constraint in Equation (14) compared with MPC-q_1. If the terminal constraint in Equation (14) is always feasible, then the switching $q_1 \overset{\sigma_1}{\rightarrow} q_0$ is feasible. Recall that the terminal constraint in Equation (14) regulates the platoon control error to be limited in a defined range at the end of the prediction horizon P. It is always feasible when the prediction horizon P is large enough. Therefore, the feasibility of this switching is ensured.

(iii). $q_2(\kappa-1) \xrightarrow{\sigma_0(\kappa)} q_1(\kappa), q_2(\kappa)$ represents the sub-platoon $A_2(\kappa-1)$ under the control of MPC-$q_2(\kappa-1)$ further splits into the sub-platoon $A_1(\kappa)$ under the control of MPC-$q_1(\kappa)$ and the sub-platoon $A_2(\kappa)$ under the control of MPC-$q_2(\kappa)$ when switching signal $\sigma_0(\kappa)$ is triggered. Similar to the first type of switching, MPC-$q_1(\kappa)$ and MPC-$q_2(\kappa)$ share the same vehicle dynamics and constraints with MPC-$q_2(\kappa-1)$, but have extra constraints in Equations (23) and (25). The switching feasibility is ensured according to the same discussion for (i) $q_0 \overset{\sigma_0}{\rightarrow} q_1, q_2$.

Overall, the discussions above ensure the sequential feasibility of MPC-q_1 and MPC-q_2 and also guarantee the feasible state switching in the well-designed hybrid system. ■

5.3. MPC control stability

The control stability plays a vital role in ensuring the system performance. It ensures the platoon system can go to the desired spacing and speed even it is exposed to disturbances. A system fails to reach or sustain its control goal if it is instable. Typically, the majority of the control stability can be analyzed and proved by linear analysis techniques (Sastry and Bodson, 2011). However, the receding horizon MPC control with constraints like the MPC-q_0, MPC-q_1 and MPC-q_2 in this study constitute a nonlinear feedback control system, of which the standard linear techniques are not sufficient to prove the stability. To tackle this challenge, we employ the Lyapunov theory in **Theorem 1** below to characterize the nominal system[2] behaviors and prove the asymptotical stability of the nominal MPC-q_0 first.

[2] Nominal system is defined as the system's dynamics without modeling uncertainty errors

Theorem 1. Asymptotic Stability (Bof et al., 2018): Let $\varphi = 0$ be an equilibrium point for the autonomous system $\varphi(k+1) = f\big(\varphi(k)\big)$, where $f: D \rightarrow \mathbb{R}^n$ is locally Lipschitz in $D \subset \mathbb{R}^n$ and $0 \in D$. Suppose there exists a function V: $D \rightarrow \mathbb{R}^n$ which is continuous and such that (i) $V(\varphi = 0) = 0$ and $V(\varphi) > 0$ for $\forall \varphi \in D - \{0\}$; (ii) $V\left(f\big(\varphi(k)\big)\right) - V(\varphi) \leq 0$ for $\forall \varphi \in D$, then $\varphi = 0$ is stable. The function V that satisfies conditions (i) and (ii) is called Lyapunov function. Moreover, if (iii) $V\left(f\big(\varphi(k)\big)\right) - V(\varphi) < 0$ for $\forall \varphi \in D - \{0\}$, then $\varphi = 0$ is asymptotic stable.

Furthermore, this study is aware that the uncertainties in Equation (3) makes our MPC-q_0 a robust system rather than the nominal system. Proving the stability of the nominal system is not sufficient for this study. We therefore continue to prove the Input-to-State stability of MPC-q_0 in **Remark 2** based upon **Theorem 2**. The Input-to-State stability ensures the system states are bounded around the equilibrium point (i.e., the steady-state of the platoon), when uncertainties are involved. In other words, the platoon under the MPC-q_0 control in this study will perform well under control uncertainties, if it is Input-to-State stable.

Theorem 2. Input-to-State Stability (Zeilinger et al., 2009): The discrete-time robust linear system $\varphi(k+1) = A\varphi(k) + Bu(k) + w(k)$ is Input-to-State stable if the corresponding nominal system $\varphi(k+1) = A\varphi(k) + Bu(k)$ is asymptotically stable and the disturbance $w(k)$ is bounded.

Please note that the stability (e.g., Input-to-State stability) is used to evaluate the performance of an MPC in the long term. MPC-q_1 and MPC-q_2 are MPC controllers with shrinking prediction horizons and are only used for a short time period when the platoon is near the intersection. Hence, we only show the stability proof for MPC-q_0.

5.3.1. Reformulation and nominal system of the MPC-q_0

Before developing our stability analyses, we first rewrite Equations (1)-(3) and (9)-(13) in Equations (38) and (39) respectively,

$$\varphi(k+1) = A\varphi(k) + Bu(k) + w(k) \tag{38}$$

$$\varphi(k+1) = A\varphi(k) + Bu(k) \tag{39}$$

where for $\forall k \in \mathbb{Z}_+$, $\varphi(k) = \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} \in \mathbb{R}^{2N}$ is the control state, $u(k) \in \mathbb{R}^{N+1}$ is the control input, $w(k) \in W \in \mathbb{R}^{2N}$ is the bounded uncertainty that is contained in a convex and compact set introduced by Equation (3). Equation (38) is discrete-time robust control dynamics of our MPC-q_0, whereas Equation (39) is the corresponding nominal control dynamics by taking off the uncertainty $w(k)$. Then, we have A, B and W defined in Equation (40). The corresponding mathematical derivations to obtain them are provided in the Appendix-II.

$$A = \begin{bmatrix} I_N & \tau I_N \\ 0 & I_N \end{bmatrix} \in \mathbb{R}^{2N \times 2N}; B = \begin{bmatrix} \tau^2 S_1 \\ \tau S_2 \end{bmatrix} \in \mathbb{R}^{2N \times (N+1)}; W = \left[-\begin{bmatrix} \tau^2 S_1 \\ \tau S_2 \end{bmatrix} \overline{\Delta}_i, -\begin{bmatrix} \tau^2 S_1 \\ \tau S_2 \end{bmatrix} \underline{\Delta}_i \right] \tag{40}$$

The matrix S_1, S_2 in Equation (40) are shown below, where $c_1 = \delta_2 + \frac{1}{2}$; $c_2 = -\left(\delta_1 + \delta_2 + \frac{1}{2}\right)$:

$$S_1 = \begin{bmatrix} c_1 & c_2 & & & \\ & c_1 & c_2 & & \\ & & \ddots & \ddots & \\ & & & c_1 & c_2 \end{bmatrix} \in \mathbb{R}^{N \times (N+1)}; \; S_2 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{N \times (N+1)}.$$

Next, we transform the robust MPC-q_0 developed in section 4.1 into the nominal system MPC-$\mathbb{Q}_0$ below using the nominal control dynamics in Equation (39). Note that the CAV control uncertainty in Equation (3) is removed.

$$\text{MPC} - \mathbb{Q}_0 \ \min_{u} \Gamma(k) = \sum_{p=1}^{P} \ell(\varphi(p|k), u(p|k))$$

$$\text{Subject to: } \varphi(p+1|k) = A\varphi(p|k) + Bu(p|k), \qquad p = 0, 1, \dots, P-1$$

$$\big(\varphi(p+1|k), u(p|k)\big) \in \Phi \times \mathrm{U}, \qquad p = 0, 1, \dots, P-1$$

$$\varphi(P|k) \in \Phi_f$$

The notation $(p|k)$ in the MPC-$\mathbb{Q}_0$ represents the time step of $p + k$; Φ and U are linear constraints on the states and inputs corresponding to Equations (4)-(8). According to the MPC-q_0 objective function in Equation (15), the stage cost is defined as $\ell(\varphi(p|k), \mu(p|k)) := \frac{1}{2}[z^T(p|k)Q_z z(p|k) + (z'(p|k))^T Q_{z'} z'(p|k)] + \frac{\tau^2}{2}\omega_1 \|u(p-1|k)\|_2^2$. Φ_f is the compact terminal constraint defined in Equation (14). Please note that under the nominal system in the Equation (39), Φ_f becomes the equilibrium point, namely $\Phi_f = 0$: $z(P|k) = 0, z'(P|k) = 0$.

Solving the MPC- $\mathbb{Q}_0$ will generate an optimal control sequence $\widehat{\boldsymbol{u}}(k) = [u^*(0|k), u(1|k), \dots, u(P-1|k)]$, in which the MPC controller will only implement the current control decision $u^*(0|k)$, and recompute MPC-$\mathbb{Q}_0$ at the next time step. $u^*(0|k)$ can be considered as a feedback control of the current state $\varphi(0|k)$ at step k as follows in Equation (41):

$$u^*(0|k) = K_f \varphi(0|k) \tag{41}$$

where K_f is the corresponding feedback control law obtained by solving MPC-$\mathbb{Q}_0$. From the optimal control sequence $\widehat{\boldsymbol{u}}(k)$, we can obtain corresponding state sequence $\widehat{\boldsymbol{\varphi}}(k) = [\varphi(1|k), \varphi(2|k), \dots, \varphi(P|k)]$ as well as the optimal value function $\Gamma_V\big(k, \widehat{\boldsymbol{u}}(k)\big)$ of the MPC-$\mathbb{Q}_0$. $\Gamma_V\big(k, \widehat{\boldsymbol{u}}(k)\big)$ is the value of the objective function $\Gamma(k)$ when the optimal control sequence $\widehat{\boldsymbol{u}}(k)$ is applied to the system. The value function $\Gamma_V\big(k, \widehat{\boldsymbol{u}}(k)\big)$ will be employed as a Lyapunov function to prove the asymptotic stability in section 5.3.2. We abbreviate $\Gamma_V\big(k, \widehat{\boldsymbol{u}}(k)\big)$ as $\Gamma_V(k)$ hereafter for discussion convenience.

5.3.2. Asymptotic Stability and Input-to-State Stability

This section proves the asymptotical stability of the nominal MPC-$\mathbb{Q}_0$ and then the Input-to-State stability of the robust MPC-q_0. Specifically, we first show $\Gamma_V(k)$ is a Lyapunov function and thus the nominal MPC-$\mathbb{Q}_0$ is asymptotic stable in **Lemma 2** according to the Lyapunov Theory in **Theorem 1**. Then together **Theorem 2**, we conclude the robust MPC-q_0 is Input-to-State stable in **Remark 2**.

Lemma 2. Suppose $\Gamma_V(k)$ is the optimal value function of MPC-$\mathbb{Q}_0$, then $\Gamma_V(k)$ is a Lyapunov function and MPC-$\mathbb{Q}_0$ is asymptotic stable.

Proof. According to the feedback control law in Equation (41), the nominal controller in Equation (39) can be converted into $\varphi(k+1) = \big(A + BK_f\big)\varphi(k)$ for $\forall k \in \mathbb{Z}_+$, which is an autonomous system stated in **Theorem 1**. Accordingly, the autonomous system dynamic function $f = A + BK_f$ is linear time invariant and thus Lipschitz in a domain $D = \mathbb{R}^N$. Then, to prove $\Gamma_V(k)$ is a Lyapunov function and MPC-$\mathbb{Q}_0$ is asymptotic stable, we need to prove the function $\Gamma_V(k)$ satisfies the three conditions in **Theorem 1**, which are (i) $\Gamma_V(\varphi(k) = 0) = 0$ and $\Gamma_V\big(\varphi(k)\big) > 0$

for $\forall\varphi(k) \in \mathbb{R}^N - \{0\}$; (ii) $\Gamma_V\left(f\big(\varphi(k)\big)\right) - \Gamma_V\big(\varphi(k)\big) \leq 0$ for $\forall\varphi(k) \in \mathbb{R}^N$; (iii) $\Gamma_V\left(f\big(\varphi(k)\big)\right) - \Gamma_V\big(\varphi(k)\big) < 0$ for $\forall\varphi(k) \in \mathbb{R}^N - \{0\}$.

Recall that $\Gamma_V(k)$ represents the optimal value of the objective function $\Gamma(k)$ in the MPC-$\mathbb{Q}_0$, which is obtained by applying the optimal control sequence $\widehat{\boldsymbol{u}}(k)$ to the system[3]. The quadratic formulation of $\Gamma(k)$ in Equation (15) indicates $\Gamma(k) \geq 0$ so that we have $\Gamma_V(k) \geq 0$. Apart from it, $\Gamma_V(k) = 0$ holds if and only if the MPC-$\mathbb{Q}_0$ reaches equilibrium point (i.e., $\varphi(k) = 0$, all CAV keep same reference speed and apart with desired spacing). Hence, condition (i) is satisfied.

Then, we need to show conditions (ii) and (iii) are satisfied. Specifically, $\Gamma_V(k+1) < \Gamma_V(k)$ for $\forall\varphi \in \mathbb{R}^N - \{0\}$ and $\Gamma_V(k+1) = \Gamma_V(k)$ when $\varphi = 0$. The mathematical definition of function $\Gamma_V(k)$ and $\Gamma_V(k+1)$ are given below in Equations (42) and (43).

$$\Gamma_V(k) = \min_u \sum_{p=1}^{P} \ell(\varphi(p|k), u(p-1|k)) = \min_u \sum_{p=1}^{P} \ell(\varphi(p+k), u(p+k-1)) \tag{42}$$

$$\Gamma_V(k+1) = \min_u \sum_{p=1}^{P} \ell(\varphi(p+k+1), u(p+k)) \tag{43}$$

According to Equations (42) and (43), we have the following mathematical derivations for $\Gamma_V(k+1)$ in Equation (44).

$$\begin{aligned} \Gamma_V(k+1) &= \min_u \left\{ \sum_{p=1}^{P-1} \ell\big(\varphi(p+k+1), u(p+k)\big) + \ell\big(\varphi(P+k+1), u(P+k)\big) \right\} \\ &= \min_u \left\{ \sum_{p=2}^{P} \ell\big(\varphi(p+k), u(p+k-1)\big) + \ell\big(\varphi(k+1), u(k)\big) \right\} \\ &\qquad - \min_{u(k)} \ell\big(\varphi(k+1), u(k)\big) + \min_{u(P+k)} \ell\big(\varphi(P+k+1), u(P+k)\big) \\ &= \Gamma_V(k) - \min_{u(k)} \ell\big(\varphi(k+1), u(k)\big) + \min_{u(P+k)} \ell\big(\varphi(P+k+1), u(P+k)\big) \end{aligned} \tag{44}$$

Since we define the terminal constraint $\varphi(P|k) \in \Phi_f = 0$ in the nominal MPC-$\mathbb{Q}_0$, we have the state φ reach the equilibrium point at step $k + P$ (i.e., $\varphi(P+k) = 0$). Then there exits feasible control input $u(P+k) = 0$ to drive the state φ stay at the equilibrium point 0, namely $\varphi(P+k+1) = 0$. Hence, $\min_{u(P+k)} \ell\big(\varphi(P+k+1), u(P+k)\big) = 0$. Together Equation (44), we have the following Equation (45).

$$\Gamma_V(k+1) \leq \Gamma_V(k) - \min_{u(k)} \ell\big(\varphi(k+1), u(k)\big) \tag{45}$$

Further, based on Equation (45) and $\ell\big(\varphi(k+1), u(k)\big) \geq 0$, we have

$$\Gamma_V(k+1) \leq \Gamma_V(k)$$

Condition (ii) is satisfied. Moreover $\Gamma_V(k+1) = \Gamma_V(k)$ holds if and only if $\ell\big(\varphi(k+1), u(k)\big) = 0$, namely $\varphi(k) = 0, u(k) = 0$. Condition (iii) is satisfied. In summary, function $\Gamma_V(k)$ is a Lyapunov function that satisfies three conditions in **Theorem 1**. Hence, MPC-$\mathbb{Q}_0$ is asymptotic stable. We conclude the proof. ■

[3] $\Gamma_V(k)$ refers to $\Gamma_V\big(\varphi(k)\big)$ and correspondingly $\Gamma_V(k+1)$ refers to $\Gamma_V\left(f\big(\varphi(k)\big)\right)$.

Remark 2. Given that the uncertainty $w(k)$ is bounded and **Lemma 2** has proved that the nominal MPC-$\mathbb{Q}_0$ is asymptotic stable, we claim that the robust MPC-q_0 is Input-to-State stable according to **Theorem 2**. It ensures that the platoon states such as vehicle speeds and spacings in this study are bounded around our desired control goal under the robust MPC control. Besides, the gap between the actual control performance and our literal control goal highly depends on the uncertainty $w(k)$. Specifically, the gap decreases as the uncertainty $w(k)$ becomes smaller. When $w(k) = 0$, the system becomes nominal system without uncertainty and the gap becomes 0.

6. 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). 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 details.

6.1. Active set based optimal condition decomposition (AS-OCD)

This study noticed that most of the constraints in the three MPC controllers are inequality constraints and they are inactive[4] 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 Equations (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., 2017) 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). By integrating AS and OCD approaches, we can solve the MPC optimizer efficiently in a distributed manner. Below we show the procedure of the AS-OCD algorithm.

```
Initialize an empty Active set $AS = \emptyset$
Repeat until 'no infeasible constraints & negative Lagrange multipliers'
    (1) Solve the Optimizer, subject to $AS$ constraints (OCD Inner loop)
        Repeat until $\Delta u$, $\Delta\eta$, $\Delta\lambda$ small enough:
            a) Solve search direction according to Equation (49)
            b) Update solution: $u \leftarrow u + \Delta u, \eta \leftarrow \eta + \Delta\eta, \lambda \leftarrow \lambda + \Delta\lambda$
        End repeat
    (2) Compute the Lagrange multipliers of the $AS$ constraints
    (3) Remove constraints with negative Lagrange multipliers from $AS$
End repeat
```

[4] 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.

In short, the AS method is used as the outer loop to identify active constraint set AS of the MPC optimizer, while the OCD algorithm is used as an inner loop to iteratively solve the optimizer subject to current active set AS in a distributed manner. The rest of this section focuses on developing the OCD approach. 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 Equation (46) only subject to active set constraints.

$$\text{Optimizer } F\text{: Min } \Gamma(u) = \sum_{i=1}^{N}\left\{z_i^T Q_{z_i} z_i + (z_i')^T Q_{z_i'} z_i' + \frac{\tau^2}{2}\|u_i\|_2^2\right\} \tag{46}$$

$$\text{Subject to } l\text{-many } g_{AS}\left(u_{j-1}, u_j\right) = 0,\ \ell\text{-many } c_{AS}\left(u_{\not{j}}\right) = 0,$$

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, $i \in I_C$. $g_{AS}\left(u_{j-1}, u_j\right) = 0$ is an active coupled[5] constraint such as safe distance constraints in Equation (8) and $c_{AS}\left(u_{\not{j}}\right) = 0$ is an active uncoupled constraint such as acceleration and speed limit constraints in Equations (6) and (7). For this optimizer F, we assume there are l-many active coupled constraints $g_{AS}\left(u_{j-1}, u_j\right) = 0$ and ℓ-many uncoupled constraints $c_{AS}\left(u_{\not{j}}\right) = 0$. Then, the Lagrangian function of the above optimizer F is given by:

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

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 $\Delta u, -\Delta\eta, -\Delta\lambda$ by solving $K\,\Delta = -f$ in Equation (47), 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.

$$\underbrace{\begin{pmatrix} \nabla^2 L_{AS}(u,\eta,\lambda) & \nabla^T c_{AS}(u) & \nabla^T g_{AS}(u) \\ \nabla c_{AS}(u) & 0 & 0 \\ \nabla g_{AS}(u) & 0 & 0 \end{pmatrix}}_{K} \underbrace{\begin{pmatrix} \Delta u \\ -\Delta\eta \\ -\Delta\lambda \end{pmatrix}}_{\Delta} = -\underbrace{\begin{pmatrix} \nabla L_{AS}(u,\eta,\lambda) \\ c_{AS}(u) \\ g_{AS}(u) \end{pmatrix}}_{f} \Leftrightarrow K\,\Delta = -f \tag{47}$$

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 Equation (47)). To adapt the Newton's method to a distributed optimization algorithm, we approximates $\nabla^2 L_{AS}(u, \sigma, \lambda)$ and $\nabla^T g_{AS}(u)$ in the KKT matrix K to block diagonal matrices[6] H and A in Equation (48).

$$H = diag\left(\frac{\partial^2 L_{AS}(u,\eta,\lambda)}{\partial u_1^2}, \dots, \frac{\partial^2 L_{AS}(u,\eta,\lambda)}{\partial u_N^2}\right);\ A = diag\left(\frac{\partial g_1(u)}{\partial u_1}, \dots, \frac{\partial g_N(u)}{\partial u_N}\right) \tag{48}$$

[5] 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).

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

This approximation provides us an approximated KKT matrix $\overline{K}$ in Equation (49), which are separable in terms of CAV control inputs. Accordingly, we solve the search direction by $\overline{\Delta}= -\overline{K}^{-1}f$. The computation load can be distributed to individual CAVs $\overline{\Delta}_i= -\overline{K}_i^{-1}f_i$ shown in Equation (49),

$$\underbrace{\begin{pmatrix} \overline{K}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \overline{K}_N \end{pmatrix}}_{\overline{K}} \underbrace{\begin{pmatrix} \overline{\Delta}_1 \\ \vdots \\ \overline{\Delta}_N \end{pmatrix}}_{\overline{\Delta}} = -\underbrace{\begin{pmatrix} f_1 \\ \vdots \\ f_N \end{pmatrix}}_{f} \Leftrightarrow \overline{\Delta}_i= -\overline{K}_i^{-1} f_i, \forall i = 1, \dots, N, \tag{49}$$

where $\overline{K}_i$ represents the CAV i's matrix block in the approximated KKT matrix $\overline{K}$; $\overline{\Delta}_i$ and f_i are CAV i's search direction and corresponding gradient. The mathematical formulations of $\overline{K}_i$, $\overline{\Delta}_i$, f_i can be seen in Appendix-III. We will discuss the solution quality and convergence of this OCD algorithm in the next section 6.2. 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 the three MPC controllers efficiently.

6.2. Convergence of the AS-OCD

This AS-OCD algorithm is an iterative distributed algorithm. Then theoretical questions regarding its convergence performance are posed (i.e., whether the AS-OCD can converge to the optimal solution and how fast it converges). We investigate these issues in this section. The main idea is given as follows. According to the study of Nocedal and Wright, (2006), the AS method can converge to the global optimal solution for a strictly convex quadratic problem with linear constraints, and it works efficiently for optimization problems with few active constraints. The MPC optimizers of this study satisfy these features. Hence, the convergence rate and solution optimality of the AS-OCD algorithm highly depend on the performance of the OCD algorithm. According to **Theorem 3** in Conejo et al., (2002), the OCD algorithm can guarantee the linear convergence rate to the global optimal solution when an optimizer satisfies certain conditions. We will use this **Theorem 3** to prove the convergence of the OCD. For completeness, we provide **Theorem 3** below.

Theorem 3. (Conejo et al., 2002) For an optimizer F in Equation (46), using the OCD algorithm can obtain the global optimal solution[7] l^* with linear convergence rate at least equal to ρ^* if the following conditions hold.
(I). Functions Γ, g, c have Lipschitz-continuous second derivatives in an open set containing l^*;
(II). The matrix $\overline{K}$ is nonsingular[8] for any l, and the sequence $\{\overline{K}\}$ converges to a nonsingular matrix $\overline{K}$ as $l \rightarrow l^*$;
(III). At the second order KKT point l^*, $\rho^* = \rho\left(I - \left(\overline{K}\right)^{-1} K\right) < 1$.

Built upon **Theorem 3**, we prove **Lemma 3** to confirm that the OCD algorithm can obtain global optimal solutions efficiently.

Lemma 3. The MPC optimizer F in Equation (46) satisfies conditions (I), (II) and (III) in **Theorem 3** and thus can obtain global optimal solution in linear convergence rate using OCD algorithm.

[7] The global optimal solution $l^* = (u^*, \sigma^*, \lambda^*)$ is the second-order KKT point of optimizer F. When optimizer F is convex, l^* is the optimal solution.

[8] Nonsingular matrix is a square matrix whose determinant is not equal to zero (i.e., has full rank).

Proof: First of all, the function Γ, g and c of the MPC optimizer F in Equation (46) are quadratic or linear. Then they are globally second-order Lipschitz-continuous according to Sohrab, (2003). Thus, Condition (I) is satisfied. Next, given that the matrix $\overline{K}$ is the hessian matrix of Γ when constraints g, c are not active, then $\overline{K}$ is constant because Γ is quadratic. Besides, $\overline{K}$ is diagonalizable[9] and we can easily show Γ is strictly convex with positive eigenvalues according to Equation (48) and MPC-q_0 optimizer in Equation (15). Hence, $\overline{K}$ has full rank and is always nonsingular (Greub, 2012). Condition (II) is satisfied.

Then, we focus on proving condition (III) holds by showing spectral radius (all the eigenvalues) of the matrix $(I - (\overline{K})^{-1}K)$ is smaller than 1 (i.e., $\rho^* < 1$). To do that, we first provide the explicit expression of the matrix $(I - (\overline{K})^{-1}K)$. Then we use the induced norm inequality[10] $\rho(A) \leq \|A\|$ to prove the spectral radius of matrix $(I - (\overline{K})^{-1}K)$ is less than 1. The mathematical process involves dense algebra work. We provide the details of the proof in Appendix-IV. ■

7. Online Adaptive Curve Learning Approach

Recall that this study estimates the movements of HDVs by Newell's car-following model in Equation (5). The two parameters $T_{\widehat{m}}$ (the aggregated time displacement) and $D_{\widehat{m}}$ (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 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.

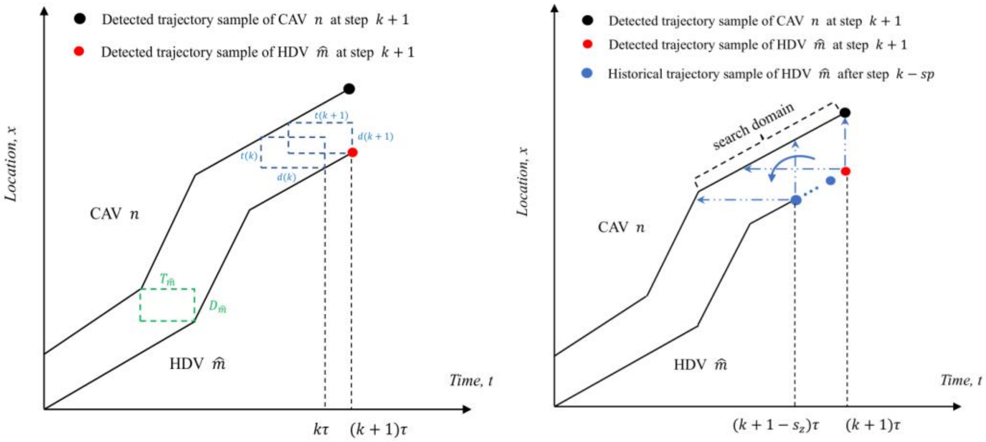
Motivated by the above view, we develop a new adaptive curve learning algorithm by taking advantage of the unique features of the Newell's car-following model. It seeks to learn and update parameters $T_{\widehat{m}}$ and $D_{\widehat{m}}$ at each control step based on the trajectory data, requiring 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 7.1 and 7.2 respectively.

7.1. 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., $T_{\widehat{m}}, D_{\widehat{m}}$ in Figure 4 (a)). By moving the trajectory data of the HDV $\widehat{m}$ at time step $k+1$ (red dots in Figure 4 (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 $k-1, \ldots, 0$. 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.

[9] If a matrix A is Hermitian symmetric, then A is diagonalizable. $\overline{K}$ is symmetric and thus diagonalizable.

[10] For a matrix A, $\rho(A) \leq \|A^r\|^{1/r}$ for all positive integers r, where $\rho(A)$ is the spectral radius of A. When $r = 1$, $\rho(A) \leq \|A\|$, $\|A\|$ is the 1-norm of matrix A.


Detected trajectory sample of CAV n at step $k+1$
Detected trajectory sample of HDV $\hat{m}$ at step $k+1$
Location, x
CAV n
$t(k+1)$
$d(k+1)$
$t(k)$
$d(k)$
$T_{\hat{m}}$
$D_{\hat{m}}$
HDV $\hat{m}$
Time, t
$k\tau$
$(k+1)\tau$
Detected trajectory sample of CAV n at step $k+1$
Detected trajectory sample of HDV $\hat{m}$ at step $k+1$
Historical trajectory sample of HDV $\hat{m}$ after step $k-sp$
search domain
Location, x
CAV n
HDV $\hat{m}$
Time, t
$(k+1-s_z)\tau$
$(k+1)\tau$

(a). Adaptive learning Approach (b). Point matching algorithm

Figure 4. Online Adaptive Curve Learning Approach

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. Equations (50)-(52) below wrap up the ideas,

$$W(k) = \sum_{j=0}^{k} w(k), \tag{50}$$

$$T(k) = \sum_{j=0}^{k} \gamma^{k-j} t(j) w(j) \,/W(k), \tag{51}$$

$$D(k) = \sum_{j=0}^{k} \gamma^{k-j} d(j) w(j) \,/W(k), \tag{52}$$

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 7.2 to determine $d(k)$ and $t(k)$. Below we first present the formulation to determine the weight $w(k)$ in Equation (53).

$$w(k) = |v_{\hat{m}}(k) - v_{\hat{m}}(k-1)| \tag{53}$$

Equation (53) 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 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 Equations (51)-(52) 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 Equations (51)-(52) is modified into the following Equations (54) and (55), which relate $T(k+1), D(k+1)$ to $T(k), D(k)$ and $t(k+1), d(k+1)$.

$$T(k+1) = \frac{W(k)}{W(k)+w(k+1)}\gamma T(k) + \frac{w(k+1)}{W(k)+w(k+1)}t(k+1) + e_D(k) \tag{54}$$

$$D(k+1) = \frac{W(k)}{W(k)+w(k+1)}\gamma D(k) + \frac{w(k+1)}{W(k)+w(k+1)}d(k+1) + e_T(k) \tag{55}$$

As such, our adaptive learning approach in Equations (54) and (55) 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 Equations (54) and (55) are calculated based on previous trajectory matching errors. The technical details can be seen in Appendix-V.

7.2. 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 $\mathcal{H}_{\widehat{m}}(k+1)$ to represent the data set containing newly detected sample trajectory data of HDV $\widehat{m}$ at time step $k+1$ (red dot in Figure 4 (b)) and the recent s_z-many historical trajectory data collected after step $k+1-s_z$ (blue dots in Figure 4(b)). Set $\mathcal{H}_{\widehat{m}}(k+1)$ includes $|\mathcal{H}_{\widehat{m}}| = (s_z+1)$-many sample data, represented as $\mathcal{H}_{\widehat{m}}(k+1) = \{\mathscr{p}_{\widehat{m}}^l = (x^l, t^l), l = 1, \dots, |\mathcal{H}_{\widehat{m}}|\}$. 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 $\mathcal{C}_n(k+1) = \{\mathscr{p}_n^l = (x^l, t^l),\ l = 1, \dots, |\mathcal{C}_n|\}$. 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 $\mathcal{H}_{\widehat{m}}(k+1)$ of HDV $\widehat{m}$.

The point matching algorithm wipes every possible $|\mathcal{H}_{\widehat{m}}|$-many consecutive data points in data set $\mathcal{C}_n(k+1)$. $\mathcal{N} = |\mathcal{C}_n| - |\mathcal{H}_{\widehat{m}}|$ represents the number of possible pairs. For any possible pair $\mathscr{j}$, $\forall \mathscr{j} = 1, \dots, \mathcal{N}$, a motion vector $\mathcal{J}^{\mathscr{j}} = (-d^{\mathscr{j}}, t^{\mathscr{j}})$ is searched for the least errors in Equation (56).

$$\text{Min}\, \mathcal{F}^{\mathscr{j}}(\mathcal{J}^{\mathscr{j}}) = \frac{1}{|\mathcal{H}_{\widehat{m}}|}\sum_{l=1}^{|\mathcal{H}_{\widehat{m}}|} \left\| \mathscr{p}_n^l - \mathscr{p}_{\widehat{m}}^l + \mathcal{J}^{\mathscr{j}} \right\|^2, \forall \mathscr{j} = 1, \dots, \mathcal{N} \tag{56}$$

Obviously, we can have an explicit solution for the unconstrained quadratic optimization problem in Equation (56), by making $d\mathcal{F}^{\mathscr{j}}(\mathcal{J}^{\mathscr{j}})/\, d\mathcal{J}^{\mathscr{j}} = 0$. The explicit optimal solution $\hat{\mathcal{J}}^{\mathscr{j}}$ is given below in Equation (57).

$$\hat{\mathcal{J}}^{\mathscr{j}} = \frac{1}{|\mathcal{H}_{\hat{m}}|}\sum_{l=1}^{|\mathcal{H}_{\hat{m}}|}\left(\mathscr{p}_{n}^{l} - \mathscr{p}_{\hat{m}}^{l}\right), \forall \mathscr{j} = 1, \dots, \mathcal{N} \tag{57}$$

Then the corresponding optimal objective function is denoted as $\hat{\mathcal{F}}^{\mathscr{j}}$. Accordingly, we will obtain an optimal motion vector $\hat{\mathcal{J}}^{\mathscr{j}} = (-\hat{d}^{\mathscr{j}}, \hat{t}^{\mathscr{j}})$ with objective value $\hat{\mathcal{F}}^{\mathscr{j}}$ for each possible data pair $\mathscr{j}$, $\forall \mathscr{j} = 1, \dots, \mathcal{N}$. We find the best data points pair $\mathscr{j}$ which minimizes these objective values $\mathcal{F}^{\mathscr{j}}(*)$ in Equation (58).

$$\mathscr{j}^{*} = \operatorname{argmin}\{\mathcal{F}^{\mathscr{j}}(*), \forall \mathscr{j} = 1, \dots, \mathcal{N}\} \tag{58}$$

Wrapping results above, we obtain the newly learned distance and time displacement $d(k+1)$ and $t(k+1)$ as follows in Equation (59).

$$d(k+1) = \hat{d}^{\mathscr{j}^{*}}; t(k+1) = \hat{t}^{\mathscr{j}^{*}} \tag{59}$$

The pseudo-code of the adaptive curve learning algorithm is given in the Appendix-VI. 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 8.

8. 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 set up on a sample mixed flow platoon shown in Figure 1. It is a platoon with 10 vehicles: 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.

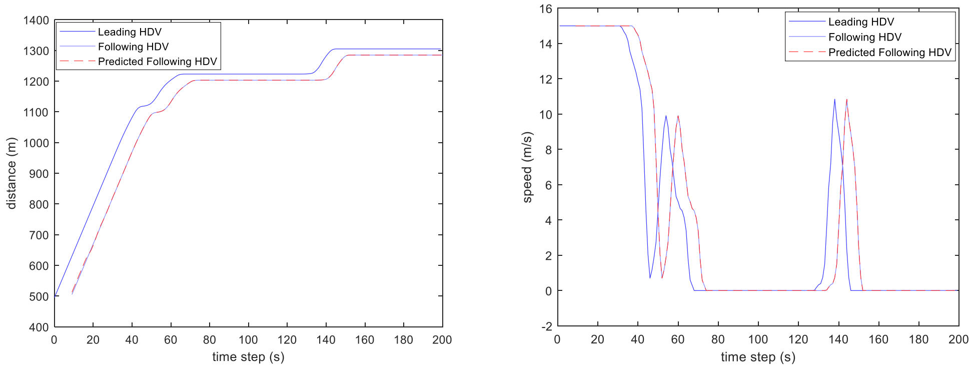
In Table 1, the spacing and speed penalty weight α_i and β_i are different in MPC controllers and the switching signal MINLP-σ_0. The MPC controllers assigns different weights to different spacing and speed errors for platoon's good stability performance. However, this setting is not proper for the MINLP-σ_0, 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.

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 ε_i	[0.2, 0.3] (Montanaro et al, 2020)
Powertrain lag σ_i	[0.1, 0.2] (Automobile drag, Wikipedia)
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
Penalty weight ω_2	$N^2 * P^2$
Penalty weight α_i	$0.3N^2$
Penalty weight β_i	$0.4N^2$
Splitting spacing difference $\mathcal{D}$	200 (m)
Splitting speed difference $\mathcal{D}^{'}$	10 (m/s)

8.1. 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. The results are summarized in Figure 5 and Table 2 below.



(a). distance – time (b). speed – time

Figure 5. Comparison of the predicted and actual trajectory of HDV

Figure 5 shows that the adaptive curve learning algorithm takes less than 20 seconds for warm-up. After that it accurately predicted the following HDV's trajectory (i.e., the predicted trajectory is almost coincided with the actual field trajectory in both location and speed). The average prediction error in the location and speed are 0.1255 (m) and 0.0511 (m/s) respectively. Besides, the average computation time at each time step is less than 0.002 sec. It thus fits this study very well.

Table 2. Solution approach computation performance (sec)

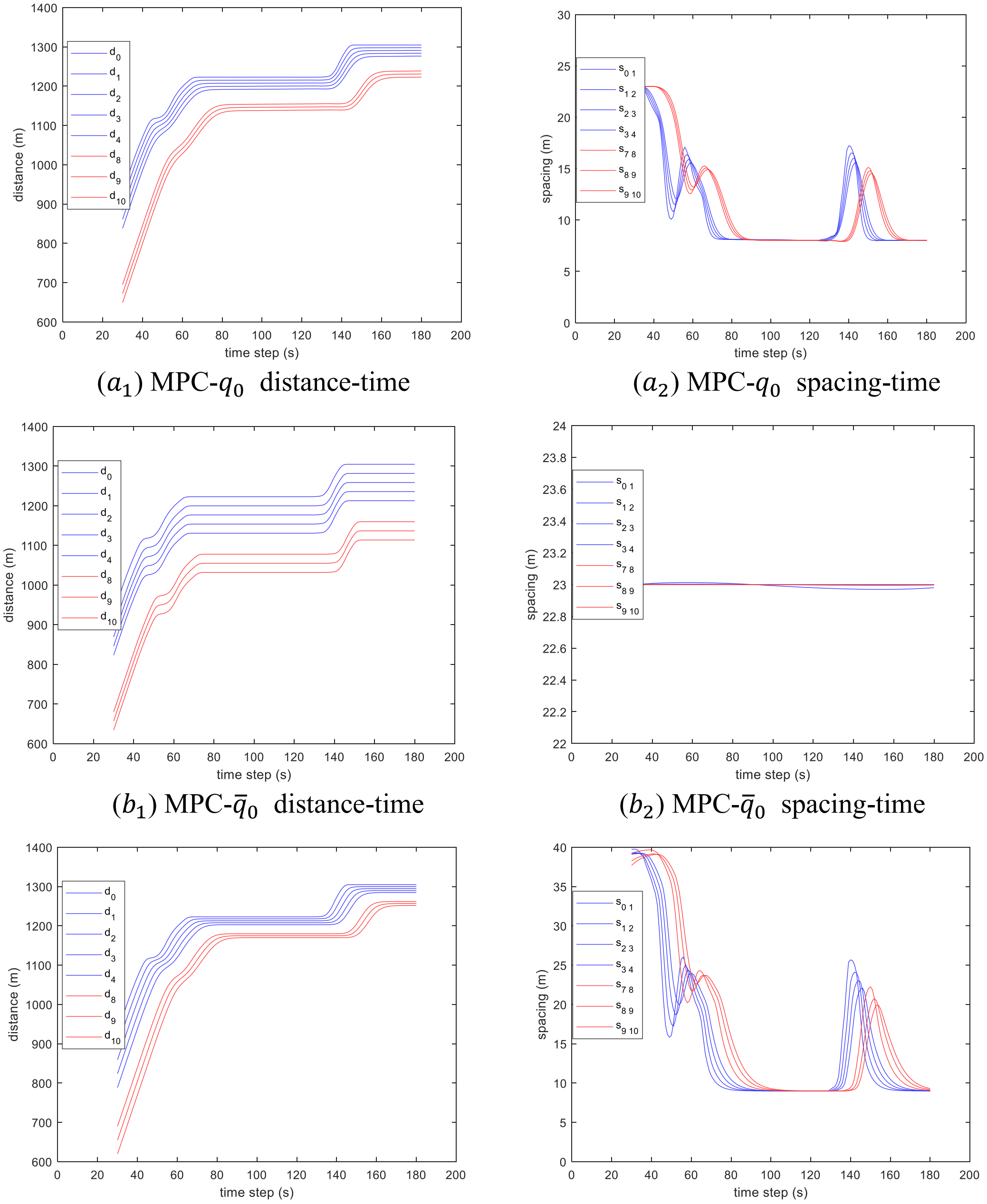
AS-OCD algorithm							DBB algorithm	
MPC horizon	MPC-q_0		MPC-q_1		MPC-q_2		σ_0 MINLP	
	Mean	Varianc e	Mean	Variance	Mean	Variance	Mean	Variance
$P = 30$	0.0762	0.00557	0.0732	0.00632	0.0621	0.00739	0.1421	0.04889
$P = 40$	0.0841	0.00724	0.0923	0.00892	0.0753	0.00772	0.1792	0.05223
$P = 50$	0.1027	0.01721	0.1152	0.01723	0.1022	0.01643	0.2232	0.06729
$P = 60$	0.1857	0.04632	0.1998	0.04012	0.1908	0.05042	0.2988	0.07011

Table 2 demonstrates that the AS-OCD algorithm can efficiently solve the MPC controllers using the computation time less than 0.2 sec. This performance satisfies the computation requirement (< 1 sec, the control interval). Please note that it needs more computation time as the prediction horizon P of the MPC controllers increases. Table 2 also shows that the DBB algorithm can solve the MINLP-σ_0 optimizer within 0.3 sec, which also meets the computation efficiency demand and can ensure the control smoothness and continuity of the hybrid MPC system.

8.2. Traffic flow smoothness and throughputs

Next, this study demonstrates the benefits on 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 Figure 6, Figure 7 and Figure 8.

Specifically, Figure 6 (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. Figure 6 (a_2, b_2) 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, Figure 6 (a_1, b_1) 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 Figure 6 (c_1, c_2) also enables adaptive spacing, Figure 6 (a_2, c_2) 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.



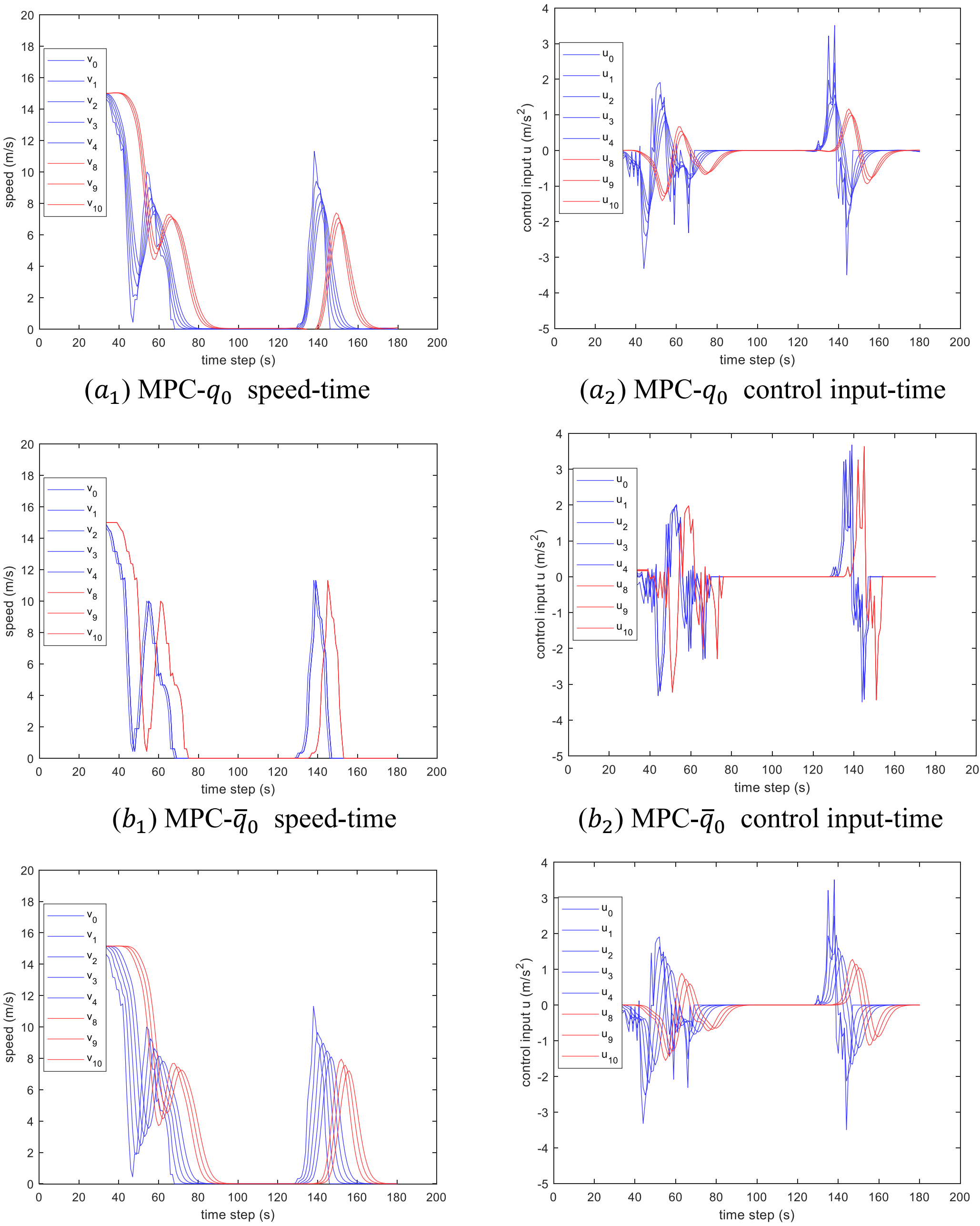
(a_1) MPC-q_0 distance-time (a_2) MPC-q_0 spacing-time

(b_1) MPC-$\bar{q}_0$ distance-time (b_2) MPC-$\bar{q}_0$ spacing-time

(c_1) CACC distance-time (c_2) CACC spacing-time

Figure 6. Distance-time and spacing-time trajectories (MPC-q_0, MPC-$\bar{q}_0$, CACC)

Figure 7 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 Figure 7 (a_1,a_2) are milder than the MPC-$\bar{q}_0$ with constant desired spacing policy in Figure 7 (b_1,b_2) and the CACC controller in Figure 7 (c_1,c_2). The mild speed and control input variations can improve driving comfort, and more importantly lead to energy consumption and emission reductions. The results in Figure 6 and Figure 7 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.



(a_1) MPC-q_0 speed-time

(a_2) MPC-q_0 control input-time

(b_1) MPC-$\bar{q}_0$ speed-time

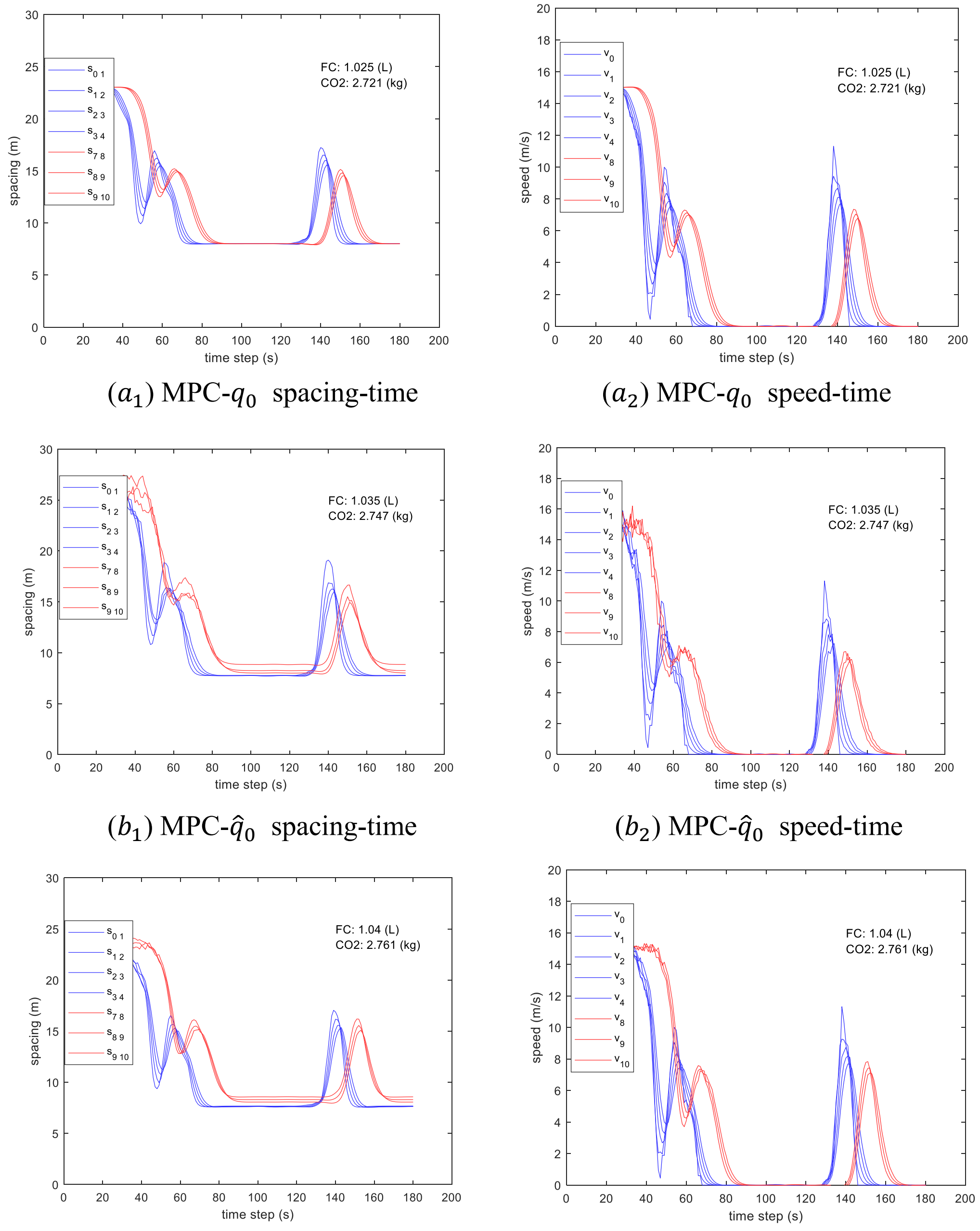
(b_2) MPC-$\bar{q}_0$ control input-time

(c_1) CACC speed-time

(c_2) CACC control input-time

Figure 7. Speed-time and control input-time trajectories (MPC-q_0, MPC-$\bar{q}_0$, CACC)

Figure 8 (a_1, a_2), (b_1, b_2) and (c_1, c_2) respectively illustrate vehicles' spacing and speed variations under the MPC-q_0 control, MPC-$\hat{q}_0$ control that ignores CAV uncertainty and MPC-$\tilde{q}_0$ control without online learning to predict HDV driving behaviors. It seeks to demonstrate the benefit on improving the traffic efficiency obtained from using the MPC controllers (e.g., MPC-q_0) that factors CAV control uncertainties (i.e., Equation (3)) and HDV online adaptive learning. Specifically, vehicle trajectories under MPC-q_0 in Figure 8 (a_1,a_2) are smoother than MPC-$\hat{q}_0$ in Figure 8 (b_1, b_2) and MPC-$\tilde{q}_0$ in Figure 8 (c_1, c_2). It results from the mismatches between the real and the mathematical CAV and HDV dynamics ignoring uncertainties. In practice, these mismatches are small and can be fixed by feedback controller such as MPC at every control step so that the platoon can be bounded around the steady-state (Input-to-State stability) shown in Figure 8 (b_1, b_2) and (c_1, c_2). However, it causes extra traffic oscillations and thus wastes fuel consumption and emission by 1% in Figure 8 (b_1, b_2) and 1.5% in Figure 8 (c_1, c_2) compared with results in Figure 8 (a_1, a_2) that factor the uncertainties.[11]



(a_1) MPC-q_0 spacing-time

(a_2) MPC-q_0 speed-time

(b_1) MPC-$\hat{q}_0$ spacing-time

(b_2) MPC-$\hat{q}_0$ speed-time

(c_1) MPC-$\tilde{q}_0$ spacing-time

(c_2) MPC-$\tilde{q}_0$ speed-time

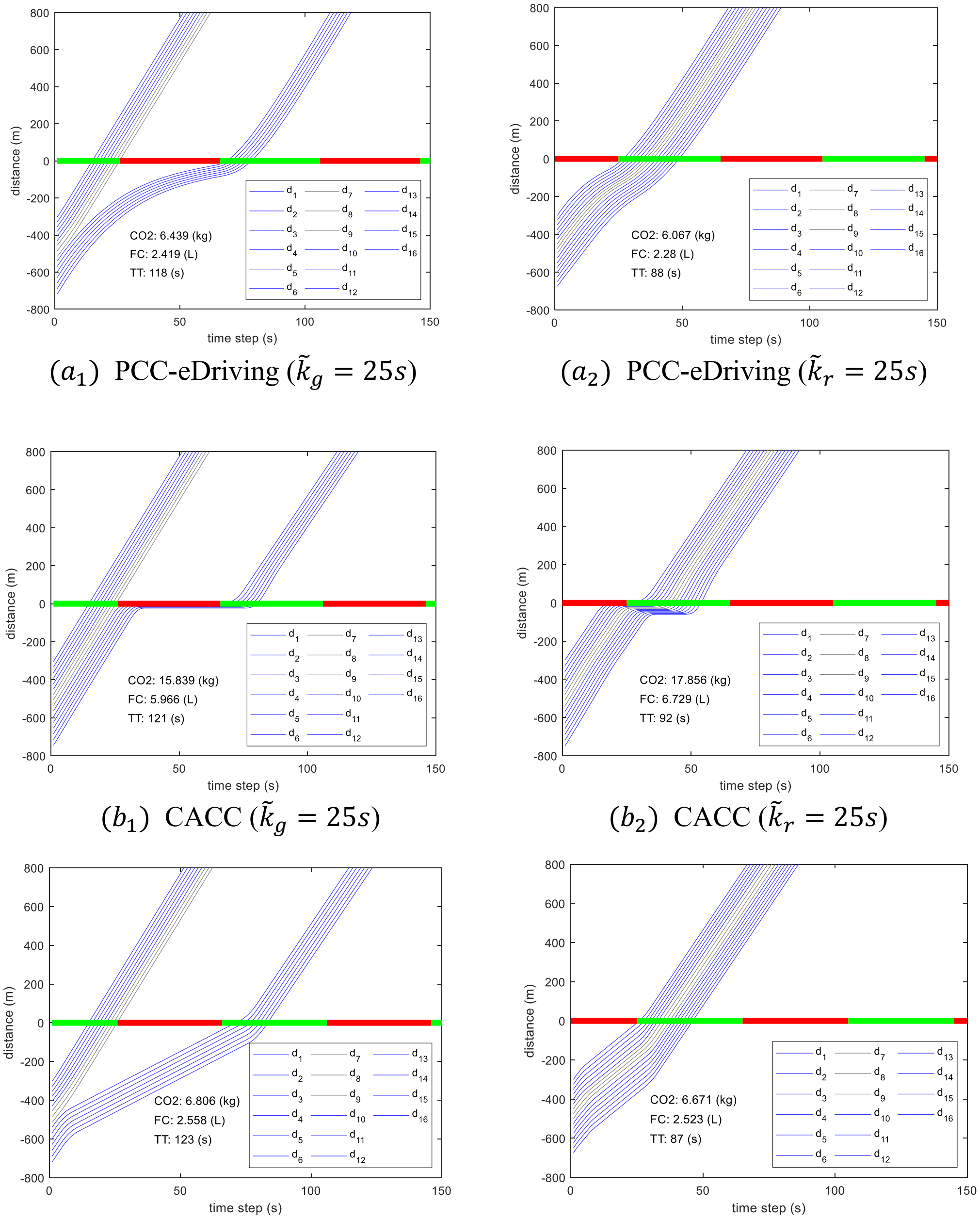
Figure 8. Spacing-time and speed-time trajectories (MPC-q_0, MPC-$\hat{q}_0$, MPC-$\tilde{q}_0$)

8.3. Performance of the PCC-eDriving control

This study finally validates the traffic performance of the PCC-eDriving control for guiding a

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

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. Figure 9 (a_1, b_1, c_1) provide the results under green scenario, where we set the remain green interval as $\tilde{k}_g = 25s$, when the mixed flow platoon arrives at the communication zone (r =300 m away from the intersection). Figure 9 (a_2, b_2, c_2) demonstrate the results under the red scenario, where we set there remains red interval as $\tilde{k}_r = 25s$. 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).



(a_1) PCC-eDriving ($\tilde{k}_g = 25s$) (a_2) PCC-eDriving ($\tilde{k}_r = 25s$)

(b_1) CACC ($\tilde{k}_g = 25s$) (b_2) CACC ($\tilde{k}_r = 25s$)

(c_1) PAVSOS ($\tilde{k}_g = 25s$) (c_2) PAVSOS ($\tilde{k}_g = 25s$)

Figure 9. Distance-time trajectory (PCC-eDriving, CACC and PAVSOS)

Figure 9 (a_1, a_2) 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 Figure 9 (b_1, b_2) and the PAVSOS control in Figure 9 (c_1, c_2) under both green and red scenarios. More exactly, under the the green scenario, Figure 9 (a_1) 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 consumptions and CO2 emissions for the entire trip, compared with CACC control in Figure 9 (b_1), which led to sharp deceleration at the intersection and consequently consumed more fuel and produced more CO2 emission. Compared with the PAVSOS control in Figure 9 (c_1), 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 it generated smoother vehicle trajectories. Under the red scenario, Figure 9 (a_2) 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 Figure 9 (b_2) and PAVSOS control in Figure 9 (c_2). 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.

9. 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. To ensure the control continuity and robustness, the hybrid MPC system's sequential/switching feasibility and control stability are theoretically proved. 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 second. 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 our study to consider left or right-turn instead of only going-through traffic at the intersection. This extension will involve both longitude and latitude trajectory control and tremendously complicate the mathematical modeling and algorithm design. We propose to address these extensions in our future study.

Acknowledgments

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

Reference

Acceleration Parameters, Police Radar Information Center. (n.d.), viewed 13 September 2020, Retrieved from <https://copradar.com/chapts/references/acceleration.html>

Ahn, K., Rakha, H., Trani, A. and Van Aerde, M., 2002. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *Journal of transportation engineering*, *128*(2), pp.182-190.

Androulakis, I.P. and Floudas, C.A., 1999. Distributed branch and bound algorithms for global optimization. In *Parallel processing of discrete problems* (pp. 1-35). Springer, New York, NY.

Alsabaan, M., Naik, K., Khalifa, T., 2013. Optimization of fuel cost and emissions using v2v communications. IEEE Transactions on intelligent transportation systems 14 (3), 1449–1461.

Asadi, B. and Vahidi, A., 2010. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE transactions on control systems technology*, *19*(3), pp.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>

Bemporad, A., Heemels, W.M.H. and De Schutter, B., 2002. On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, *47*(5), pp.863-869.

Borrelli, F., Bemporad, A. and Morari, M., 2017. *Predictive control for linear and hybrid systems*. Cambridge University Press.

Bof, N., Carli, R. and Schenato, L., 2018. Lyapunov theory for discrete time systems. *arXiv preprint arXiv:1809.05289*.

Boyd, S., Boyd, S.P. and Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press.

Chakroborty, P. and Kikuchi, S., 1999. Evaluation of the General Motors based car-following models and a proposed fuzzy inference model. *Transportation Research Part C: Emerging Technologies*, *7*(4), pp.209-235.

Chen, C., Wang, J., Xu, Q., Wang, J. and Li, K., 2021. Mixed platoon control of automated and human-driven vehicles at a signalized intersection: dynamical analysis and optimal control. *Transportation Research Part C: Emerging Technologies*, *127*, p.103138.

Chen, J., Zhou, Y. and Liang, H., 2019. Effects of ACC and CACC vehicles on traffic flow based on an improved variable time headway spacing strategy. *IET Intelligent Transport Systems*, *13*(9), pp.1365-1373.

Conejo, A.J., Nogales, F.J. and Prieto, F.J., 2002. A decomposition procedure based on approximate Newton directions. *Mathematical programming*, *93*(3), pp.495-515.

Faraj, M., Sancar, F.E. and Fidan, B., 2017, June. Platoon-based autonomous vehicle speed optimization near signalized intersections. In *2017 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1299-1304). IEEE.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G. and Diehl, M., 2014. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, *6*(4), pp.327-363.

Feng, Y., Yu, C. and Liu, H.X., 2018. Spatiotemporal intersection control in a connected and automated vehicle environment. *Transportation Research Part C: Emerging Technologies*, *89*, pp.364-383.

Gong, S. and Du, L., 2018. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation research part B: methodological*, *116*, pp.25-61.

Greub, W.H., 2012. *Linear algebra* (Vol. 23). Springer Science & Business Media.

Guo, Y., Ma, J., Xiong, C., Li, X., Zhou, F. and Hao, W., 2019. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation research part C: emerging technologies*, *98*, pp.54-72.

Guo, Q., Angah, O., Liu, Z. and Ban, X.J., 2021. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transportation Research Part C: Emerging Technologies*, *124*, p.102980.

Hall, F.L., 1996. Traffic stream characteristics. *Traffic Flow Theory. US Federal Highway Administration*, *36*.

Hu, Y., Li, W., Xu, K., Zahid, T., Qin, F. and Li, C., 2018. Energy management strategy for a hybrid electric vehicle based on deep reinforcement learning. *Applied Sciences*, *8*(2), p.187.

Kamal, M.A.S., Mukai, M., Murata, J. and Kawabe, T., 2012. Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on control systems technology*, *21*(3), pp.831-841.

Lee, H., Kim, N. and Cha, S.W., 2020. Model-based reinforcement learning for eco-driving control of electric vehicles. *IEEE Access*, *8*, pp.202886-202896.

Li, L.H., Gan, J. and 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. *Journal of Advanced Transportation*, *2018*.

Lioris, J., Pedarsani, R., Tascikaraoglu, F.Y. and Varaiya, P., 2016. Doubling throughput in urban roads by platooning. *IFAC-PapersOnLine*, *49*(3), pp.49-54.

Löfberg, J., 2012. Oops! I cannot do it again: Testing for recursive feasibility in MPC. *Automatica*, *48*(3), pp.550-555.

Ma, F., Yang, Y., Wang, J., Li, X., Wu, G., Zhao, Y., Wu, L., Aksun-Guvenc, B. and Guvenc, L., 2021. Eco-driving-based cooperative adaptive cruise control of connected vehicles platoon at signalized intersections. *Transportation Research Part D: Transport and Environment*, *92*, p.102746.

Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O., 2000. Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), pp.789-814.

Montanaro, U., Wroblewski, M., Dixit, S., Creighton, S., Pragalathan, S. and Sorniotti, A., 2020. Linearising Longitudinal Vehicle Dynamics through Adaptive Control Techniques for Platooning Applications. *International Journal of Powertrains*.

Nak, H., Akkaya, Ş. and Yumuk, E., Active set method based model predictive control for a ball and beam system. In *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 871-875). IEEE.

Newell, G.F., 2002. A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, *36*(3), pp.195-205.

Nie, Z. and Farzaneh, H., 2021. Role of Model Predictive Control for Enhancing Eco-Driving of Electric Vehicles in Urban Transport System of Japan. *Sustainability*, *13*(16), p.9173.

Niroumand, R., Tajalli, M., Hajibabai, L. and Hajbabaie, A., 2020. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. *Transportation research part C: emerging technologies*, *116*, p.102659.

Nocedal, J. and Wright, S., 2006. *Numerical optimization*. Springer Science & Business Media.

Öncü, S., Ploeg, J., Van de Wouw, N. and Nijmeijer, H., 2014. Cooperative adaptive cruise control: Network-aware analysis of string stability. *IEEE Transactions on Intelligent Transportation Systems*, *15*(4), pp.1527-1537.

Ploeg, J., Scheepers, B.T., Van Nunen, E., Van de Wouw, N. and Nijmeijer, H., 2011, October. Design and experimental evaluation of cooperative adaptive cruise control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 260-265). IEEE.

Punzo, V. and Simonelli, F., 2005. Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. *Transportation Research Record*, *1934*(1), pp.53-63.

Qu, X., Yu, Y., Zhou, M., Lin, C.T. and Wang, X., 2020. Jointly dampening traffic oscillations and improving energy consumption with electric, connected and automated vehicles: a reinforcement learning based approach. *Applied Energy*, *257*, p.114030.

Rakha, H., Kang, Y.S. and Dion, F., 2001. Estimating vehicle stops at undersaturated and oversaturated fixed-time signalized intersections. *Transportation Research Record*, *1776*(1), pp.128-137.

Sastry, S. and Bodson, M., 2011. *Adaptive control: stability, convergence and robustness*. Courier Corporation.

Shen, J., Kammara, E.K.H. and Du, L., 2021. Nonconvex, Fully Distributed Optimization based CAV Platooning Control under Nonlinear Vehicle Dynamics. *arXiv preprint arXiv:2104.08713*.

Shi, J., Qiao, F., Li, Q., Yu, L. and Hu, Y., 2018. Application and evaluation of the reinforcement learning approach to eco-driving at intersections under Infrastructure-to-Vehicle communications. *Transportation Research Record*, *2672*(25), pp.89-98.

Stevanovic, A., Stevanovic, J. and Kergaye, C., 2013. Green light optimized speed advisory systems: Impact of signal phasing information accuracy. *Transportation research record*, *2390*(1), pp.53-59. Simchon, L. and Rabinovici, R., 2020. Real-time implementation of green light optimal speed advisory for electric vehicles. *Vehicles*, *2*(1), pp.35-54.

Sohrab, H.H., 2003. *Basic real analysis* (Vol. 231). Boston, Basel, Berlin: Birkhäuser.

Sun, C., Guanetti, J., Borrelli, F. and Moura, S.J., 2020. Optimal eco-driving control of connected and autonomous vehicles through signalized intersections. *IEEE Internet of Things Journal*, *7*(5), pp.3759-3773.

Wan, N., Vahidi, A., Luckow, A., 2016. Optimal speed advisory for connected vehicles in arterial roads and the impact on mixed traffic. Transportation Research Part C: Emerging Technologies 69, 548–563.

Wang, Z., Wu, G. and Barth, M.J., 2019. Cooperative eco-driving at signalized intersections in a partially connected and automated vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, *21*(5), pp.2029-2038.

Wei, Y., Avcı, 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. Transportation research part B: methodological 106, 102–129.

Wu, C., Xu, Z., Liu, Y., Fu, C., Li, K. and Hu, M., 2020. Spacing policies for adaptive cruise control: A survey. *IEEE Access*, *8*, pp.50149-50162.

Xiuzheng, Z., Liguo, Z. and Kholodov, Y., 2015, July. Model predictive control of eco-driving for transit using V2I communication. In *2015 34th Chinese Control Conference (CCC)* (pp. 2511-2516). IEEE.

Yang, H., Jin, W.-L., 2014. A control theoretic formulation of green driving strategies based on inter-vehicle communications. Transportation Research Part C: Emerging Technologies 41, 48–60.

Zeilinger, M.N., Jones, C.N., Raimondo, D.M. and Morari, M., 2009, December. Real-time MPC-Stability through robust MPC design. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference* (pp. 3980-3986). IEEE.

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. Transportation Research Part C: Emerging Technologies 95, 802–821.

Appendix

Appendix-I. We present the reference speed controller for the leading CAV $i = 1$ as follows in Equation (60):

$$\textbf{Min}\ \Gamma(u_1) = \sum_{p=1}^{P} \left\{ \tau \|v_1(p) - v_r\|_Q^2 + \frac{\tau^2}{2} \omega_1 \|u(p-1)\|_2^2 \right\} \tag{60}$$

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

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. Mathematical derivations of Equations (38)-(40).

According to the control dynamics in Equations (10)-(13) and CAV dynamics in Equations (1)-(3), we can have the following mathematical derivations of the CAV i's control dynamics in the Equations (61) and (62). Note that we denote $a_i(k) = u_i(k) - \Delta u_i(k)$ for notation simplicity.

$$\begin{aligned} z_i(k+1) &= x_{i-1}(k+1) - x_i(k+1) - s_i(k+1) \\ &= x_{i-1}(k) - x_i(k) + \tau\big(v_{i-1}(k) - v_i(k)\big) + \frac{\tau^2}{2}\big(a_{i-1}(k) - a_i(k)\big) - s_i(k+1) \\ &= z_i(k) + s_i(k) - s_i(k+1) + \tau z_i'(k) + \frac{\tau^2}{2}\big(a_{i-1}(k) - a_i(k)\big) \\ &= z_i(k) + \tau z_i'(k) - \delta_1 \tau^2 a_i(k) - \delta_2 \tau^2 \big(a_{i-1}(k) - a_i(k)\big) + \frac{\tau^2}{2}\big(a_{i-1}(k) - a_i(k)\big) \\ &= z_i(k) + \tau z_i'(k) + \tau^2 \left(\frac{1}{2} + \delta_2\right) a_{i-1}(k) - \tau^2 \left(\delta_1 + \delta_2 + \frac{1}{2}\right) a_i(k) \end{aligned} \tag{61}$$

$$\begin{aligned} z_i'(k+1) &= v_{i-1}(k+1) - v_i(k+1) \\ &= v_{i-1}(k) + \tau a_{i-1}(k) - \big(v_i(k) + \tau a_i(k)\big) \\ &= z_i'(k) + \tau\big(a_{i-1}(k) - a_i(k)\big) \end{aligned} \tag{62}$$

Let $c_1 = \delta_2 + \frac{1}{2}$; $c_2 = -\left(\delta_1 + \delta_2 + \frac{1}{2}\right)$. We have the following Equation (63) according to Equations (61) and (62).

$$\begin{bmatrix} z_i(k+1) \\ z_i'(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_i(k) \\ z_i'(k) \end{bmatrix} + \begin{bmatrix} \tau^2 c_1 & \tau^2 c_2 \\ \tau & -\tau \end{bmatrix} \begin{bmatrix} a_{i-1}(k) \\ a_i(k) \end{bmatrix} \tag{63}$$

According to the control dynamics of CAV i in Equation (63), we can then derive out the following platoon control dynamics in Equation (64),

$$\begin{bmatrix} z(k+1) \\ z'(k+1) \end{bmatrix} = \begin{bmatrix} I_N & \tau I_N \\ 0 & I_N \end{bmatrix} \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} + \begin{bmatrix} \tau^2 S_1 \\ \tau S_2 \end{bmatrix} a(k) \tag{64}$$

where $a(k) = [a_0(k) \quad \dots \quad a_i(k) \quad \dots \quad a_N(k)]^T \in \mathbb{R}^{N+1}$ and the matrices S_1 and S_2 are given below with $c_1 = \delta_2 + \frac{1}{2}$; $c_2 = -\left(\delta_1 + \delta_2 + \frac{1}{2}\right)$.

$$S_1 = \begin{bmatrix} c_1 & c_2 & & & \\ & c_1 & c_2 & & \\ & & \ddots & \ddots & \\ & & & c_1 & c_2 \end{bmatrix} \in \mathbb{R}^{N\times(N+1)};\ S_2 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{N\times(N+1)}$$

Appendix-III. The mathematical formulations of $\overline{K}_i$, $\overline{\Delta}_i$, f_i are presented as follows according to Equations (47) and (49).

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

Appendix-IV. Proof of Condition (iii) in **Lemma 3**.

To prove condition (iii) $\rho^* = \rho\left(I - \left(\overline{K}\right)^{-1} K\right) < 1$ holds, we first write the mathematical formulations of the matrices K and $\overline{K}$ in Equation (65). Note that the objective function $\Gamma(u)$ in Equation (46) is locally coupled, namely CAV i's control input u_i is only coupled with two control inputs u_{i-1}, u_{i+1} of its immediate neighboring CAVs. According to this feature, we have $\frac{\partial^2 \Gamma}{\partial u_i u_j} = 0$ if $|i-j| \geq 2$.

$$K = \begin{bmatrix} \frac{\partial^2\Gamma}{\partial u_1^2} & \frac{\partial^2\Gamma}{\partial u_1 u_2} & 0 & \dots & 0 \\ \frac{\partial^2\Gamma}{\partial u_2 u_1} & \frac{\partial^2\Gamma}{\partial u_2^2} & \frac{\partial^2\Gamma}{\partial u_2 u_3} & \ddots & \vdots \\ 0 & \frac{\partial^2\Gamma}{\partial u_3 u_2} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \frac{\partial^2\Gamma}{\partial u_{N-1}^2} & \frac{\partial^2\Gamma}{\partial u_{N-1} u_N} \\ 0 & \dots & 0 & \frac{\partial^2\Gamma}{\partial u_N u_{N-1}} & \frac{\partial^2\Gamma}{\partial u_N^2} \end{bmatrix} \overline{K} = \begin{bmatrix} \frac{\partial^2\Gamma}{\partial u_1^2} & 0 & 0 & \dots & 0 \\ 0 & \frac{\partial^2\Gamma}{\partial u_2^2} & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & \frac{\partial^2\Gamma}{\partial u_{N-1}^2} & 0 \\ 0 & \dots & 0 & 0 & \frac{\partial^2\Gamma}{\partial u_N^2} \end{bmatrix}. \tag{65}$$

It is noted that matrices K and $\overline{K}$ in Equation (65) do not consider the constraints in Equations (6)-(8) and (14) because they are not active under normal traffic conditions. Besides, involving these constraints will introduce Lagrangian multipliers η and λ into the Lagrangian function L_{AS} in Equations (47)-(49). Given multipliers η and λ can be any negative or positive value, the matrices K and $\overline{K}$ in Equations (47)-(49) are hard to bound. Consequently, theoretically proving condition (iii) becomes mathematically intractable according to Conejo et al., (2002).

According to Equation (65), we can sequentially derive the matrix $\left(\overline{K}\right)^{-1}K$ in Equation (66) and finally the matrix $I-\left(\overline{K}\right)^{-1}K$ in Equation (67) below.

$$\left(\overline{K}\right)^{-1}K=\begin{bmatrix} I_P & A_{1,2} & 0 & \dots & 0 \\ A_{2,1} & I_P & A_{2,3} & \ddots & \vdots \\ 0 & A_{3,2} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & I_P & A_{N-1,N} \\ 0 & \dots & 0 & A_{N,N-1} & I_P \end{bmatrix}, \tag{66}$$

where $A_{i,i-1}=\left(\frac{\partial^2\Gamma}{\partial u_i^2}\right)^{-1}\left(\frac{\partial^2\Gamma}{\partial u_i u_{i-1}}\right)\in\mathbb{R}^{P\times P}, i=2,\dots,N;$ $A_{i,i+1}=\left(\frac{\partial^2\Gamma}{\partial u_i^2}\right)^{-1}\left(\frac{\partial^2\Gamma}{\partial u_i u_{i+1}}\right)\in\mathbb{R}^{P\times P}, i=1,\dots,N-1$.

$$B=\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{N-1} \\ B_N \end{bmatrix}=I-\left(\overline{K}\right)^{-1}K=\begin{bmatrix} 0 & -A_{1,2} & 0 & \dots & 0 \\ -A_{2,1} & 0 & -A_{2,3} & \ddots & \vdots \\ 0 & -A_{3,2} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & -A_{N-1,N} \\ 0 & \dots & 0 & -A_{N,N-1} & 0 \end{bmatrix}, \tag{67}$$

Note that we use $B=I-\left(\overline{K}\right)^{-1}K$ hereafter for discussion convenience. To prove condition (III) holds, it is equivalent to showing the spectral radius of matrix B is less than 1. We intend to use the induced norm inequality[12] to prove it. Specifically, we first calculate the 1-norm of matrix B (i.e, $\|B\|$) and show $\|B\|$ is less than 1. It is equivalent to showing the summation of the row elements' absolute values is less than 1 for every row of matrix B. For discussion convenience, we denote $B=[B_1 \quad B_2 \quad \dots \quad B_{N-1} \quad B_N]^T$, where $B_i\in\mathbb{R}^{P\times NP}$ represents i^{th} row block of matrix B with P rows. B_i corresponds to CAV i's P-step matrix block in the optimizer $\Gamma(u)$. Further, we use the notation $B_{i,p}$ to represent the p^{th} row of the matrix block B_i, $\forall p=1,\dots,P$. $B_{i,p}$ corresponds to CAV i's matrix block at time step p. Wrapping above, if the 1-norm of $B_{i,p}$ (i.e., $\|B_{i,p}\|$) is less than 1 for $\forall i=1,\dots,N, \forall p=1,\dots,P$, condition (III) holds. Below we show $\|B_{i,p}\|$ is always less than 1.

To do that , we first have $B_i, \forall i=1,\dots,N$ represented as follows in Equation (68).

$$B_i=\begin{cases} -A_{i,i+1}=-\left(\dfrac{\partial^2\Gamma}{\partial u_i^2}\right)^{-1}\dfrac{\partial^2\Gamma}{\partial u_i u_{i+1}} & if\ i=1 \\ -A_{i,i-1}-A_{i,i+1}=-\left(\dfrac{\partial^2\Gamma}{\partial u_i^2}\right)^{-1}\left(\dfrac{\partial^2\Gamma}{\partial u_i u_{i-1}}+\dfrac{\partial^2\Gamma}{\partial u_i u_{i+1}}\right) & if\ i\neq 1,N \\ -A_{i,i-1}=-\left(\dfrac{\partial^2\Gamma}{\partial u_i^2}\right)^{-1}\dfrac{\partial^2\Gamma}{\partial u_i u_{i-1}} & if\ i=N \end{cases} \tag{68}$$

[12] For a matrix A, $\rho(A)\le\|A^r\|^{1/r}$ for all positive integers r, where $\rho(A)$ is the spectral radius of A. When $r=1$, $\rho(A)\le\|A\|$

According to Equation (68), we can obtain $\|B_{i,p}\|$ in Equation (69) for each row $p_1 \in \{1, \ldots, P\}$ in B_i.

$$\|B_{i,p}\| = \begin{cases} \sum_{p_2=1}^{P} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right|^{-1} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}\right| & i = 1 \\ \sum_{p_2=1}^{P} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right|^{-1} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)} + \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}\right| & i \neq 1, N \\ \sum_{p_2=1}^{P} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right|^{-1} \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)}\right| & i = N \end{cases} \quad (69)$$

Consequently, to prove $\|B_{i,p}\| < 1$, we need to derive out the mathematical representation of $\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}, \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)}, \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}$ and show each element of $\|B_{i,p}\|$ is strictly less than 1. To do that, we calculate the second derivatives $\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}$, $\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)}$ and $\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}$ respectively in Equations (70), (71) and (72), according to Equation (15).

$$\begin{aligned} \frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)} = \sum_{p=max\{p_1,p_2\}}^{P} \Bigg[\alpha_i \frac{\partial z_i(p)}{\partial u_i(p_1)}\frac{\partial z_i(p)}{\partial u_i(p_2)} + \beta_i \frac{\partial z_i'(p)}{\partial u_i(p_1)}\frac{\partial z_i'(p)}{\partial u_i(p_2)} \\ + \alpha_{i+1}\frac{\partial z_{i+1}(p)}{\partial u_i(p_1)}\frac{\partial z_{i+1}(p)}{\partial u_i(p_2)} + \beta_{i+1}\frac{\partial z_{i+1}'(p)}{\partial u_i(p_1)}\frac{\partial z_{i+1}'(p)}{\partial u_i(p_2)}\Bigg] + \omega_1\tau^2\varsigma(p_1,p_2), \end{aligned} \quad (70)$$

$$\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)} = \sum_{p=\max\{p_1,p_2\}}^{P} \left[\alpha_i \frac{\partial z_i(p)}{\partial u_i(p_1)}\frac{\partial z_i(p)}{\partial u_{i-1}(p_2)} + \beta_i \frac{\partial z_{\mathrm{i}}'(p)}{\partial u_i(p_1)}\frac{\partial z_{\mathrm{i}}'(p)}{\partial u_{i-1}(p_2)}\right], \quad (71)$$

$$\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)} = \sum_{p=max\{p_1,p_2\}}^{P} \left[\alpha_{i+1} \frac{\partial z_{i+1}(p)}{\partial u_i(p_1)}\frac{\partial z_{i+1}(p)}{\partial u_{i+1}(p_2)} + \beta_{i+1} \frac{\partial z_{i+1}'(p)}{\partial u_i(p_1)}\frac{\partial z_{i+1}'(p)}{\partial u_{i+1}(p_2)}\right], \quad (72)$$

In Equation (70), $\varsigma(p_1,p_2) = \begin{cases} 1 & \text{if } p_1 = p_2 \\ 0 & \text{otherwise} \end{cases}$ is an indicator function. In Equations (70)-(72), the first derivative $\frac{\partial z_i(p)}{\partial u_{i-1}(p_2)}$, $\frac{\partial z_i(p)}{\partial u_i(p_2)}$, $\frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_{i+1}(p_2)}$ and $\frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_i(p_2)}$ are formulated below in Equation (73) according to Equations (9)-(13).

$$\frac{\partial z_i(p)}{\partial u_{i-1}(p_2)} = \frac{\partial x_{i-1}(p)}{\partial u_{i-1}(p_2)} + \delta_2\tau\frac{\partial v_{i-1}(p)}{\partial u_{i-1}(p_2)}; \ \frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_i(p_2)} = \frac{\partial v_i(p)}{\partial u_i(p_2)} \quad (73)$$

$$\frac{\partial z_i(p)}{\partial u_i(p_2)} = -\frac{\partial x_i(p)}{\partial u_i(p_2)} - (\delta_1 + \delta_2)\tau\frac{\partial v_i(p)}{\partial u_i(p_2)}; \ \frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_{i+1}(p_2)} = -\frac{\partial v_{i+1}(p)}{\partial u_{i+1}(p_2)}$$

Without loss of generality, we consider CAVs' aerodynamic drag and powertrain lag coefficients are the same for discussion convenience so that we can have $\left|\frac{\partial v_i(p)}{\partial u_i(p_2)}\right| = \left|\frac{\partial v_{i+1}(p)}{\partial u_{i+1}(p_2)}\right|$ and $\left|\frac{\partial x_{i-1}(p)}{\partial u_{i-1}(p_2)}\right| = \left|\frac{\partial x_i(p)}{\partial u_i(p_2)}\right|$ according to Equations (1)-(3). Then we have the following relations regarding $\frac{\partial z_i(p)}{\partial u_{i-1}(p_2)}$, $\frac{\partial z_i(p)}{\partial u_i(p_2)}$, $\frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_{i+1}(p_2)}$ and $\frac{\partial z_{\mathrm{i+1}}'(p)}{\partial u_i(p_2)}$ in Equation (74) based on Equation (73). Note that if CAVs' aerodynamic drag and powertrain lag are different, $\left|\frac{\partial v_i(p)}{\partial u_i(p_2)}\right|$ and $\left|\frac{\partial x_i(p)}{\partial u_i(p_2)}\right|$ may be smaller or larger than $\left|\frac{\partial v_{i+1}(p)}{\partial u_{i+1}(p_2)}\right|$ and $\left|\frac{\partial x_{i-1}(p)}{\partial u_{i-1}(p_2)}\right|$ respectively. It will make the math relations in Equation (74)-(75) very complicated with a a large number of scenarios, and consequently lead to Equation (69) mathematically intractable. Apart from it, CAVs' aerodynamic drag and powertrain lag are normally small so that it is reasonable to consider them the same values without affecting the final results.

$$\left|\frac{\partial z'_{i+1}(p)}{\partial u_{i+1}(p_2)}\right| = \left|\frac{\partial z'_{i+1}(p)}{\partial u_i(p_2)}\right|;\ \left|\frac{\partial z_i(p)}{\partial u_{i-1}(p_2)}\right| < \left|\frac{\partial z_i(p)}{\partial u_i(p_2)}\right| \tag{74}$$

According to Equations (70)-(72) and (74), we have the following inequality formulations regarding $\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}, \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)}, \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}$ in Equation (75).

$$\begin{gathered}\left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}\right| < \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right| \\ \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)} + \frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i+1}(p_2)}\right| < \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right| \\ \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_{i-1}(p_2)}\right| < \left|\frac{\partial^2\Gamma}{\partial u_i(p_1)u_i(p_2)}\right|\end{gathered} \tag{75}$$

Based upon Equations (69) and (75), we thus have $\|B_{i,p}\| < 1, i \in N, p \in P$ Hence, we know that 1-norm of every row in matrix $I - (\overline{K})^{-1}K$ is smaller than 1, mathematically $\left\|I - (\overline{K})^{-1}K\right\| < 1$. Finally, we apply the induced norm theorem and obtain

$$\rho^* = \rho\left(I - (\overline{K})^{-1}K\right) \le \left\|I - (\overline{K})^{-1}K\right\| < 1.$$

Condition (III) is satisfied. We conclude the proof. #

Appendix-V. The error gains $e_D(k)$ and $e_T(k)$ in Equations (54) and (55) are calculated based on trajectory matching errors $e(k)$ using the following Equations (76)-(78),

$$e(k) = \widetilde{H}(k) - H(k), \tag{76}$$

$$e_D(k) = a_D e(k), \tag{77}$$

$$e_T(k) = \frac{a_T e(k)}{\max(v_{\widehat{m}}(k), 1)}, \tag{78}$$

where a_D and a_T are gain coefficients for distance error gain $e_D(k)$ and time error gain $e_T(k)$ respectively; $v_{\widehat{m}}(k)$ is the speed of HDV $\widehat{m}$ at step k; $H(k)$ is the actual trajectory point of a HDV $\widehat{m}$ at step k; $\widetilde{H}(k)$ is the predicted trajectory point of a HDV $\widehat{m}$ 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 $\widehat{m}$ at step k. Below we explain the technical details to develop Equations (76)-(78).

We consider two trajectories $H(k)$ for HDV $\widehat{m}$ 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 $\widehat{m}$ 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:$\left(k, C(k-T(k)) - D(k) = \widetilde{H}(k)\right)$. It is very likely that points A and C won't match exactly and generate the trajectory matching error $e(k)$ in Equation (76). 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 explan our main idea as follows. The same idea can be applied for the the case $e(k) < 0$. To make $\widetilde{H}(k)$ 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 Equations (77) and (78), 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., $10^{-3}\sim10^{-2}$) 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 $\mathcal{H}_{\hat{m}}(k+1) = \{\wp_{\hat{m}}^{l} = (x^{l}, t^{l})\}$, $\mathcal{C}_n(k+1) = \{\wp_{n}^{l} = (x^{l}, t^{l})\}$
    Determine $d(k+1), t(k+1), w(k+1)$ according to Equations (53),(56)-(59).
    Update $D(k+1), T(k+1), W(k+1)$ according to Equations (50), (53)-(54).
```