

Received March 31, 2022, accepted April 15, 2022, date of publication April 22, 2022, date of current version May 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169577

Delay-Aware Multi-Layer Multi-Rate Model Predictive Control for Vehicle Platooning Under Message-Rate Congestion Control

AMR IBRAHIM¹, DIP GOSWAMI¹, (Member, IEEE), HONG LI²,
AND TWAN BASTEN^{1,3}, (Senior Member, IEEE)

¹Department of Electrical Engineering, Eindhoven University of Technology, 5612 AP Eindhoven, The Netherlands

²RF Processing, NXP Semiconductors, 5656 AG Eindhoven, The Netherlands

³ESI, TNO, 5656 AE Eindhoven, The Netherlands

Corresponding author: Amr Ibrahim (a.ibrahim@tue.nl)

This work was supported by the European Union's Horizon 2020 Framework Program for Research and Innovation under Grant 674875 (oCPS—The Platform-aware Model-driven Optimization of Cyber-Physical Systems Program).

ABSTRACT Vehicle platooning is an enabler technology for increasing road capacity, improving safety and reducing fuel consumption. Platoon control is a two-layered system where each layer runs under a different communication standard and rate – (i) the upper-layer operates under a specific V2V communication standard such as IEEE 802.11p and (ii) the lower-layer operates over high-speed in-vehicle communication networks such as FlexRay, CAN. The upper-layer, under 802.11p, uses periodic Cooperative Awareness Messages (CAMs) for exchanging vehicle motion information (i.e., acceleration, velocity and so on), the rate of which is adapted depending on the network congestion level. With over 70% channel load, the CAMs experience significant delay and packet loss, jeopardizing the stability of the platoon control. Under such high congestion, the European Telecommunications Standard Institute (ETSI) proposes to engage Decentralized Congestion Control (DCC) to control the channel load. We propose a platoon control and DCC scheme to tackle this scenario. Our contribution is three-fold. First, we propose a multi-layer platoon model explicitly augmenting the communication delay in the state-space. Second, the augmented delay-aware platoon model is integrated in the state-of-the-art multi-layer multi-rate model predictive control (MPC) for the upper-layer. Third, we adopt a message-rate congestion control scheme to keep the channel load under a given threshold. We use the proposed delay-aware MPC scheme under the message-rate congestion control scheme which may lead to switching under dynamic network conditions. Using the proposed technique, we show that platoon performance can be maintained under high network congestion while maintaining string stability.

INDEX TERMS Vehicle platooning, multi-layer multi-rate control, V2V communication, communication delay, model predictive control, message-rate congestion control.

I. INTRODUCTION

Autonomous driving and connected mobility are important technologies for future Intelligent Transportation Systems (ITS) since they can improve safety, increase road capacity, reduce fuel consumption and emissions [30], [34]. A platoon of vehicles is a group of autonomous vehicles closely following each other while maintaining a safe inter-vehicle distance. Vehicle platooning is based on the Cooperative Adaptive Cruise Control (CACC) technology which

is an extension to Adaptive Cruise Control (ACC). Besides on-board sensors (radar or lidar) used in ACC for measuring the distance to the preceding vehicle [45], CACC integrates Vehicle-to-Vehicle (V2V) wireless communication between vehicles along with other sensors. This enhances the functionality of ACC and enables significant reduction in *headway* time (i.e., the time needed by the follower vehicle to reach the position of the preceding vehicle), improves safety and reduces fuel consumption. Wireless communication allows a richer set of information (acceleration, position, velocity, road intersection and traffic flow status such as existence of moving or stationary obstacles) to be shared between

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo¹.

vehicles. Many studies investigated the design and functionality of CACC e.g., [30], [47], [48], [51], [55], [63]. The effective capacity of highways might be improved by increasing CACC market penetration [59].

Vehicles in a platoon exchange periodic Cooperative Awareness Messages (CAMs) over V2V communication. There are two types of channels, as per IEEE 802.11p under the European Telecommunications Standards Institute (ETSI) [29], one control channel (CCH) and six service channels (SCHs). The CCH is dedicated to safety-critical applications like platoons and it allows for a message rate of 10Hz for CAMs when the channel load is $< 70\%$ [15]. When the channel load is $> 70\%$ (under heavy vehicular density), parameters such as message rate, data rate and transmit power can be adapted to prevent channel congestion and packet drops. The message rate can be as low as 1Hz and is controlled by a Decentralized Congestion Control (DCC) algorithm where one or more parameters are adapted. Those parameters are tuned according to the measured channel load at run time to avoid channel congestion and keep the channel load $< 70\%$.

Generally, platoon control is achieved in each vehicle by running software tasks on the electronic control units (ECUs) and it is split into an upper-layer and a lower-layer running over two different communication standards and rates, one for inter-vehicle communication and one for intra-vehicle communication. The usage of different communication standards in each layer with a different operating frequency introduces the multi-rate control concept. In that spirit, a multi-layer multi-rate control scheme is adopted in this paper since the separation between layers reduces complexity in design and analysis. This paper builds upon previous work [25] where the focus is on high-performance platoon control under low network congestion. In this paper, the performance under high network congestion is studied, thus, considering communication delay in the control design.

Communication delay happens in Networked Control Systems (NCSs) since the message has to be processed by the sender and the receiver, and also because of the limited bandwidth of the network and because multiple nodes share the same channel causing interference. This delay (either constant or time-varying) might degrade the performance if not considered in the control design and might destabilize the system especially in case of long delay or interval of no-received-packets. The value of the delay depends on the vehicular density i.e., the delay increases if too many vehicles share information via the communication channel. Interestingly, communication delay is not generally considered in platoon control literature. This is mainly supported by the assumption that the delay is quite low in a non-congested network [25]. As shown in [38], string stability (attenuation of disturbances throughout the platoon) is seriously compromised by a high communication delay. In this paper, a novel augmented state-space, multi-layer multi-rate, platoon model is introduced that considers the delay in the received signal

under predecessor-follower (PF) topology.¹ The eigenvalues of such system depends on different parameters (e.g., the upper-layer, and the lower-layer sampling periods, the headway time, etc.) and thus the stability (the convergence of each of the vehicles to the reference acceleration) depends on the chosen parameters. The stability of such system is analyzed using Lyapunov theory in order to check the stability region under different sets of parameters. Next, the proposed delay-aware platoon model is used in the design of a switching platoon control. Communication delay is stochastic or time varying. In this paper, we assume that the delay is mostly lower than one sampling period. For the cases where the delay is more than one sampling period, it is assumed that the packet is lost.

In each vehicle, the upper-layer receives CAM messages from its preceding vehicle with a certain delay. The upper-layer computes the desired acceleration which is then attained by the lower-layer. The role of the upper-layer is to compute the desired acceleration while ensuring safety, fuel efficiency, driving comfort, tracking capability and string stability. The desired acceleration is computed based on the vehicle status and the status of its preceding vehicle (assuming a PF topology). The upper-layer controller is designed using Model Predictive Control (MPC) [43] because of its ability to handle different constraints on input and states. For MPC to compensate for the delay, an augmented prediction model is formulated based on the new augmented state-space platoon model. Then, the desired acceleration is passed over to the high-speed in-vehicle network, e.g., FlexRay, to be realized by the lower-layer controller which is responsible for reaching the desired acceleration within a certain time.

We implement a DCC algorithm by tuning the message rate (thus called message-rate congestion control) based on the vehicular density to keep the channel load $\leq 70\%$. We create traffic scenarios on a highway where the number of vehicles changes. We estimate the channel load in each scenario and therefore compute the message transmission rate that keeps the channel load close to 70%. When the DCC algorithm is active, the V2V message rate is going to change depending on vehicular densities. Therefore, the upper-layer sampling rate should be adapted according to the V2V message rate. However, designing a platoon control which can run for multiple sampling rates is a challenging task. We tackle this challenge considering slow-changing traffic behaviour resulting in a switched system which switches slowly between different controllers where each controller is pre-designed and tuned for each specific message rate.

We use our co-simulation framework CReTS (Control, Network and Traffic Simulator) [26] to evaluate our platoon control framework under realistic network behavior. CReTS is a framework composed of network simulator ns-3, traffic simulator SUMO and Matlab. ns-3 provides packet reception ratios and delays to the controller implemented in

¹In PF topology, a follower vehicle receives information from its direct predecessor only.

Matlab. Different network congestion levels can be simulated by adjusting the number of communicating vehicles using SUMO and the corresponding control performance is evaluated. Using the proposed method in this paper, we show the possibility of maintaining the platoon performance in terms of string stability and fuel economy under high network congestion by adapting the V2V message rate. This paper extends our results presented in [25] on three main aspects:

- 1) How to include the V2V communication delay in the platoon model.
- 2) How to design a delay-aware MPC controller to improve performance.
- 3) The evaluation of the delay-aware MPC under a proposed DCC system.

This paper is organized as follows. In Section II, we present the related work. Section III introduces the overall architecture of the system with different layers and rates, and their interaction/relation. In Section IV, we give an overview of the IEEE 802.11p communication standard, DCC and message-rate congestion control. Section V introduces the vehicle and platoon models under communication delay and the lower-layer control design using state-feedback control. In Section VI, we analyse the stability of the platoon model using Lyapunov theory. In Section VII, we present the MPC control design based on the augmented state-space platoon model. The evaluation of our approach is presented in Section VIII where we show the control and network performance and fuel consumption. Section IX concludes our paper.

II. RELATED WORK

In [25], we propose a multi-layer control scheme where vehicles follow the recommended V2V message rate of 10Hz (100ms) under light traffic and network congestion levels. In this paper, we extend our earlier work where communication delay and different V2V message rates are taken into account under high traffic and network congestion levels. In the following, we describe the related work in the relevant directions.

A. NETWORKED CONTROL SYSTEM (NCS) WITH DELAY

In a NCS, the information (a control input or a reference signal for example) is exchanged through a real network among the sensors, actuators and the controllers. The network-induced delay may lead to instability and degrades the performance of the control system. In [4], the influence of the sampling period and network delay on the stability of the system is analyzed using a stability region plot. The stability was checked by increasing the delay slowly and testing the eigenvalues of the closed-loop system matrix under a fixed value of the linear control gain. In [8], the stability of the NCS under delays that are less than and larger than the sampling period is analyzed where LMI conditions are derived using Lyapunov theory to ensure stability and also a stabilizing control gain is found. In our work, in order to find the

stability region of the multi-layer multi-rate platoon system under different sets of parameters, we extended the approach presented in [8] by finding LMI conditions using Lyapunov theory. We solved those LMIs under different parameters: lower-layer sampling periods, upper-layer sampling periods, headway times, pole locations of the lower-layer controller, delay values. We are only interested in feasibility of achieving stability and not in finding the stabilizing control gain, since the control gain is designed using MPC in this work.

B. PLATOON CONTROL IN A CONGESTED NETWORK

Some work studied the effect of communication imperfections (delay and packet loss) on the string stability of the platoon of vehicles. In [38], the authors investigate the design of a robust controller against packet loss and communication delay and they show that string stability is seriously compromised by the communication delay. They also show that string stability can be maintained if the delay in the received preceding-vehicle information is small and that, if the delay is a random variable and not a constant, stability cannot be guaranteed. Constant communication delay is considered in [20], [38], [50] whereas time-varying communication delay is considered in [10], [19], [52]. The authors in [38] studied the effects of communication delay on string stability. A control law is designed that uses the information received from the preceding vehicle and the lead vehicle. In [20], an H_∞ control method is proposed for heterogeneous platoons with uniform communication delay and uncertain dynamics. In [52], the authors investigated the effects of stochastic communication delay on the stability of connected cruise control. In [10], the authors considered the platooning problem as a problem of achieving consensus in a network of dynamical systems under time-varying heterogeneous communication delay. In this work, we consider a constant communication delay. We derive the multi-layer multi-rate platoon model considering communication delay where the delayed acceleration is augmented to the state-space platoon model. Using our CReTS simulation framework, we measure the communication delay experienced between vehicles for low, medium, and high congestion scenarios. This time-varying communication delay is upper-bounded by the measured maximum delay under different congestion levels. The delay is small in scenarios with low or medium congestion. Control performance and string stability do not suffer in such cases. On the other hand, string stability and the tracking performance of the platoon system degrade with high congestion. Message-rate control is then used to reduce congestion.

C. DELAY-AWARE MPC

In our earlier work [25], we extended the state-of-the-art multi-objective MPC approach for platooning in a distributed manner by considering a realistic vehicle dynamics (a more descriptive platoon model), the multi-rate concept in the design of the upper-layer and the lower-layer controllers, and realistic network behavior with packet loss. Multi-objective MPC is implemented in order to satisfy different objectives

such as minimal fuel consumption, tracking capability and desired driver response. In our current paper, we redesigned the MPC controller to compensate for the communication delay where the prediction model for MPC is constructed based on the new augmented state-space model. String stability is satisfied through constraints satisfaction whereas better fuel consumption and good tracking capability are obtained through cost function minimization. In [31], string stability is accomplished by matching MPC through tuning its weighting matrices with a linear controller such that the behavior of MPC matches the string-stable controller (in the unconstrained case). When the constraints are active, MPC is used to satisfy the constraints. This does not match the linear controller anymore and string stability is not guaranteed. Model augmentation is considered where the acceleration of the preceding vehicle (considered as a measured disturbance) is augmented to the system state. However, the delay in the received acceleration was not taken into account and only the actuator delay was considered. In [36], an MPC algorithm is introduced for a NCS to handle packet loss and delay using an augmented state-space model. However, the analysis is done for an unconstrained MPC. In [39], MPC was designed to compensate for input and state delay using a state-space augmented system. The input delay appears due to the time needed to compute the control input and the state delay appears due to the in-vehicle induced network delay (CAN bus). The delay due to V2V communication is not considered in the paper. In [40], a distributed MPC is developed for vehicle platooning while guaranteeing stability and recursive feasibility. The communication delay can be an integer multiple of the sampling period and the desired acceleration is computed based on the old (outdated) received acceleration. For a system with coupled dynamics, as in our approach, computing a cost function and terminal constraint set every time step, as proposed in [40], requires huge computational resources. As we aim at low computational costs, we followed the proposed theorem in [37] which states that stability of MPC can be obtained when considering a zero terminal constraint set.

D. CONTROL IN COMBINATION WITH DCC

A joint communication and control design approach is considered in [23] where the parameters of the CACC controller are adapted based on the measured network parameters. A lookup table is created for each packet error rate where the parameters of the CACC controller are tuned such that collision is avoided between vehicles. However, this simple control scheme cannot guarantee safety and string stability under highly dynamic traffic situations. No earlier work has been found in literature where DCC algorithms are explicitly considered in the platoon control.

III. PLATOON CONTROL ARCHITECTURE

Fig. 1 shows the overall platoon architecture that is considered in this paper. The role of the upper-layer is to compute the desired acceleration based on the current vehicle

information and the information of the preceding vehicle received via V2V wireless communication. The leading platoon vehicle (platoon leader) is manually controlled by a human driver where the driver follows a pre-defined acceleration profile. Thus, the leading platoon vehicle has only the lower-layer controller which is responsible for attaining the desired acceleration. The actual acceleration of each vehicle is transmitted over the V2V wireless communication to other vehicles and received with some delay based on the network congestion level. The upper-layer in each following vehicle computes the desired acceleration based on two versions of the received acceleration (i) the current received acceleration at time step k , $a^{i-1}(k)$, that is received with delay (at most) τ and (ii) the old “outdated” acceleration of the preceding vehicle $a^{i-1}(k-1)$ that is received at time step $k-1$. The computed desired acceleration (a_{des}^i) of the i^{th} vehicle is then passed to the lower-layer controller to be attained within certain time. The upper-layer controller runs at 10Hz when the channel load is $\leq 70\%$. When the channel load is $> 70\%$ (under high network congestion levels), the message rate, and thus the upper-layer sampling period, is modified to have a value between 1–10Hz in order to keep the channel load at 70%. This is called message-rate congestion control.

The lower-layer controller receives the desired acceleration over fast and reliable in-vehicle networks such as FlexRay or Ethernet [44]. The lower-layer controller can be implemented with a short sampling period such as 2ms, 10ms, or 20ms considering typical automotive architectures [22].

It should be noted that time synchronization is required for the platoon to generate a shared notion of time in all vehicles so they can operate properly. Time synchronization can be achieved using GPS.

Fig. 2 shows the road segment (from SUMO) considered in our experiments. It is a road section of 3km with four lanes in each direction. Platooning is most effective on (busy) highways, which is why we focus on such highway scenarios. In our setup, we consider a single platoon that uses a dedicated lane, since we only consider longitudinal control, i.e., regulating the longitudinal motion of vehicles via controlling their speed, acceleration and the distance gap between vehicles on the same lane. Lateral control, where platoon vehicles are kept in the desired lane through steering or follow a reference trajectory in case of lane changing, is beyond the scope of this paper. The non-platoon vehicles in our experiments can change their lanes, which means that the lane for platoon vehicles, “Lane 1” in Fig. 2, is actually shared between platoon and non-platoon vehicles. In such a case, the platoon leader should adapt its acceleration or velocity (and therefore the acceleration and velocity of the followers) to avoid collisions with preceding non-platoon vehicles.

IV. MESSAGE-RATE CONGESTION CONTROL

A. IEEE 802.11p V2V COMMUNICATION STANDARD

IEEE 802.11p based connected vehicle technology is known in the US as DSRC (Dedicated Short-Range Communication)

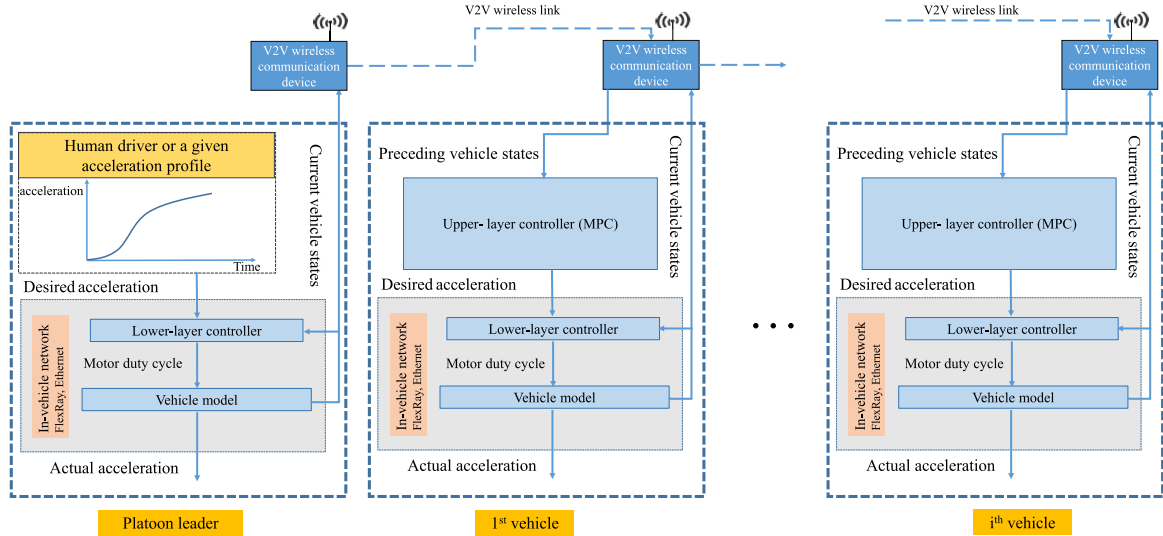


FIGURE 1. Multi-layer multi-rate platoon control architecture.

and in Europe as ITS-G5 (Intelligent Transportation System-5GHz). IEEE 802.11p is an amendment to IEEE 802.11a where the physical layer properties are modified to cope with the rapidly changing vehicles position. Moreover, IEEE 802.11p provides more robustness against fading and increases the tolerance for multipath propagation effects of signals in a vehicular environment [35]. In ITS-G5, a 50MHz spectrum in the 5.9GHz range is licensed to Cooperative Intelligent Transport Systems (C-ITS) to be used exclusively for V2V and V2I (Vehicle-to-Infrastructure) communications. This spectrum is divided into six service channels (SCHs) and one control channel (CCH) each of 10MHz bandwidth. CCH is dedicated to safety-critical data, cooperative road safety and critical applications such as platoons. The six service channels (SCHs) are dedicated to safety and non-safety ITS applications [29].

As defined by the European Telecommunications Standards Institute (ETSI) [12], [13], V2V wireless communication for platoons under the IEEE 802.11p standard [29] uses the CCH to share information between vehicles. Two types of messages share the control channel, periodic Cooperative Awareness Message (CAM) and event-triggered warnings Decentralized Environmental Notification Message (DENM) [12]. DENMs are used to issue warnings in emergency situations [14]. CAMs are sent periodically to create and maintain awareness between vehicles and roadside units and therefore used for platoon applications. Data types contained in all generated CAMs by a vehicle shall include all fast-changing (dynamic) measured status information such as heading, speed, position and acceleration.

Under high network congestion, the probability of packet collisions increases with the limited channel capacity of 10MHz. Moreover, there is no re-transmission mechanism for lost packets in IEEE 802.11p. The medium access control (MAC) algorithm deployed by IEEE 802.11p is based on the CSMA/CA algorithm (carrier sense multiple access with collision avoidance). Vehicles first listen to the channel

before transmission and if the channel is perceived as idle for a predetermined listening period the vehicle can start to transmit directly. If two vehicles decide to send at the same time, a collision happens [18]. The messages are broadcast periodically at a certain message rate to each other, without any acknowledge nor re-transmission mechanisms to guarantee reception. The sequence number of each message allows vehicles to estimate the channel and the information reception rate, and it provides awareness of traffic around the vehicle. A DCC algorithm is engaged to keep the channel load under a certain threshold.

B. DECENTRALIZED CONGESTION CONTROL (DCC)

CAMs follow a default message transmission rate of 10Hz (sampling period = 100ms) when the channel load is $\leq 70\%$. Under high vehicular density (i.e., channel load is $> 70\%$), this default value (10Hz) causes channel congestion, packet drops and long delay. Therefore, reliability of safety-critical applications, e.g., platoons, might be compromised. To solve this issue, under a channel load greater than 70%, message rate of CAMs can be lowered to as low as 1Hz and controlled by a Decentralized Congestion Control (DCC) algorithm [15]. In literature, there are different DCC algorithms (e.g., Limeric) [3], [32], [57] that control the channel load to avoid channel congestion. Those DCC algorithms tune either one parameter, such as message rate, data rate, transmit power, carrier sensing threshold or more communication parameters such as both message rate and transmit power. It should be noted that congestion control can be implemented in a centralized or a decentralized way. In this paper, we assume that all the vehicles in a platoon have the same V2V message rate, as elaborated in Section IV-E.

C. CHANNEL MODELING

Modeling the V2V communication channel is important when analysing DCC-aware platoon control. Channel modeling is challenging because of the different aspects that play

a role in wireless communication [33], [61]. For example, the type of environment (urban or highway) and vehicles (cars, vans or trucks), the mobility pattern, and the antenna design and positioning all have impact on the quality of communication. In urban areas, interference with other wireless communication and signal blocking, shadowing, reflection, and diffraction because of buildings are the main challenges that need to be considered. Mobility of vehicles should be considered in the channel model for urban contexts or highways since they may cause reflection or block the line-of-sight (LOS). There is no one single model that can consider all these characteristics together. Instead, there are different models in literature that can be used to describe a realistic channel in different environments.

The strength of the received signal power is affected by small-scale fading (diffraction, reflection, scattering) and large-scale fading related to distance (shadowing, path loss). It is a common practice in literature to represent these aspects in separate models. In this work, the large-scale propagation (path loss) between vehicles is modeled using a dual-slope log-distance model that is commonly used in literature [61]. Small-scale fading is modeled using the Nakagami statistical channel model. The channel model used in our scenario, a Nakagami- \mathcal{M} model where \mathcal{M} varies based on the distance between the sending and the receiving vehicles, is a widely accepted channel model [46]. It was shown in [9] to be a suitable probabilistic channel model for vehicular communication. Since we consider a single platoon with a PF topology in a highway scenario in which all control-related communication is between contiguous vehicles in the platoon, over at most several tens of meters distance, a more detailed channel model differentiating LOS from obstructed LOS or including shadowing is not needed, as illustrated for instance through the comparison of channel models made in [2].

D. CHANNEL LOAD ESTIMATION

The IEEE 802.11 standard has defined channel sensing in its Clear Channel Access (CCA) protocol, which is used to measure channel load at run time [27], [28]. One way to estimate average channel load is to calculate the total air time of messages sent by all vehicles in one access range. The number of vehicles that causes 70% channel load can be estimated as follows. The 100% channel load can be calculated assuming that one vehicle can start sending a message immediately after the transmission of the previous message is completed, which is an ideal case excluding MAC protocol delays. One 300Byte message takes around 0.4ms in a 6Mbps channel, which is used in the IEEE 802.11p V2X (vehicle-to-everything) communication. If we consider a 300Byte message length of 0.4ms, the total number of messages per second is $1000/0.4 = 2500$. At 10Hz message rate per vehicle, this allows for 250 vehicles to transmit without conflict at 100% load. The 250 vehicles is calculated based on a 10Hz message rate of 300Byte messages in a 6Mbps channel. At 70% load, the number of vehicles is $250 \times 0.7 = 175$.

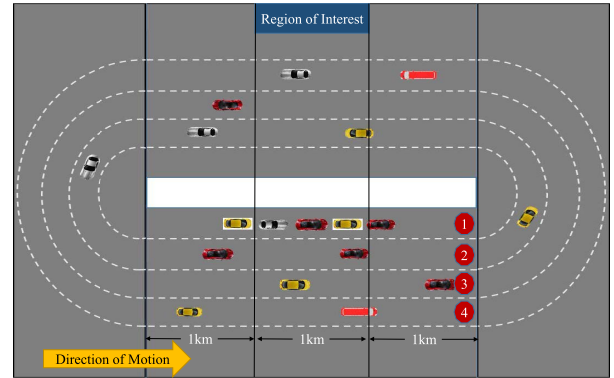


FIGURE 2. Road segment considered in our experiments, as visualized in SUMO.

We can take 1km access zone to estimate the vehicle-vehicle distance, which means a vehicle can listen to others within 500 meter distance. If we consider an 8-lane highway and 1km communication zone, the 250 vehicles can be distributed over 8 lanes within 500 meter around the listening vehicle to derive the average inter-vehicle distance: $1000/(250/8) = 32\text{m}$. 32m vehicle-to-vehicle distance (excluding the length of the vehicle) is a quite normal situation in busy times. With such an estimation, 70% channel load means 175 vehicles using 10Hz message rate. The average vehicle-to-vehicle distance on an 8-lane highway (using 1km communication zone) is $1000/(175/8) = 45.6\text{m}$. In other words, if the vehicle density is as high as one per 45 meter on an 8-lane highway, the channel is 70% loaded.

Note that the above is an ideal estimation without considering message collision due to the MAC protocol, which means the measured channel load by a vehicle is usually lower due to collisions and the received signal power loss over the channel from the sender.

E. MESSAGE-RATE CONGESTION CONTROL ALGORITHM

As mentioned earlier, a DCC algorithm can be implemented by tuning one or more parameters such as message rate, data rate, etc. In this section, we consider the DCC by computing the message rate that keeps the channel load at certain threshold while keeping the other parameters constants. Thus it is called message-rate congestion control.

The average message transmission rate under different vehicular density, assuming the default data rate of 6Mbps for all scenarios, can be computed using the following formula:

$$m_r = \frac{\text{channel load (in \%)} / 100}{m_t \times \text{number of vehicles (in 1km)}} \quad (1)$$

where m_r , m_t denote message transmission rate and message transmission time, respectively. Note that Eq. 1 is similar to the estimation done in Section IV-D (by letting channel load = 100%, $m_t = 0.4\text{ms}$ and number of vehicles in 1km = 250, we obtain $m_r = 10\text{Hz}$). In order to find the message rate that keeps the channel load fixed at 70% under certain vehicular density, we find the number of vehicles in 1km and set channel load = 70% and $m_t = 0.4\text{ms}$ in Eq. 1.

Recall that Fig. 2 shows (the SUMO view of) the experimental setup we consider in this paper. For different vehicular densities, we compute the corresponding message rates at a certain threshold using the formula in Eq. 1. This is shown in Table 1, which summarizes the traffic and network scenarios under consideration. As shown in Table 1, we consider certain scenarios where DCC is active and scenarios where DCC is not active. DCC is disengaged for the low level of congestion (channel load $\leq 70\%$) i.e., for the traffic scenarios where the number of vehicles in the road segment shown in Fig. 2 is ≤ 525 and therefore the standard V2V message rate of 10Hz is considered. For high congestion levels, DCC is engaged and the message rate is computed using Eq. 1. For example, for 1000 vehicles in this 3km road segment, the average message transmission rate that keeps channel load at 70% can be computed as follows:

$$m_r = \frac{0.7}{(0.4 \times 10^{-3}) \times (1000/3)} = 5.25\text{Hz}.$$

For the sake of comparison, we disengage DCC for some traffic scenarios under high traffic congestion levels. We use the standard 10Hz V2V message rate for 600, 800, 900, and 1000 vehicles in the 3km road segment. By doing this, we cause channel load to reach 80%, 106%, 120%, and 133%, respectively. In these cases, we can see the importance of using DCC instead of the standard V2V message rate in highly congested network and traffic scenarios.

The message rate in a vehicle is computed based on the measured channel load and the number of vehicles in a moving window of a given road length (vehicle density) as per Eq. 1. Stability issues might arise if vehicles have different V2V message rates. The stability of the distributed platoon control under heterogeneous message rates is an open question that is not addressed in the literature and needs further research. State-of-the-art methods for channel estimation often rely on traffic information from I2V (Infrastructure-to-Vehicle) [7], [53]. Both through using Eq. 1 and when relying on I2V traffic information, nearby vehicles are expected to have similar channel-load estimates as they experience similar channel characteristics. Moreover, the platoon management system has a coordinator vehicle [1] that communicates and sends commands to the other vehicles in the platoon. It is possible for the coordinator vehicle to send a message-rate command to the rest of the platoon. Our work assumes that the message rate is shared among platoon members (through the earlier mentioned CAMs), where the platoon leader decides and sets the rate to the lowest received value. This common message rate is then shared and set as the upper-layer sampling rate by every controller in each platoon vehicle.

Our method can also be applied in the case where each vehicle has a different message rate since we designed a distributed MPC, where each vehicle has its own controller that can run at different rates. This, in fact, also allows to consider heterogeneous vehicles (non-identical vehicles). The question then is whether the platoon can maintain a short inter-vehicle distance and maintain string stability. This is an

open question not addressed in the literature. This requires further investigation and is an interesting direction for further research.

V. PLATOON MODELING UNDER COMMUNICATION DELAY

In this section, we derive the state-space augmented platoon model that considers the communication delay in the received signal for the multi-rate multi-layer system. Sections V-A and V-B introduce the vehicle model and lower-layer controller. The platoon model is then introduced for continuous and discrete systems in Section V-C. The platoon model under communication delay is presented in Section V-D.

A. VEHICLE MODEL

The model of vehicle i combines the longitudinal vehicle dynamics and the powertrain system (throttle, engine, driveline). The powertrain system can be simplified and modeled as a DC-servo motor (throttle actuator) as proposed in [56], [58]. The throttle actuator adjusts throttle plate angle. Then it is possible to regulate vehicle speed through throttle plate angle adjustments. The combined model is given by:

$$\dot{x}_v^i = A_v^i x_v^i + B_v^i u_v^i \quad (2)$$

where A_v^i and B_v^i are the state and input matrices respectively, u_v^i is the duty cycle of the input to the motor and $x_v^i = [a^i \ \dot{a}^i]^T$ is the state vector, where a^i and \dot{a}^i are the acceleration and the rate of change of the acceleration of vehicle i , respectively. Moreover, state matrix A_v^i and input vector B_v^i of vehicle i are defined as:

$$A_v^i = \begin{pmatrix} 0 & 1 \\ -1 & -(\tau^i + \tau_a^i) \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \quad B_v^i = \begin{pmatrix} 0 \\ \frac{K^i K_a^i}{\tau^i \tau_a^i} \end{pmatrix} \in \mathbb{R}^{2 \times 1},$$

where τ^i , τ_a^i , K^i , K_a^i are model parameters of vehicle i . \mathbb{R} is the set of real numbers.

B. LOWER-LAYER CONTROLLER

The lower-layer controller deals with vehicle model Eq. 2 to achieve the desired acceleration. Since the controller will be implemented on an ECU (electronic control unit), we first need to discretize the vehicle model. Discretizing Eq. 2 with the lower-layer sampling period h_l , we obtain,

$$x_v^i(k+1) = \Phi_v^i x_v^i(k) + \Gamma_v^i u_v^i(k), \quad (3)$$

where $x_v^i(k) = [a^i(k) \ \delta a^i(k)]^T$ and $a^i(k)$, $\delta a^i(k)$ are the discretized acceleration and rate of change of acceleration at time step k of the vehicle i . Φ_v^i , Γ_v^i represent the discretized system matrices with period h_l . The choice of h_l is driven by the sampling periods supported by the common automotive operating systems such as OSEK [49]. For example, OSEK supports periods of 2, 5, 10, ...ms. A shorter h_l requires a higher resource usage (in terms of communication and computation) of the in-vehicle electrical and electronic architecture.

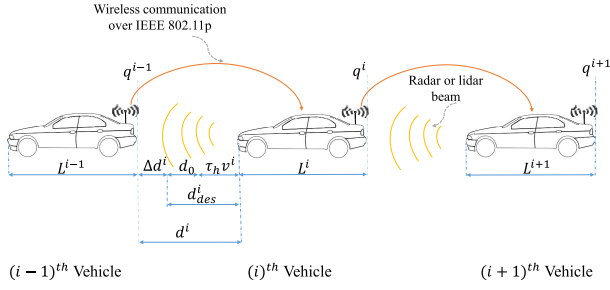


FIGURE 3. Consecutive vehicles in a platoon.

In order to compute the control input $u_v^i(k)$, we consider a state feedback control law in the lower-layer controller of the following form,

$$u_v^i(k) = \kappa^i x_v^i(k) + F^i a_{des}^i, \quad (4)$$

where a_{des}^i , κ^i , F^i are the desired acceleration, feedback gain and feedforward gain of vehicle i , respectively. Feedback gain κ^i is designed using pole placement technique which places the poles of the discrete-time system inside the unit circle to guarantee stability. See [25] for more details. By substituting Eq. 4 into Eq. 3 we obtain the closed-loop system:

$$x_v^i(k+1) = \Phi_v^i x_v^i(k) + \Gamma_v^i (\kappa^i x_v^i(k) + F^i a_{des}^i), \quad (5)$$

gains κ^i , F^i have to be designed such that the discrete-time closed-loop system is stable and a^i converges to a_{des}^i .

C. PLATOON MODEL

In Predecessor-Follower (PF) topology, the platoon model is distributed; each vehicle has a model of itself (vehicle model) and its relation with its predecessor (inter-vehicle dynamics).

1) CONTINUOUS-TIME PLATOON MODEL

To obtain the platoon model under PF topology, the inter-vehicle dynamics that relate vehicle i to vehicle $i-1$ is introduced. This is achieved by defining two new states, Δv^i and Δd^i , which represent the speed difference and the gap error between vehicles, respectively (see Fig. 3). They are defined as follows,

$$\Delta d^i = d^i - d_{des}^i, \quad \Delta v^i = v^i - v^{i-1},$$

where Δd^i is the error between the actual gap d^i and the desired inter-vehicle gap d_{des}^i between vehicle i and vehicle $i-1$. Δv^i is the velocity difference between vehicle i and vehicle $i-1$, where v^i denotes the velocity of vehicle i . d^i and d_{des}^i are defined as,

$$d_{des}^i = d_0 + \tau_h v^i, \quad d^i = q^{i-1} - q^i - L^i,$$

where d_0 is the gap between vehicles at standstill, τ_h is the constant headway time (the time vehicle i needs to reach the position of vehicle $i-1$ when $d_0 = 0$). L^i and q^i are the length and position of vehicle i , respectively.

Combining the vehicle model with the inter-vehicle dynamics we obtain the following continuous-time

TABLE 1. Traffic and network scenarios under consideration in the road segment shown in Fig. 2.

	number of vehicles in 3km	Message rate (Hz)	Upper-layer sampling period (ms)	Channel load (%)
No DCC	5	10	100	< 70
	300	10	100	< 70
	525	10	100	70
DCC	600	8.75	114	70
	700	7.5	133	70
	800	6.56	152	70
	900	5.833	171	70
	1000	5.25	191	70
	1300	4.0385	248	70
	1500	3.5	286	70
No DCC	600	10	100	80
	800	10	100	106
	900	10	100	120
	1000	10	100	133

platoon model:

$$\dot{x}_p^i = A_p^i x_p^i + B_p^i \underbrace{\begin{bmatrix} u_v^i \\ a^{i-1} \end{bmatrix}}_{\in \mathbb{R}^{2 \times 1}}, \quad (6)$$

where $x_p^i = [a^i \quad \dot{a}^i \quad \Delta d^i \quad \Delta v^i]^T$ is the state vector, a^{i-1} is the acceleration of the preceding vehicle. The state and input matrices are defined as:

$$A_p^i = \begin{bmatrix} A_v^i & \mathbf{0} \\ H_m^i & A_m^i \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \quad B_p^i = \begin{bmatrix} B_v^i & \mathbf{0} \\ \mathbf{0} & G_m^i \end{bmatrix} \in \mathbb{R}^{4 \times 2}.$$

where

$$A_m^i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad H_m^i = \begin{bmatrix} -\tau_h & 0 \\ -1 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2},$$

$$G_m^i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{2 \times 1}.$$

2) DISCRETE-TIME PLATOON MODEL

The platoon model Eq. 6 has to be discretized with the upper-layer sampling period h_u to be implemented on a digital device (ECU). h_u is computed based on the message rate calculations in Section IV-E (see Eq. 1). Therefore, h_u can be computed as,

$$h_u = \frac{1}{m_r}, \quad (7)$$

where h_u has to be approximated to the nearest integer.

Overall system behavior depends on the relation between the lower-layer sampling period h_l and the upper-layer sampling period h_u . Generally, $h_u \gg h_l$ and we choose h_u as an integer multiple of h_l , i.e.,

$$\eta = \frac{h_u}{h_l}, \quad \eta \in \mathbb{N}. \quad (8)$$

If we choose $h_l = 2\text{ms}$, then η has to be approximated to the nearest integer in the cases of $h_u = 133\text{ms}$ (700 vehicles), 171ms (900 vehicles), or 191ms (1000 vehicles) shown in Table 1 since $\eta \in \mathbb{N}$ (\mathbb{N} is the set of natural numbers). After the approximation of η in those particular cases, h_u that is actually implemented are 172ms, 192ms, or 134ms.

Using Eq. 1, those upper-layer sampling periods correspond to channel load of approximately 69.5% which is still close to 70% channel load.

We first discretize Eq. 6 at sampling period h_l and substitute $u_v^i(k)$ as per Eq. 4:

$$x_p^i(k+1) = \Phi_p^i x_p^i(k) + \Gamma_p^i \begin{bmatrix} \kappa^i x_p^i(k) + F^i a_{des}^i(k) \\ a^{i-1}(k) \end{bmatrix}, \quad (9)$$

where Φ_p^i and Γ_p^i are the discretized A_p^i and B_p^i , respectively. Φ_p^i and Γ_p^i are defined as follows,

$$\Phi_p^i = e^{A_p^i h_l} \in \mathbb{R}^{4 \times 4}, \quad \Gamma_p^i = \int_0^{h_l} e^{A_p^i s} B_p^i ds \in \mathbb{R}^{4 \times 2},$$

Simplifying the previous equation yields the following:

$$x_p^i(k+1) = \Pi_p^i x_p^i(k) + \Xi_p^i F^i u^i(k) + \Psi_p^i a^{i-1}(k), \quad (10)$$

where

$$u^i(k) = a_{des}^i(k).$$

In Eq. 10, the platoon model is discretized with the lower-layer sampling period h_l and it does not yet capture the upper-layer dynamics which runs with the sampling period h_u . Since $h_u = \eta h_l$ (Eq. 8), the lower-layer control loop should execute η times within one upper-layer sampling period h_u . We unroll the loop (i.e., Eq. 10) η times to obtain the upper-layer dynamics. In other words, $x_p^i(k+\eta)$ is found by recursively solving $x_p^i(k+j)$ for $j = 1, \dots, \eta$. For example, $x_p^i(k+2)$ is found as,

$$x_p^i(k+2) = \Pi_p^i x_p^i(k+1) + \Xi_p^i F^i u^i(k+1) + \Psi_p^i a^{i-1}(k+1).$$

By substituting the definition of $x_p^i(k+1)$ as per Eq. 10 in the above equation, we obtain the following:

$$x_p^i(k+2) = (\Pi_p^i)^2 x_p^i(k) + (\Pi_p^i + \mathcal{I}) \Xi_p^i F^i u^i(k) + (\Pi_p^i + \mathcal{I}) \Psi_p^i a^{i-1}(k).$$

It is noted that in the previous formula we set $u^i(k+1) = u^i(k)$ and $a^{i-1}(k+1) = a^{i-1}(k)$. This is because the new acceleration of the preceding vehicle $a^{i-1}(k+1)$ is received only at time $t = (k+\eta)h_l$ and then the new control input $u^i(k+1)$ is computed (once every ηh_l). Therefore, at step $k+j$ or at time $t = (k+j)h_l$, $j = 1, \dots, \eta-1$, $u^i(k+j) = u^i(k)$ and $a^{i-1}(k+j) = a^{i-1}(k)$. See Fig. 4a for more details.

Then $x_p^i(k+\eta)$ can be represented in terms of $x_p^i(k)$, $a^{i-1}(k)$, $u^i(k)$ as follows,

$$\begin{aligned} x_p^i(k+\eta) &= (\Pi_p^i)^\eta x_p^i(k) + \left((\Pi_p^i)^{\eta-1} + (\Pi_p^i)^{\eta-2} \right. \\ &\quad \left. + \dots + \Pi_p^i + \mathcal{I} \right) \Xi_p^i F^i u^i(k) \\ &\quad + \left((\Pi_p^i)^{\eta-1} + (\Pi_p^i)^{\eta-2} \right. \\ &\quad \left. + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i a^{i-1}(k). \end{aligned}$$

Therefore the platoon model for vehicle i under PF topology is represented as:

$$x^i(k+1) = \alpha^i x^i(k) + \beta^i u^i(k) + \gamma^i a^{i-1}(k), \quad (11)$$

where,

$$x^i(k+1) := x_p^i(k+\eta), \quad (12)$$

$$x^i(k) := x_p^i(k) = \begin{bmatrix} a^i(k) \\ \delta a^i(k) \\ \Delta d^i(k) \\ \Delta v^i(k) \end{bmatrix} \in \mathbb{R}^{4 \times 1}, \quad (13)$$

$$\alpha^i := (\Pi_p^i)^\eta,$$

$$\beta^i := ((\Pi_p^i)^{\eta-1} + (\Pi_p^i)^{\eta-2} + \dots + \Pi_p^i + \mathcal{I}) \Xi_p^i F^i,$$

$$\gamma^i := ((\Pi_p^i)^{\eta-1} + (\Pi_p^i)^{\eta-2} + \dots + \Pi_p^i + \mathcal{I}) \Psi_p^i. \quad (14)$$

$\mathcal{I} \in \mathbb{R}^{4 \times 4}$ is the identity matrix where the diagonal elements are ones and the off-diagonal elements are all zeros. $\alpha^i \in \mathbb{R}^{4 \times 4}$, β^i and $\gamma^i \in \mathbb{R}^{4 \times 1}$. Fig. 4a explains the relation between x^i and x_p^i that is defined in Eq. 12 and Eq. 13. Initially at step k ($k \in \mathbb{N}$), $x^i(k) = x_p^i(k)$. When time evolves for one step i.e., at $k+1$, $x^i(k+1)$ is the new state of the platoon vehicle obtained at time $t = (k+1)h_u$. Whereas $x_p^i(k+1)$ is the state of the platoon vehicle obtained at time $t = (k+1)h_l$ i.e., after 2ms. Therefore, by unrolling the loop in Eq. 10 η times then we obtain $x^i(k+1) = x_p^i(k+\eta)$.

D. PLATOON MODEL CONSIDERING COMMUNICATION DELAY

In the platoon model of Eq. 11, the acceleration of the preceding vehicle a^{i-1} is assumed to be instantly received by vehicle i without any delay. For example, to compute the updated system state $x^i(k+2)$ at time $t = (k+2)h_u$, the following states should be available at time $t = (k+1)h_u$: the measured state $x^i(k+1)$, the computed input $u^i(k+1)$, and the acceleration of the preceding vehicle $a^{i-1}(k+1)$. In reality, due to communication imperfections, at time $t = (k+1)h_u$ the acceleration of the preceding vehicle $a^{i-1}(k+1)$ is received by the following vehicle with a delay τ (ms) where $\tau \leq h_u$. Such delay must be considered in the platoon model to be taken into account in the controller design. Note that the case when $\tau > h_u$ is considered as a packet loss. The MPC platoon controller (elaborated in Section VII) then essentially uses the next optimal value from the optimized control input sequence. Since the previously received acceleration $a^{i-1}(k)$ is available for time duration τ before the reception of the new acceleration $a^{i-1}(k+1)$, both acceleration values $\{a^{i-1}(k), a^{i-1}(k+1)\}$ should be used in computing the new system state $x^i(k+2)$. Fig. 4b shows an example of the signal $a^{i-1}(k+1)$ that is received with the delay τ i.e., after m lower-layer samples where,

$$m = \left\lceil \frac{\tau}{h_l} \right\rceil. \quad (15)$$

Following the same procedure done in the previous section, the platoon model can be derived such that it takes into account the old acceleration and the new delayed acceleration of preceding vehicle. For the first m samples, the platoon model considers the old acceleration of the preceding vehicle. After the new acceleration is received with delay τ (after m samples), the platoon model considers the newly received value. This can be modelled as follows:

$$\begin{aligned}
& x_p^i(k+1) \\
&= \Pi_p^i x_p^i(k) + \Xi_p^i F^i u^i(k) + \Psi_p^i a^{i-1}(k-1), \\
& \vdots \\
& x_p^i(k+m) \\
&= (\Pi_p^i)^m x_p^i(k) \\
&+ \left((\Pi_p^i)^{m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Xi_p^i F^i u^i(k) \\
&+ \left((\Pi_p^i)^{m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i a^{i-1}(k-1), \\
& x_p^i(k+m+1) \\
&= (\Pi_p^i)^{m+1} x_p^i(k) \\
&+ \left((\Pi_p^i)^m + \dots + \Pi_p^i + \mathcal{I} \right) \Xi_p^i F^i u^i(k) \\
&+ \Pi_p^i \left((\Pi_p^i)^{m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i a^{i-1}(k-1) \\
&+ \Pi_p^i \Psi_p^i a^{i-1}(k), \\
& x_p^i(k+m+2) \\
&= (\Pi_p^i)^{m+2} x_p^i(k) \\
&+ \left((\Pi_p^i)^{m+1} + \dots + \Pi_p^i + \mathcal{I} \right) \Xi_p^i F^i u^i(k) \\
&+ \Pi_p^{i2} \left((\Pi_p^i)^{m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i a^{i-1}(k-1) \\
&+ (\Pi_p^i + \mathcal{I}) \Psi_p^i a^{i-1}(k), \\
& \vdots \\
& x_p^i(k+\eta) \\
&= (\Pi_p^i)^\eta x_p^i(k) \\
&+ \left((\Pi_p^i)^{\eta-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Xi_p^i F^i u^i(k) \\
&+ (\Pi_p^i)^{\eta-m} \left((\Pi_p^i)^{m-1} + \dots + \mathcal{I} \right) \Psi_p^i a^{i-1}(k-1) \\
&+ \left((\Pi_p^i)^{\eta-m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i a^{i-1}(k).
\end{aligned}$$

Thus, the platoon model that captures the lower-layer dynamics, the upper-layer sampling period h_u , and the communication delay can be represented as,

$$x^i(k+1) = \alpha^i x^i(k) + \beta^i u^i(k) + \gamma_1^i a^{i-1}(k-1) + \gamma_2^i a^{i-1}(k), \quad (16)$$

where,

$$\begin{aligned}
\gamma_1^i &:= (\Pi_p^i)^{\eta-m} \left((\Pi_p^i)^{m-1} + \dots + \mathcal{I} \right) \Psi_p^i, \\
\gamma_2^i &:= \left((\Pi_p^i)^{\eta-m-1} + \dots + \Pi_p^i + \mathcal{I} \right) \Psi_p^i, \quad (17)
\end{aligned}$$

where \mathcal{I} , α^i and β^i are the same as those defined in Eq. 14. γ_1^i and $\gamma_2^i \in \mathbb{R}^{4 \times 1}$.

1) AUGMENTED MODEL

The model in Eq. 16 has one additional term with respect to the standard non-delayed model Eq. 11. To deal with this non-standard form, we augment the system state as follows,

$$z^i(k) = \begin{bmatrix} x^i(k) \\ a^{i-1}(k) \end{bmatrix} \in \mathbb{R}^{5 \times 1}. \quad (18)$$

Then the platoon model in Eq. 16 can be represented in the standard form similar to Eq. 11 using the new system state z^i :

$$z^i(k+1) = \bar{\alpha}^i z^i(k) + \bar{\beta}^i u^i(k) + \bar{\gamma}^i a^{i-1}(k-1), \quad (19)$$

where,

$$\bar{\alpha}^i = \begin{bmatrix} \alpha^i & \gamma_2^i \\ 0 & 0 \end{bmatrix}, \quad \bar{\beta}^i = \begin{bmatrix} \beta^i \\ 0 \end{bmatrix}, \quad \bar{\gamma}^i = \begin{bmatrix} \gamma_1^i \\ 0 \end{bmatrix}, \quad (20)$$

and,

$$\bar{\alpha}^i \in \mathbb{R}^{5 \times 5}, \quad \bar{\beta}^i \in \mathbb{R}^{5 \times 1}, \quad \bar{\gamma}^i \in \mathbb{R}^{5 \times 1}.$$

Note that in previous augmented system we assumed that

$$a^{i-1}(k+1) = 0.$$

In other words, the acceleration of the preceding vehicle at the next time step $(k+1)h_u$ is assumed to be zero (i.e., the vehicle moves with a constant velocity).

VI. STABILITY ANALYSIS OF THE PLATOON SYSTEM

The multi-rate multi-layer platoon system derived in Eq. 19 is composed of five system parameters: the upper-layer sampling period h_u , the lower-layer sampling period h_l , the headway time τ_h , the delay $\tau \leq h_u$ and the pole locations of the lower-layer controller. Changing one of these parameters will lead to a different system and input matrices. Therefore these parameters have a direct influence on the computed eigenvalues and thus, the stability of the platoon system. In this section, we investigate the stability region of the platoon system under different sets of parameters. In other words, we would like to know which sets of parameters ensure that the system is stable (the eigenvalues of the closed-loop system matrix are in the unit circle [4]). This is done by checking the *stabilizability* of the closed-loop system i.e., whether there exists a linear control law that makes the system Eq. 19 exponentially stable.

To obtain the closed-loop system, let us augment the state $a^{i-1}(k-1)$ to the state $z^i(k)$. The new augmented state is called $\bar{z}^i(k)$, where

$$\bar{z}^i(k) = \begin{bmatrix} z^i(k) \\ a^{i-1}(k-1) \end{bmatrix}.$$

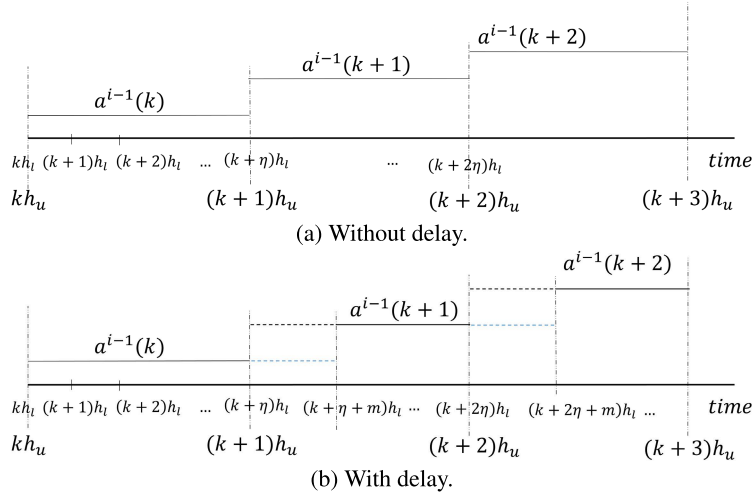


FIGURE 4. Relation between h_l and h_u . The acceleration of the preceding vehicle is received without delay in (a) and with delay in (b). In (b), the signals $a^{i-1}(k+1)$, $a^{i-1}(k+2)$ are received with delay τ , i.e., after m lower-layer samples.

This augmentation is necessary to check stabilizability as the dynamics of the platoon system depends not only on γ_2^i but also on γ_1^i . Note that $a^{i-1}(k-1)$ in Eq. 19 can be considered as an external input. Eq. 19 can be represented as,

$$\bar{z}^i(k+1) = \underbrace{\begin{bmatrix} \alpha^i & \gamma_2^i & \gamma_1^i \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\tilde{\alpha}^i \in \mathbb{R}^{6 \times 6}} \bar{z}^i(k) + \underbrace{\begin{bmatrix} \tilde{\beta}^i \\ 0 \end{bmatrix}}_{\tilde{\beta}^i \in \mathbb{R}^{6 \times 1}} u^i(k). \quad (21)$$

For the stability analysis, let us consider the following control law: $u^i(k) = \mathbf{K}\bar{z}^i(k)$. Then the closed-loop system of Eq. 21 is represented as:

$$\bar{z}^i(k+1) = (\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K}) \bar{z}^i(k), \quad (22)$$

for any $\mathbf{K} \in \mathbb{R}^{1 \times 6}$ that stabilizes the system (i.e., the eigenvalues of $|\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K}|$ are < 1). Clearly, the pair $(\tilde{\alpha}^i, \tilde{\beta}^i)$ is uncontrollable. The stability of such discrete time system is guaranteed if there exists a *Lyapunov function* for the system.

Definition 1 (Control Lyapunov Functions): For the discrete-time system of Eq. 22, a function $\mathbf{V} : \mathbb{X} \rightarrow \mathbb{R}^+$ defined on a region $\mathbb{X} \subseteq \mathbb{R}^n$ and containing the origin in its interior is called a *Lyapunov function* if:

- 1) $\mathbf{V}(0) = 0$,
- 2) $\mathbf{V}(\xi) > 0 \forall \xi \in \mathbb{X} \setminus \{0\}$,
- 3) There exists a control law $u^i(k) = \mathbf{K}\bar{z}^i(k)$ such that:

$$\mathbf{V}(\bar{z}^i(k+1)) - \mathbf{V}(\bar{z}^i(k)) < 0, \quad \forall \bar{z}^i(k) \in \mathbb{X}.$$

The asymptotic stability can be guaranteed for our closed-loop system if there exists a quadratic Lyapunov function $\mathbf{V}(\xi) = \xi^T P \xi$ satisfying the aforementioned conditions. The first condition is obviously satisfied ($\mathbf{V}(0) = 0$). The second condition is satisfied if P is positive definite, i.e., $P > 0$. The third condition is satisfied if:

$$\bar{z}^i(k)^T [(\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K})^T P (\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K}) - P] \bar{z}^i(k) < 0.$$

This can be achieved by finding a solution for the following LMI,

$$(\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K})^T P (\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K}) - P < 0,$$

for $P > 0$. Applying Schur's complement on the previous LMI gives the following,

$$\begin{pmatrix} O & (\tilde{\alpha}^i O + \tilde{\beta}^i Y)^T & O \\ (\tilde{\alpha}^i O + \tilde{\beta}^i Y) & O & 0 \\ O & 0 & Z^{-1} \end{pmatrix} \geq 0,$$

where $O = P^{-1}$, $Y = \mathbf{K}P^{-1}$. We used Yalmip [41] to solve the previous LMI for any given $Z > 0$. If we can find $P > 0$ and any \mathbf{K} such that the eigenvalues of $|\tilde{\alpha}^i + \tilde{\beta}^i \mathbf{K}|$ are < 1 , then the stability is guaranteed. Otherwise, we cannot say if the system is stable or not since finding a Lyapunov function is a sufficient but not a necessary condition for guaranteeing stability.

Table 2 shows the stability region under the upper-layer sampling periods $h_u = 114\text{ms}$ for different pole locations of the lower-layer controller and with different headway times τ_h and lower-layer sampling periods h_l . The stability is tested by trying to solve the previous LMI while changing the delay values and checking if there exists a Lyapunov matrix $P > 0$. A table entry $\tau \leq 114\text{ms}$ means that $P > 0$ can be found for all $\tau \leq h_u$, i.e., asymptotic stability is guaranteed for all delay values that are less than the sampling period. In some cases, asymptotic stability cannot be guaranteed for all $\tau \leq h_u$. In Table 2, “—” refers to such cases where the LMI cannot be solved and, therefore, asymptotic stability cannot be guaranteed. In other cases, asymptotic stability can be guaranteed for only zero delay. It is noted that with poles closer to zero, the stability region in Table 2 is larger than the cases where poles are closer to one.

From Table 2, it can be concluded that under certain sets of parameters the stabilizability of the platoon system can be guaranteed. This table can be used as a reference while

designing the upper-layer controller using MPC since it is expected to see similar behaviour under similar sets of parameters. However, the direct relation between the stabilizability using a linear controller and MPC cannot be trivially established and is part of our future research.

VII. UPPER-LAYER CONTROLLER USING MPC

A. UPPER-LAYER CONTROL

The role of the upper-layer controller is to compute the desired acceleration considering the platoon model in Eq. 19.

Following the speed of the preceding vehicle, avoiding cut-ins from adjacent lanes and rear-end collisions, minimizing the spacing error between vehicles and sudden changes in acceleration (for comfortable driving) are important objectives that should be satisfied by each platoon vehicle. In addition to that, acceleration and speed should not exceed certain bounds. Model Predictive Control (MPC) [43] is the control framework that can handle these objectives and constraints. MPC is widely used in industry for its ability to handle highly complex multi-variable processes with constraints on inputs, internal states and outputs.

MPC solves an optimization problem on-line every time step based on a model of the system (prediction model) and constraints on input and states. At each time step k , the current state of the system is fed back to the MPC controller and by using the prediction model, the future evolution of the system is calculated from time step k to time step $k + N_p$, where N_p is the prediction horizon length. Next, a predefined cost function is minimized taking into account the constraints defined on inputs and states. The output of this optimization problem is a sequence of optimal control inputs $u_{k|k}, \dots, u_{N_p-1+k|k}$. Only the first input $u_{k|k}$ is applied to the system and the whole process is repeated in the next time step $k + 1$.

In this section, we show how to formulate our optimization problem into the following standard (convex) quadratic programming (QP) form.

$$\min_{U_k^i} \frac{1}{2} (U_k^i)^T G^i U_k^i + (U_k^i)^T \mathcal{F}^i \quad (23)$$

$$\text{s.t. } LU_k^i \leq C + wx^i(k) \quad (24)$$

The QP form is preferred since a global minimum can be found and different QP solvers can be used. In the following, we provide more details on how to formulate the MPC problem in QP form while considering communication delay. The prediction model, the cost function and the constraints are represented in a more compact form.

B. MPC QP FORM

1) PREDICTION MODEL

Over a horizon of length N_p , the prediction model is required to anticipate the future evolution of the system states. Therefore, after minimizing the cost function, a sequence of length N_p of the optimal control inputs can be obtained that satisfy the control objectives, and state and input constraints.

The prediction model for vehicle i can be obtained from the platoon model (Eq. 19) and written as following,

$$z_{k+j+1|k}^i = \bar{\alpha}^i z_{k+j|k}^i + \bar{\beta}^i u_{k+j|k}^i + \bar{\gamma}^i a_{k+j|k-1}^{i-1}, \quad j = 0, \dots, N_p - 1. \quad (25)$$

In the previous equation, $z_{k+j+1|k}^i \in \mathbb{R}^{5 \times 1}$ represents the predicted state at step $k + j + 1$ when the prediction is done at step k for vehicle i . For $j = 0$,

$$z_{k|k}^i = z^i(k), \quad \text{where } z^i(k) = \begin{bmatrix} a^i(k) \\ \delta a^i(k) \\ \Delta d^i(k) \\ \Delta v^i(k) \\ a^{i-1}(k) \end{bmatrix} \in \mathbb{R}^{5 \times 1},$$

i.e., the predicted state at step k when the prediction is done at step k is equal to the current measured state of vehicle i . Similarly, $u_{k+j|k}^i$ for $j = 0, \dots, N_p - 1$ represents the calculated optimal control inputs over the prediction horizon N_p . It is assumed that the predicted acceleration values $a_{k+j|k-1}^{i-1}$ ($j = 0, \dots, N_p - 1$) of vehicle $i - 1$ are constants and equal to the previously received acceleration value $a^{i-1}(k - 1)$ (since the future evolution of the preceding vehicle is not known in advance), i.e., the actual acceleration of vehicle $i - 1$ received at step $k - 1$ (every h_u).

To write the prediction model Eq. 25 in a more compact form, we unroll the loop over the prediction horizon N_p and by recursively substituting $z_{k+j-1|k}^i$ into $z_{k+j|k}^i$ we can rewrite the predicted state $z_{k+j|k}^i$ as a function of the current measured state (since $z_{k|k}^i = z^i(k)$) and the predicted control inputs $u_{k|k}^i, \dots, u_{k+N_p-1|k}^i$.

The predicted states can be combined and represented into the following matrix formulation,

$$Z_k^i = \bar{\zeta}^i z^i(k) + \bar{v}^i U_k^i + \bar{x}^i \Lambda_{k-1}^{i-1}, \quad (26)$$

where,

$$\begin{aligned} Z_k^i &= \begin{bmatrix} z_{k+1|k}^i \\ z_{k+2|k}^i \\ \vdots \\ z_{k+N_p|k}^i \end{bmatrix}, \quad \bar{\zeta}^i = \begin{bmatrix} \bar{\alpha}^i \\ (\bar{\alpha}^i)^2 \\ \vdots \\ (\bar{\alpha}^i)^{N_p-1} \end{bmatrix}, \\ \bar{v}^i &= \begin{bmatrix} \bar{\beta}^i & 0 & \dots & 0 \\ \bar{\alpha}^i \bar{\beta}^i & \bar{\beta}^i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\bar{\alpha}^i)^{N_p-1} \bar{\beta}^i & (\bar{\alpha}^i)^{N_p-2} \bar{\beta}^i & \dots & \bar{\beta}^i \end{bmatrix}, \\ U_k^i &= \begin{bmatrix} u_{k|k}^i \\ u_{k+1|k}^i \\ \vdots \\ u_{k+N_p-1|k}^i \end{bmatrix}, \quad \Lambda_{k-1}^{i-1} = \begin{bmatrix} a^{i-1}(k-1) \\ a^{i-1}(k-1) \\ \vdots \\ a^{i-1}(k-1) \end{bmatrix}, \\ \bar{x}^i &= \begin{bmatrix} \bar{\gamma}^i & 0 & \dots & 0 \\ \bar{\alpha}^i \bar{\gamma}^i & \bar{\gamma}^i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (\bar{\alpha}^i)^{N_p-1} \bar{\gamma}^i & (\bar{\alpha}^i)^{N_p-2} \bar{\gamma}^i & \dots & \bar{\gamma}^i \end{bmatrix}. \end{aligned}$$

TABLE 2. Stability region of the platoon system for different pole locations, headway times and lower-layer sampling periods under an upper-layer sampling period $h_u = 114$ ms. Asymptotic stability is guaranteed for some cases where the delay τ is at most h_u . An entry “–” means that asymptotic stability cannot be guaranteed.

Lower layer sampling period h_l (ms)		Poles at [0.9, 0.9]				Poles at [0.75, 0.75]				Poles at [0.6, 0.6]			
		2	5	10	20	2	5	10	20	2	5	10	20
Headway time τ_h (seconds)	1	$\tau \leq 114$	–	–	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$
	0.8	$\tau \leq 114$	–	–	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$
	0.6	$\tau \leq 114$	–	–	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$
	0.4	$\tau \leq 114$	–	–	–	$\tau \leq 114$	$\tau \leq 114$	$\tau = 0$	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 40$
	0.2	$\tau \leq 114$	–	–	–	$\tau \leq 114$	$\tau \leq 114$	$\tau = 0$	–	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 114$	$\tau \leq 80$

2) COST FUNCTION

The quadratic cost function over the horizon N_p is formulated as follows,

$$\begin{aligned}
 J(z^i(k), U_k^i) &= (z_{k+N_p|k}^i)^T \bar{P} (z_{k+N_p|k}^i) \\
 &+ \sum_{j=0}^{N_p-1} \left[(z_{k+j|k}^i)^T \bar{Q} (z_{k+j|k}^i) + (u_{k+j|k}^i)^T R (u_{k+j|k}^i) \right].
 \end{aligned} \quad (27)$$

The elements of the weighting matrix \bar{Q} are the corresponding weighting coefficients that penalize the distance of the predicted states $z_{k+j|k}^i$ to zero. Similarly, R is the weighting parameter that penalizes the control inputs $u_{k+j|k}^i$. \bar{Q} and R are defined as follows,

$$\bar{Q} = \begin{bmatrix} w_a & 0 & 0 & 0 & 0 \\ 0 & w_{da} & 0 & 0 & 0 \\ 0 & 0 & w_{\Delta d} & 0 & 0 \\ 0 & 0 & 0 & w_{\Delta v} & 0 \\ 0 & 0 & 0 & 0 & w_a^{prec} \end{bmatrix} \in \mathbb{R}^{5 \times 5},$$

$$R = [w_u] \text{ (scalar)}.$$

\bar{P} and \bar{Q} are positive semi-definite matrices ($\bar{P} \geq 0$ and $\bar{Q} \geq 0$). R is positive definite i.e., $R > 0$. \bar{P} is the terminal cost that penalizes the terminal state $z_{k+N_p|k}^i$.

Defining the cost function in this form ensures that tracking capability is achieved by minimizing the position error Δd^i and the velocity error Δv^i . This also helps to avoid cut-ins from adjacent lanes in case that the preceding vehicle moves with constant speed (no vehicles will occupy the small gap between platoon vehicles). Also, by minimizing acceleration and rate of change of acceleration in this cost function, better fuel economy is achieved since it is directly affected by acceleration and its rate of change [62].

It should be noted that the desired acceleration a_{des}^i (the control input u^i that has to be computed using MPC and delivered to the lower-layer) and vehicle acceleration a^i (the lower-layer control output) have a similar effect on improving fuel efficiency; thus, to avoid redundancy w_a can be set to zero. Moreover, by minimizing the rate of change of acceleration δa , driving comfort can be achieved.

By expanding the summation in Eq. 27 over the horizon N_p , the cost function can be represented as a function of the predicted states and calculated inputs as follows,

$$\begin{aligned}
 J(z^i(k), U_k^i) &= (z_{k|k}^i)^T \bar{Q} z_{k|k}^i \\
 &+ \underbrace{\begin{bmatrix} z_{k+1|k}^i \\ z_{k+2|k}^i \\ \vdots \\ z_{k+N_p|k}^i \end{bmatrix}^T}_{(Z_k^i)^T} \underbrace{\begin{bmatrix} \bar{Q} & 0 & \dots & 0 \\ 0 & \bar{Q} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{P} \end{bmatrix}}_{\bar{Q}} \underbrace{\begin{bmatrix} z_{k+1|k}^i \\ z_{k+2|k}^i \\ \vdots \\ z_{k+N_p|k}^i \end{bmatrix}}_{Z_k^i} \\
 &+ \underbrace{\begin{bmatrix} u_{k|k}^i \\ u_{k+1|k}^i \\ \vdots \\ u_{k+N_p-1|k}^i \end{bmatrix}^T}_{(U_k^i)^T} \underbrace{\begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R \end{bmatrix}}_{\bar{R}} \underbrace{\begin{bmatrix} u_{k|k}^i \\ u_{k+1|k}^i \\ \vdots \\ u_{k+N_p-1|k}^i \end{bmatrix}}_{U_k^i}.
 \end{aligned}$$

Since $z_{k|k}^i = z^i(k)$, then we have,

$$J(z^i(k), U_k^i) = (z^i(k))^T \bar{Q} z^i(k) + (Z_k^i)^T \bar{Q} Z_k^i + (U_k^i)^T \bar{R} U_k^i. \quad (28)$$

The above cost function can be simplified by using the definition of Z_k^i (see Eq. 26) which yields,

$$\begin{aligned}
 J(z^i(k), U_k^i) &= z^i(k)^T \bar{Q} z^i(k) + (U_k^i)^T \bar{R} U_k^i \\
 &= (\bar{\xi}^i z^i(k) + \bar{v}^i U_k^i + \bar{x}^i \Lambda_{k-1}^{i-1})^T \\
 &\quad \times \bar{Q} (\bar{\xi}^i z^i(k) + \bar{v}^i U_k^i + \bar{x}^i \Lambda_{k-1}^{i-1}).
 \end{aligned}$$

After simplifying the above expression, the cost function can be expressed more compactly as,

$$J(z^i(k), U_k^i) = \frac{1}{2} (U_k^i)^T \bar{G}^i U_k^i + U_k^{iT} \bar{F}^i, \quad (29)$$

where,

$$\bar{G}^i = 2 \left(\bar{R} + (\bar{v}^i)^T \bar{Q} \bar{v}^i \right), \quad (30)$$

$$\bar{F}^i = 2 (\bar{v}^i)^T \bar{Q} \left(\bar{\xi}^i z^i(k) + \bar{x}^i \Lambda_{k-1}^{i-1} \right). \quad (31)$$

It is noted that while simplifying the above formula the terms that are not functions of the predicted control input U_k^i were

omitted. They can be considered as constants and the value that minimizes the cost function will not change by adding constants to the cost function.

3) CONSTRAINTS MATRICES

The constraints defined in Eq. 32 are added so that objectives such as driving comfort and fuel economy can be achieved. In other words, acceleration ($a^i(k)$), desired acceleration ($a_{des}^i(k)$) and rate of change of acceleration ($\delta a^i(k)$) are bounded within certain limits.

$$\begin{aligned} a_{min} &\leq a^i(k) \leq a_{max}, \\ a_{min} &\leq a^{i-1}(k) \leq a_{max}, \\ a_{min} &\leq a_{des}^i(k) \leq a_{max}, \\ \delta a_{min} &\leq \delta a^i(k) \leq \delta a_{max}, \end{aligned} \quad (32)$$

where a_{min} and a_{max} are the lower and upper bounds of the acceleration. δa_{min} and δa_{max} are the lower and upper bounds of the rate of change of acceleration.

Moreover, inequalities defined in Eq. 33 are necessary, when the preceding vehicle accelerates or decelerates, to guarantee that tracking errors do not exceed specific bounds. Therefore large inter-vehicle gaps are avoided between vehicles to prevent cut-ins from adjacent lanes.

$$\begin{aligned} \Delta d_{min} &\leq \Delta d^i \leq \Delta d_{max}, \\ \Delta v_{min} &\leq \Delta v^i \leq \Delta v_{max}, \end{aligned} \quad (33)$$

where Δd_{min} and Δd_{max} are the lower and upper bounds of the position error. Δv_{min} and Δv_{max} are the lower and upper bounds of the velocity error.

To avoid rear-end collision, when the preceding vehicle begins decelerating, we add the inequality 34. The actual gap between vehicles d^i is bounded from above by d_{max} and from below by d_0 .

$$d_0 \leq d^i \leq d_{max}. \quad (34)$$

Then, from Eq. 33 and Eq. 34 the constraint on the system state is defined as follows,

$$z_{min} \leq z^i(k) \leq z_{max}, \quad (35)$$

$$\text{where } z_{min} = \begin{bmatrix} a_{min} \\ \delta a_{min} \\ \Delta d_{min} \\ \Delta v_{min} \\ a_{min} \end{bmatrix} \text{ and } z_{max} = \begin{bmatrix} a_{max} \\ \delta a_{max} \\ \Delta d_{max} \\ \Delta v_{max} \\ a_{max} \end{bmatrix}.$$

To include the inequality Eq. 34 to Eq. 35, we need to rewrite d^i as a function of the system states. By knowing that $d^i = \Delta d^i + d_{des}^i$, $d_{des}^i = d_0 + \tau_h v^i$, $v^i = v^{i-1} - \Delta v^i$, inequality Eq. 34 can be written as,

$$-\tau_h v^{i-1} \leq \Delta d^i - \tau_h \Delta v^i \leq d_{max} - d_0 - \tau_h v^{i-1}. \quad (36)$$

Therefore, the constraint on the state $z^i(k)$ is defined as,

$$\underbrace{\begin{bmatrix} a_{min} \\ \delta a_{min} \\ \Delta d_{min} \\ \Delta v_{min} \\ -\tau_h v^{i-1} \\ a_{min} \end{bmatrix}}_{\tilde{z}_{min}} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -\tau_h & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} a^i(k) \\ \delta a^i(k) \\ \Delta d^i(k) \\ \Delta v^i(k) \\ a^{i-1}(k) \end{bmatrix}}_{\tilde{z}_{max}} \leq \underbrace{\begin{bmatrix} a_{max} \\ \delta a_{max} \\ \Delta d_{max} \\ \Delta v_{max} \\ d_{max} - d_0 - \tau_h v^{i-1} \\ a_{max} \end{bmatrix}}_{\tilde{z}_{max}}, \quad (37)$$

or,

$$\tilde{z}_{min} \leq \tilde{A} z^i(k) \leq \tilde{z}_{max}. \quad (38)$$

The constraints of the predicted states over the horizon N_p can be obtained from the previous inequality as follows,

$$\tilde{z}_{min} \leq \tilde{A} z_{k+j+1|k}^i \leq \tilde{z}_{max}, \quad j = 0, \dots, N_p - 1. \quad (39)$$

Similarly, the constraints on the predicted control inputs over the horizon N_p can be obtained from the control input constraint

$$a_{min} \leq u^i(k) \leq a_{max}$$

as follows,

$$a_{min} \leq u_{k+j|k}^i \leq a_{max}, \quad j = 0, \dots, N_p - 1. \quad (40)$$

Combining Eq. 39 and Eq. 40 into one inequality, we obtain the following,

$$\underbrace{\begin{bmatrix} \mathbf{0}_{1 \times 5} \\ \mathbf{0}_{1 \times 5} \\ -\tilde{A}_{6 \times 5} \\ \tilde{A}_{6 \times 5} \end{bmatrix}}_{\tilde{M}_j} z_{k+j+1|k}^i + \underbrace{\begin{bmatrix} -1 \\ 1 \\ \mathbf{0}_{6 \times 1} \\ \mathbf{0}_{6 \times 1} \end{bmatrix}}_{\tilde{E}_j} u_{k+j|k}^i \leq \underbrace{\begin{bmatrix} -a_{min} \\ a_{max} \\ -\tilde{z}_{min} \\ \tilde{z}_{max} \end{bmatrix}}_{\tilde{b}_j},$$

or,

$$\tilde{M}_j z_{k+j+1|k}^i + \tilde{E}_j u_{k+j|k}^i \leq \tilde{b}_j, \quad j = 0, \dots, N_p - 1, \quad (41)$$

where $\tilde{M}_j \in \mathbb{R}^{14 \times 5}$, $\tilde{E}_j \in \mathbb{R}^{14 \times 1}$, $\tilde{b}_j \in \mathbb{R}^{14 \times 1}$. By separating the inequality on the terminal constraint, we obtain the following,

$$\tilde{M}_j z_{k+j|k}^i + \tilde{E}_j u_{k+j|k}^i \leq \tilde{b}_j, \quad j = 0, \dots, N_p - 1 \quad (42)$$

$$\tilde{M}_{N_p} z_{k+N_p|k}^i \leq \tilde{b}_{N_p}. \quad (43)$$

The terminal constraint matrices \tilde{M}_{N_p} and \tilde{b}_{N_p} have to be computed for ensuring stability [5]. We consider \tilde{M}_{N_p} and \tilde{b}_{N_p} to be zero. It is proved in [37] that stability of MPC can be obtained when considering a zero terminal constraint set.

By expanding the above inequalities over the horizon N_p , we obtain the following,

$$\underbrace{\begin{bmatrix} \bar{M}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\bar{D}} z_{k|k}^i + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ \bar{M}_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{M}_{N_p} \end{bmatrix}}_{\bar{M}} \underbrace{\begin{bmatrix} z_{k+1|k}^i \\ z_{k+2|k}^i \\ \vdots \\ z_{k+N_p|k}^i \end{bmatrix}}_{Z_k^i} + \underbrace{\begin{bmatrix} \bar{E}_0 & 0 & \dots & 0 \\ 0 & \bar{E}_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{E}_{N_p-1} \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\bar{I}} \underbrace{\begin{bmatrix} u_{k|k}^i \\ u_{k+1|k}^i \\ \vdots \\ u_{k+N_p-1|k}^i \end{bmatrix}}_{U_k^i} \leq \underbrace{\begin{bmatrix} \bar{b}_0 \\ \bar{b}_1 \\ \vdots \\ \bar{b}_{N_p} \end{bmatrix}}_{\bar{C}}.$$

Knowing that $z_{k|k}^i = z^i(k)$, then,

$$\bar{D}z^i(k) + \bar{M}Z_k^i + \bar{I}U_k^i \leq \bar{C}. \quad (44)$$

Using the definition of Z_k^i (Eq. 26), the constraints can be rewritten as,

$$\bar{L}U_k^i \leq \bar{C} + \bar{w}z^i(k) - \bar{M}\bar{x}^i \Lambda_{k-1}^{i-1} \quad (45)$$

where $\bar{L} = \bar{I} + \bar{M}\bar{v}^i$ and $\bar{w} = -\bar{D} - \bar{M}\bar{\xi}^i$.

C. NUMERICAL COMPUTATION

Minimizing the cost function Eq. 29 while respecting the constraints in Eq. 45 gives a sequence of optimal control inputs as in the following form:

$$(U_k^i)^* := \begin{bmatrix} (u_{k|k}^i)^* \\ (u_{k+1|k}^i)^* \\ \vdots \\ (u_{k+N_p|k}^i)^* \end{bmatrix}. \quad (46)$$

Only the first optimal value $(u_{k|k}^i)^*$ has to be applied and the other elements will be discarded. At the next step $k+1$, a new sequence $(U_{k+1}^i)^*$ is computed and the first optimal value $(u_{k+1|k+1}^i)^*$ will be applied.

D. INTEGRATED MPC AND DCC

MPC is implemented with the upper-layer sampling periods h_u corresponding to the message rates mentioned in Table 1. These message rates correspond to the particular traffic densities discussed earlier and shown also in Table 1. For other traffic densities, the message rate and therefore the sampling periods h_u will have different values. The proposed MPC controller is tuned for each upper-layer sampling periods h_u shown in Fig. 5 and the switching happens between these MPC controllers based on the traffic density. In other words, based on the channel congestion level, the DCC algorithm (and hence MPC controller) switches between certain message rates e.g., 8.75Hz (114ms) \leftrightarrow 7.5Hz (133ms). We assume that the switching between different MPC controllers

is slow enough not to cause any switching instabilities. This is a realistic assumption in real traffic conditions, where vehicular density does not change quickly to cause stability problems.

E. PACKET LOSS

In case of packet loss or if a packet is received with delay greater than the upper-layer sampling period, i.e., $\tau > h_u$, the next elements in the optimal control input sequence in Eq. 46 can be used. In other words, if the received signal at time step $k+1$ is lost or delayed, and therefore the new optimal input sequence $(U_{k+1}^i)^*$ cannot be computed, the second optimal value in Eq. 46 $(u_{k+1|k}^i)^*$ is used as a control input. Similarly, if the packet is not received at time step $k+2$, the third optimal value $(u_{k+2|k}^i)^*$ is used. In case of persistent packet loss in the wireless link, the platoon control should switch from CACC to ACC where the control action is taken based on the information received from on-board sensors such as radar or lidar. This is beyond the scope of this paper.

VIII. RESULTS

In this section, we discuss the performance of the multi-rate multi-layer system under communication delay and DCC in terms of maintaining string stability, safety and fuel efficiency. The analysis is done with respect to three independent parameters: communication delay τ , headway time τ_h and V2V message rates (upper-layer sampling rate h_u). First, we analyse the performance and stability in case of considering and not considering communication delay τ . In this case, we use Matlab to generate the simulation results since we only focus on analysing the stability and safety of the control system with and without considering τ in the platoon model in Eq. 16. Second, we investigate how engaging DCC affects the stability and fuel efficiency of the platoon control system under heavy network and traffic congestion levels. In this case, we use our co-simulation framework CReTS to obtain the network and control performance under different network and traffic simulation scenarios. Finally, we perform a fuel efficiency analysis.

We consider a platoon of five vehicles in all simulation scenarios. Platoon vehicles are denoted by veh0, ..., veh4 where veh0 refers to the platoon leader (first platoon vehicle). The first follower is named veh1 and so on. For the considered simulation scenarios, we use the following parameters: the lower-layer sampling rate $h_l = 2\text{ms}$ and the upper-layer sampling rates mentioned in Table 1 i.e., $h_u \in \{100, 114, 133, 152, 171, 191, 248, 286\}\text{ms}$. Therefore, η can be computed as shown in Eq. 8 (where the outcome is approximated to the nearest integer). The MPC weighting parameters are selected as follows: $w_a = 0$, $w_{\delta a} = 50$, $w_d = 1000$, $w_v = 1000$, $w_a^{prec} = 100$, $w_u = 200$. The prediction horizon N_p of MPC is chosen to be $N_p = 15$.

A. EFFECT OF DELAY MODELLING

In this section, we discuss the effect of delay modeling on the performance of the platoon vehicles in terms of maintaining

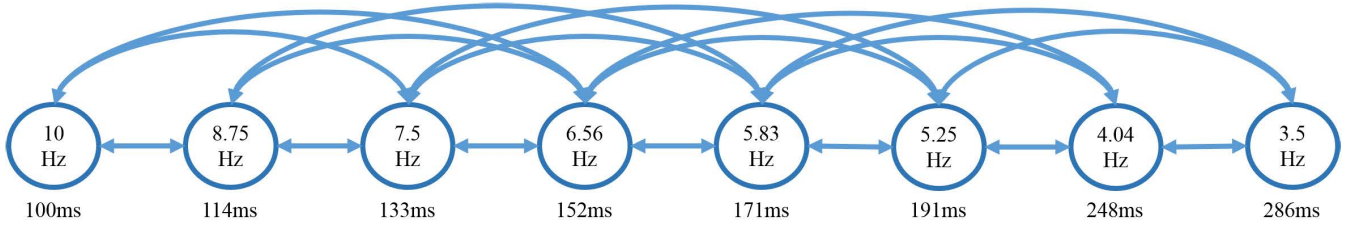


FIGURE 5. An example of the switching sequence between different V2V message rates, and therefore MPC controllers, due to varying network congestion levels and vehicular densities.

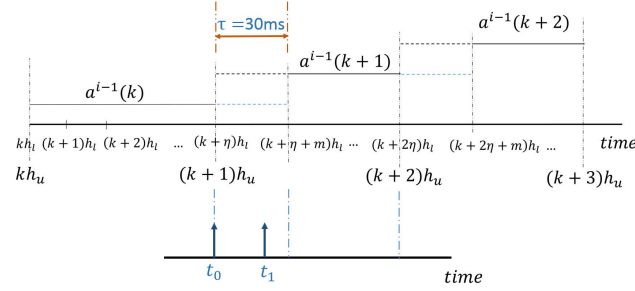


FIGURE 6. Examining the received acceleration in the interval $[(k+1)h_u, (k+2)h_u]$. At time t_0 , i.e., at the beginning of the sampling period, the old acceleration $a^{i-1}(k)$ already exists. The new acceleration $a^{i-1}(k+1)$ can be received at time t_1 where $t_1 - t_0 \leq \tau$ and $\tau = 30\text{ms}$, in this example. If $a^{i-1}(k+1)$ is received at time t_1 where $t_1 - t_0 > \tau$, then $a^{i-1}(k+1)$ is considered as a lost packet.

string stability, safety and achieving minimum position error. As shown in Fig. 6, we analyse the received signals in the interval $[(k+1)h_u, (k+2)h_u]$. We consider $\tau = 30\text{ms}$, i.e., the new acceleration $a^{i-1}(k+1)$ is expected to be received within 30ms. If the packet is received with delay $\tau > 30\text{ms}$, it is considered as a lost packet. Let us consider two points in time in the interval $[(k+1)h_u, (k+2)h_u]$, t_0 at the beginning and t_1 such that $t_1 - t_0 \leq \tau$. The controller designed without delay model, as presented in [25], expects that $a^{i-1}(k+1)$ is available at t_0 while only $a^{i-1}(k)$ is available at that point. Due to presence of delay, $a^{i-1}(k+1)$ is available at some point $t_1 > t_0$ and $t_1 - t_0 \leq \tau$. The controller without delay model actuates/updates at t_0 , and cannot use $a^{i-1}(k+1)$ before the next point in actuation at $t_0 + h_u$ since it is not available at t_0 . Therefore, in presence of delay, such a controller ends up using $a^{i-1}(k)$ for the entire sampling interval $[(k+1)h_u, (k+2)h_u]$. In essence, the controller without delay model will end up using one sample old update $a^{i-1}(k)$ in all samples.

Recall the platoon model Eq. 16 discussed earlier in Section V-D which captures the delay in the received signals:

$$x^i(k+2) = \alpha^i x^i(k+1) + \beta^i u^i(k+1) + \gamma_1^i a^{i-1}(k) + \gamma_2^i a^{i-1}(k+1).$$

This model represents the case shown in Fig. 6 where the focus is on the received acceleration in the interval $[(k+1)h_u, (k+2)h_u]$.

The effect of delay modeling is studied by considering two cases. Delay modeling is considered in *Case I* while delay modeling is not considered in *Case II*. The Matlab simulations are shown in Fig. 7 for the platoon vehicles under

headway time = 0.4s and upper-layer sampling rate $h_u = 191\text{ms}$.

- *Case I – with delay (the current paper)*: The new acceleration is received with delay $\tau \leq 30\text{ms}$ (at time t_1 in Fig. 6 with $t_1 - t_0 \leq 30\text{ms}$). The desired acceleration is then computed based on the old acceleration $a^{i-1}(k)$ and the new received acceleration $a^{i-1}(k+1)$. The old acceleration $a^{i-1}(k)$ is active for 30ms at the beginning and the new acceleration $a^{i-1}(k+1)$ is active for the rest of the sampling period. In this case γ_1^i and γ_2^i in the above equation exist. The performance of this case is shown in Fig. 7a, 7v, 7e, and 7g where string stability, safety and minimal position error are maintained.
- *Case II – without delay [25]*: The desired acceleration is computed based only on the old outdated acceleration $a^{i-1}(k)$. In other words, $a^{i-1}(k)$ is active for the entire sampling period, i.e., γ_1^i has a value and $\gamma_2^i = 0$ in the equation. The performance of this case is shown in Fig. 7b, 7d, 7f, and 7h. Fig. 7b, 7d shows string instability, i.e., the acceleration and velocity of the following vehicles have greater values than their leading vehicles in case of accelerating, and lower values in case of decelerating. Fig. 7f shows that the absolute position errors have greater bounds than in *Case I*. Moreover, a collision happens between the third and the fourth platoon vehicles. This is shown in Fig. 7h where the actual gap curve goes below zero.

Fig. 7 illustrates the effect of (not) taking into account delays for an upper-layer sampling period $h_u = 191\text{ms}$. This corresponds to a congested network state. The same effects occur also for smaller upper-layer sampling periods, including the non-congested state $h_u = 100\text{ms}$. The effects are less pronounced, but instability and collisions occur also in non-congested network conditions when delays are ignored.

B. PERFORMANCE OF THE INTEGRATED DCC-MPC

1) CReTS – CO-SIMULATION FRAMEWORK

CReTS [26] is a co-simulation framework composed of the network simulator ns-3, traffic simulator SUMO and Matlab. ns-3 implements the communication architecture for IEEE 802.11p and simulates V2V communication between vehicles. It provides packet reception ratios and delays to the controller implemented in Matlab. SUMO generates real driving behavior on highways or urban areas and provides a graphical user interface (GUI) to observe the motion of vehicles.

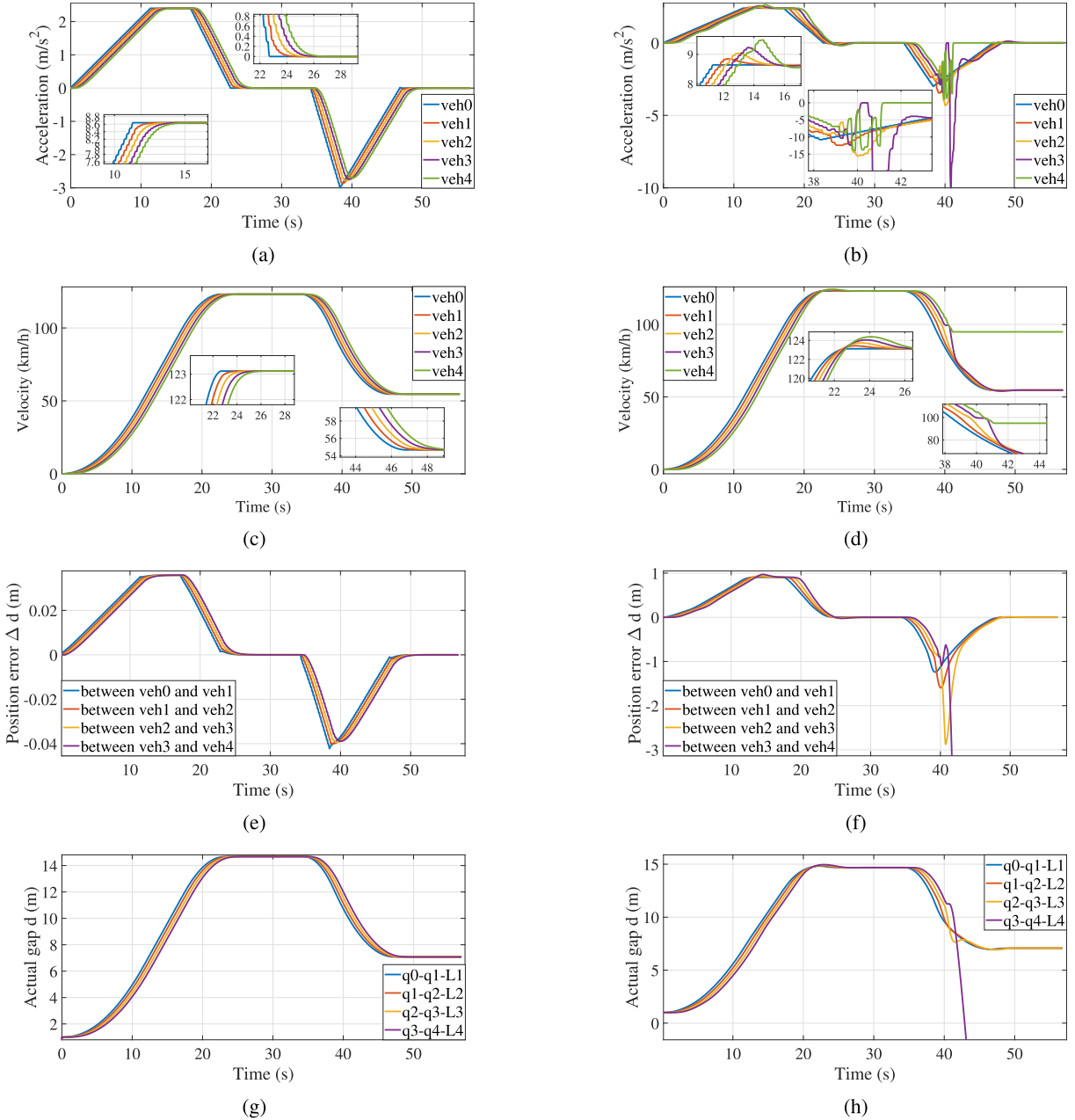


FIGURE 7. Effect of delay modeling on the performance of the platoon vehicles under headway time = 0.4s, upper-layer sampling rate $h_u = 191$ ms and delay $\tau = 30$ ms. The right column shows the case where the delay is not considered (using the old outdated acceleration for the entire sampling period to compute the desired acceleration). The left column shows the case where the delay is considered (using both the old outdated acceleration and the new updated acceleration). String stability is preserved and minimal position error is achieved if considering delay as seen in the figures of the left column. In contrast, string stability cannot be achieved and vehicle collision happens if delay is not considered as seen in the figures of the right column.

Different network congestion levels can be simulated by adjusting the number of communicating vehicles using SUMO and the corresponding control performance is evaluated using Matlab.

The ns-3 network simulator simulates communication between vehicles using dual-slope large scale fading and a Nakagami statistical channel model discussed earlier in Section IV-C. In ns-3, we consider the communication parameters listed in Table 3, which are reported in [16]. The channel model parameters are obtained from the highway

scenario specifications in the ETSI standard [17]. SUMO provides traffic/vehicle positions and motion, and each simulated platoon vehicle uses both SUMO and ns-3 to get traffic awareness input for platoon control.

To simulate a realistic highway in SUMO, we consider a road section of 3km with four lanes in each direction, as already elaborated in Section III and illustrated in Fig. 2. Vehicles move from the left to the right. They take the U-turn when they reach the end of the road and they keep moving in the opposite direction. The performance evaluation is

TABLE 3. Communication parameters considered in ns-3.

Carrier sensing threshold	-85 dBm		
Transmit power	24 dBm		
(Large-scale fading)	Path-loss exponent 1 (until 80 m)	1.9	
Dual slope model [18]	Path-loss exponent 2 (after 80 m)	3.8	
(Small-scale fading)	Distance bin in meters	\mathcal{M}	
	0-50	3	
	51-150	1.5	
	> 150	1	
Payload size	300Bytes		

TABLE 4. V2V links between platoon vehicles.

Link-0	Link-1	Link-2	Link-3
<i>veh0</i> → <i>veh1</i>	<i>veh0</i> → <i>veh2</i>	<i>veh0</i> → <i>veh3</i>	<i>veh0</i> → <i>veh4</i>
<i>veh1</i> → <i>veh2</i>	<i>veh1</i> → <i>veh3</i>	<i>veh1</i> → <i>veh4</i>	
<i>veh2</i> → <i>veh3</i>	<i>veh2</i> → <i>veh4</i>		
<i>veh3</i> → <i>veh4</i>			

restricted to the region of interest shown in Fig. 2 in order to eliminate the network boundaries.

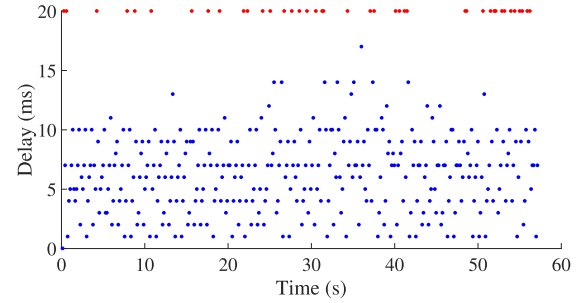
2) TRAFFIC SCENARIOS AND SIMULATION PARAMETERS

Table 1 shows the network and traffic scenarios under consideration. These scenarios are named Scenario-0–12 as shown in Table 5. The platoon vehicles and the additional traffic vehicles communicate via IEEE 802.11p leading to congestion in the network by increasing the number of vehicles in SUMO. message rate and therefore the upper-layer sampling rate is adapted in each scenario based on the vehicular density and network congestion level to keep the channel load $\leq 70\%$ (see Section IV-D). Low level of congestion is shown in Scenario-0, Scenario-1 and Scenario-2 where the standard V2V message rate of 10Hz is considered. Scenario-0 is the case where there is no additional traffic but only the five platoon vehicles. In Scenario-1, we consider the five platoon vehicles in addition to 295 other vehicles. In these two scenarios, the channel load is $< 70\%$. Scenario-2 (525 vehicles) considers the case where the channel load = 70%. From Scenario-3 to Scenario-9, the number of vehicles increases gradually and for each scenario the DCC algorithm is implemented by adjusting the message rate to keep the channel load $\leq 70\%$. Scenario-10–12 show the scenarios where the channel load is more than 100%.

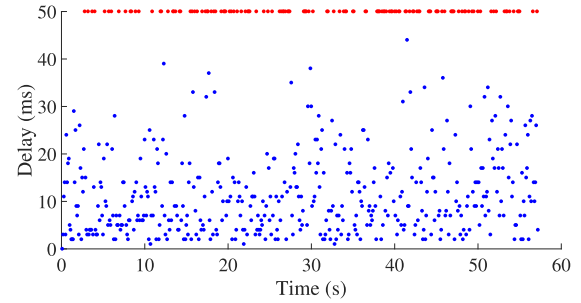
The data rate is 6Mbps for all scenarios, which is the value used in IEEE 802.11p V2X (vehicle-to-everything) communication.

3) NETWORK PERFORMANCE

The packet reception ratio (PRR) and average delay calculated by ns-3 for each scenario mentioned in Table 1 are averaged and mentioned in Table 5. The average is computed by running the simulation multiple times where SUMO places the vehicles in different locations in each run so we obtain non-identical results. We change the headway time and the



(a) 70% channel load.



(b) 106% channel load.

FIGURE 8. Delay and packet loss in Link-0 (veh2-veh3) of Scenario-5 under 70% channel load (top figure) and Scenario-10 under 106% channel load (bottom figure). The red dots indicate lost packets. The blue dots indicate received packets with the indicated delay.

delay for the different simulation runs for each scenario for a fair comparison. For example, we chose $\tau_h = 0.2s, 0.6s, 0.8, 1s$ and $\tau = 20ms, 30ms, 50ms$. It should be noted that τ here is the maximum delay based on the level of congestion which is measured via ns-3. τ is used in the control design, which means that any received signal beyond this value is considered as packet loss. This is different from the average delay calculated and mentioned in Table 5. Table 4 shows the wireless links between vehicles, which are used to compute the average PRR and delay shown in Table 5. Link-0 refers to the connection between any two consecutive vehicles where there is no vehicles in between e.g., the link between *veh0* and *veh1*. Link-1 shows a link where there is one vehicle between two platoon vehicles e.g., the link between *veh0* and *veh2*. Link-2 and Link-3 refer to the cases where there are two and three vehicles between the platoon vehicles, respectively.

The measured PRR in simulations shown in Table 5 are divided into three parts: (i) low network and traffic congestion where DCC is inactive i.e., Scenario-0–2 (5, 300, 525 vehicles), (ii) high network and traffic congestion where DCC is active i.e., Scenario-3–9 (600, 700, 800, 900, 1000, 1300, 1500 vehicles) (iii) high network and traffic congestion where DCC is inactive i.e., Scenario-10–12 (800, 900, 1000 vehicles).

Table 5 shows the following:

- The average PRR drops when the number of vehicles increases (i.e., Scenario-0–2).
- When the DCC is engaged, it reduces message rate to control the channel load below 70%; the PRR is around 80% (i.e., Scenario-3–9).

TABLE 5. Average packet reception ratio (PRR) and average delay in different scenarios experienced between platoon vehicles.

		Link-0	Link-1	Link-2	Link-3
Scenario-0 – 5 vehicles (10Hz – no DCC)	Average PRR %	100	100	100	100
	Average delay (ms)	2.9048	3.0848	3.9890	4.3527
Scenario-1 – 300 vehicles (10Hz – no DCC)	Average PRR %	94.8774	94.5320	93.3742	95.2715
	Average delay (ms)	3.2418	3.3380	4.2946	4.5752
Scenario-2 – 525 vehicles (10Hz – no DCC)	Average PRR %	85.3182	83.7128	83.1582	87.3322
	Average delay (ms)	5.3767	5.4551	5.9977	5.2633
Scenario-3 – 600 vehicles (8.75Hz – with DCC)	Average PRR %	82.7678	82.7678	76.7465	71.9228
	Average delay (ms)	8.1619	7.9722	8.7214	8.5968
Scenario-4 – 700 vehicles (7.5Hz – with DCC)	Average PRR %	80.2448	79.7591	76.1364	73.7762
	Average delay (ms)	7.6562	7.5318	7.6859	7.5410
Scenario-5 – 800 vehicles (6.56Hz – with DCC)	Average PRR %	82.9398	81.2997	78.5069	73.8042
	Average delay (ms)	7.4302	7.5190	8.0393	7.7650
Scenario-6 – 900 vehicles (5.833Hz – with DCC)	Average PRR %	79.2879	79.0159	76.5056	78.0228
	Average delay (ms)	7.6296	7.6679	7.8371	6.9911
Scenario-7 – 1000 vehicles (5.25Hz – with DCC)	Average PRR %	80.9354	79.9497	76.4681	73.5738
	Average delay (ms)	7.7058	7.8409	7.9378	7.5364
Scenario-8 – 1300 vehicles (4.0385Hz – with DCC)	Average PRR %	78.8934	77.6248	78.1250	77.8925
	Average delay (ms)	7.7400	7.7580	7.0701	7.0143
Scenario-9 – 1500 vehicles (3.5Hz – with DCC)	Average PRR %	80.6533	79.7878	77.7219	78.5595
	Average delay (ms)	7.5889	7.7086	7.4997	6.7823
Scenario-10 – 800 vehicles (10Hz – no DCC)	Average PRR %	69.0674	65.6451	59.8074	49.6497
	Average delay (ms)	11.7215	11.2659	11.1832	9.8844
Scenario-11 – 900 vehicles (10Hz – no DCC)	Average PRR %	61.2960	56.7426	49.8249	36.0771
	Average delay (ms)	15.9223	15.0640	14.1338	10.3107
Scenario-12 – 1000 vehicles (10Hz – no DCC)	Average PRR %	47.0541	40.8844	34.3364	24.1221
	Average delay (ms)	19.0651	17.5398	15.6794	10.3577

- With the DCC inactive and a high number of vehicles (≤ 1000 vehicles), the measured PRR drops further (i.e., Scenario-10–12).
- The average message delay in the DCC-active scenarios (3-9) is around 8ms, which reflects the maximum back-off counter in the 802.11p CSMA protocol. The average delay is around 4ms for Scenario-0–2 and it increases further for the DCC-inactive Scenario-10–12.

For vehicle densities above 1000 without DCC, Table 5 does not contain entries. The PRR drops further compared to the Table 5 entries for Scenarios 10-12 and the delay increases, implying that it is impossible to ensure string stability or to keep the platoon formation.

Fig. 8 shows the communication delay measured in the wireless link between the third and the fourth platoon vehicles for Scenario-5 (800 vehicles with DCC, see Fig. 8a)

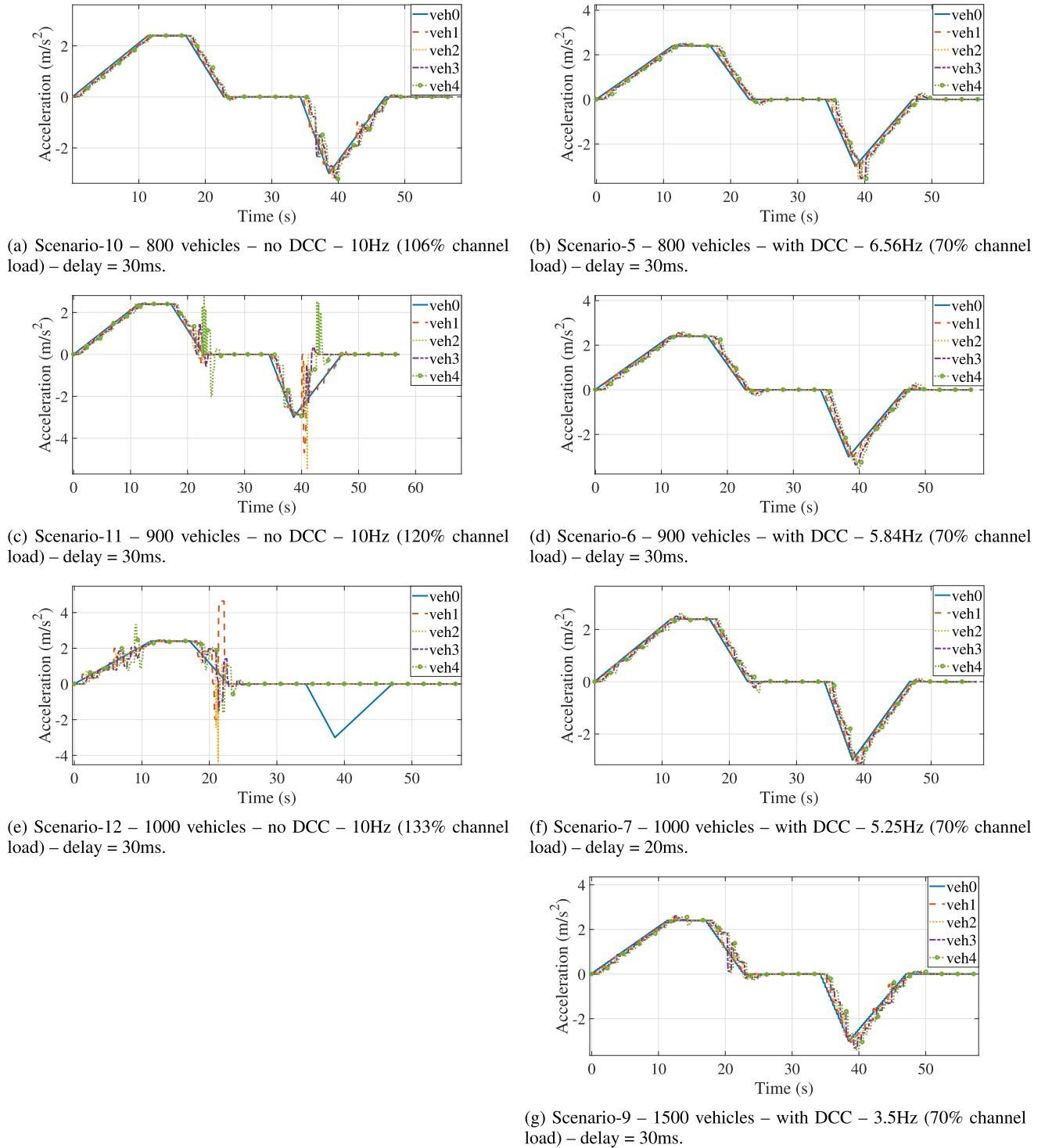


FIGURE 9. Performance of platoon vehicles for 800, 900, 1000, 1500 vehicles under headway time = 0.2s.

and Scenario-10 (800 vehicles without DCC, see Fig. 8b). In both scenarios, the road is occupied with 800 vehicles but Scenario-5 shows the case where the channel is 70% loaded. On the other hand, in Scenario-10, the channel is 106% loaded where the standard V2V message rate (10Hz) is selected. From Fig. 8a, we notice that the delay is mostly < 20ms for Scenario-5, whereas the delay in Scenario-10

increases as seen in Fig. 8b and it is mostly < 50ms. Packet losses can be seen also from Fig. 8 by noticing the red dots which represent loss of packets at certain samples. For the particular case shown in Fig. 8, PRR = 89.07% for Scenario-5 (Fig. 8a) and PRR = 76.88% for Scenario-10 (Fig. 8a). This can be visualized by noticing the increase in density of the red dots in Fig. 8b over Fig. 8a. The location of the vehicles

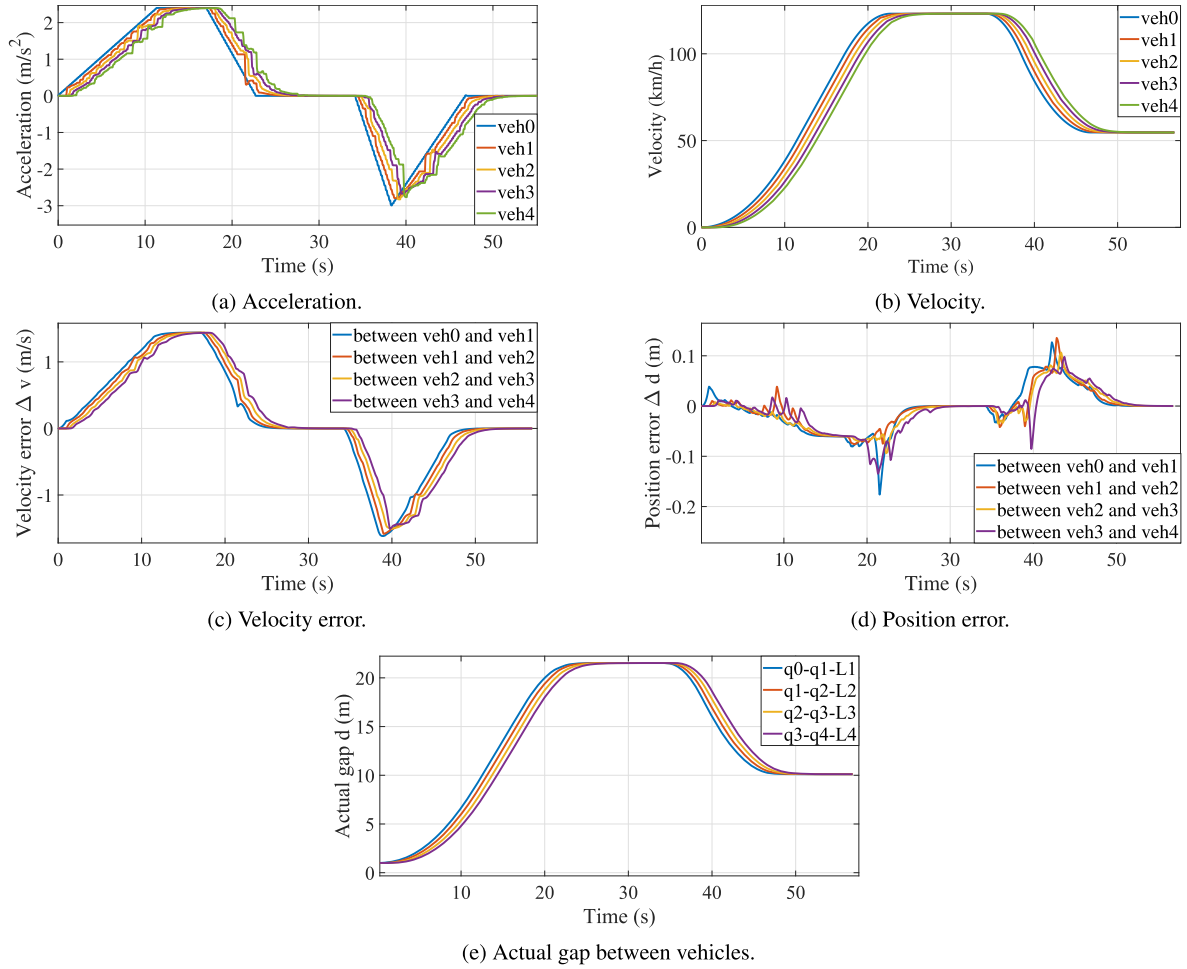


FIGURE 10. Performance of platoon vehicles for Scenario-7 (1000 vehicles). Delay $\tau = 20\text{ms}$, headway time $\tau_h = 0.6\text{s}$.

changes in SUMO in each simulation run, giving different patterns in each run.

4) CONTROL PERFORMANCE

In Fig. 9, we compare between the performance of the platoon vehicles in the DCC-active (right column) and DCC-inactive cases (left column) under the number of vehicles = 800, 900, 1000, 1500. We notice that the acceleration profiles of the platoon vehicles are similar for the low traffic congestion scenarios (800 vehicles) in the DCC-active and DCC-inactive cases, i.e Scenario-10 (106% channel load) and Scenario-5 (70% channel load), see Fig. 9a, 9b. On the other hand, tracking the acceleration of the platoon leader cannot be achieved and the formation of the platoon cannot be maintained as seen in Fig. 9c, 9e. In those experiments, the number of vehicles is increased (900 and 1000 vehicles) while the DCC is disengaged. The necessity of the DCC can be concluded by comparing Fig. 9c, 9e with Fig. 9d, 9f, respectively. The DCC is active in Fig. 9d, 9f under vehicular densities of 900 and 1000. It is clear that string stability and tracking capability are achieved if the DCC is engaged.

Fig. 9g shows also a nice tracking capability under very high traffic congestion (1500 vehicles). The figure where DCC is inactive corresponding to Scenario-9 (1500 vehicles)

shown in Fig. 9g cannot be obtained via simulations. This is due to the very low PRR values and long delays experienced between platoon vehicles. Therefore, platoon formation cannot be maintained using only V2V communication devices and vehicles should rely on their on-board sensors. This proves that a DCC is necessary to maintain string stability under high network and traffic congestion levels.

Fig. 10 and Fig. 11 show the tracking capability (velocity, acceleration, position error, and velocity error profiles) of the platoon vehicles for Scenario-7 (1000 vehicles) and Scenario-9 (1500 vehicles). The headway time $\tau_h = 0.6\text{s}$ and delay $\tau = 20\text{ms}$ are chosen for Scenario-7. On the other hand, a longer headway time ($\tau_h = 0.8\text{s}$) and delay ($\tau = 50\text{ms}$) are chosen for Scenario-9 to show that our scheme works for a wide range of parameters. From Fig. 10 and Fig. 11, it is observed that string stability is preserved and minimum position error is achieved. Other scenarios presented in Table 5 have similar tracking capabilities.

C. FUEL EFFICIENCY ANALYSIS

In this section, we analyse the fuel consumption of the platoon vehicles in DCC-active and DCC-inactive scenarios.

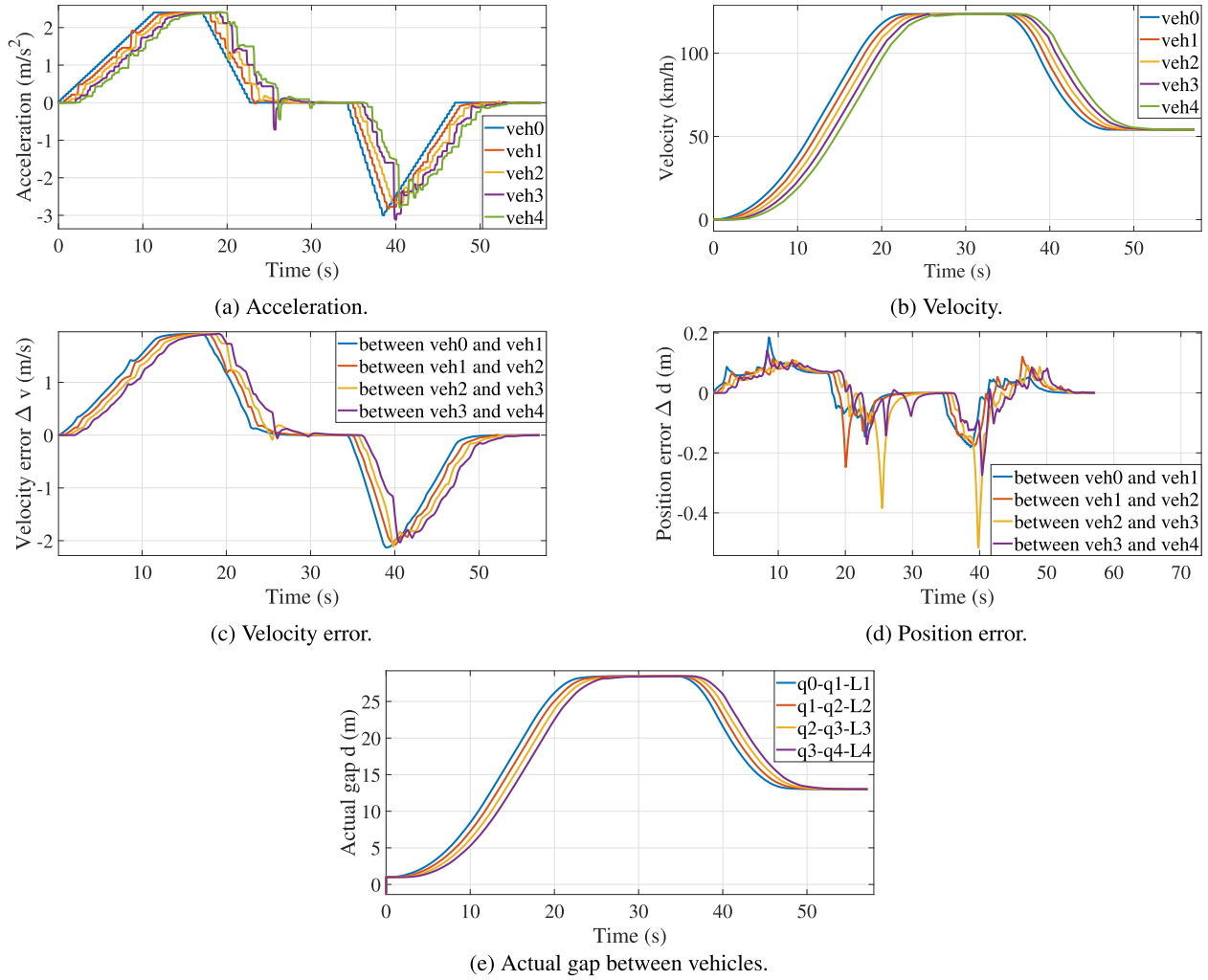


FIGURE 11. Performance of platoon vehicles for 1500 vehicles. Delay $\tau = 50\text{ms}$, headway time $\tau_h = 0.8\text{s}$.

1) FUEL CONSUMPTION MODEL

The fuel consumption model for vehicle i can be represented as follows [11]:

$$fuel^i = \frac{1}{H\varpi} \int_{t_0}^{t_f} \left[\varrho \mu m^i g v^i + 0.5 \rho A C (1 - \phi_{drag}^i) (v^i)^3 + m^i a^i v^i \right] dt, \quad (47)$$

where,

$$\varrho = \begin{cases} 1 & \text{if } \mu m^i g v^i + 0.5 \rho A C (1 - \phi_{drag}^i) (v^i)^3 + m^i a^i v^i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Using the above model, the instantaneously consumed fuel can be computed by considering the changing acceleration and velocity. ϕ_{drag}^i is the air-drag reduction for vehicle i due to the proximity of other vehicles. It is noted that the percentage of air-drag reduction is directly related to the inter-vehicle gap [11], [24]. On the other hand, the aerodynamic drag force is inversely related to the percentage of air-drag reduction.

TABLE 6. Parameters for the power function $\phi_{drag}^i = \mathfrak{C} - \mathfrak{A}(d^i)^{\mathfrak{B}}$ obtained using curve fitting tools for the curves in Fig. 12.

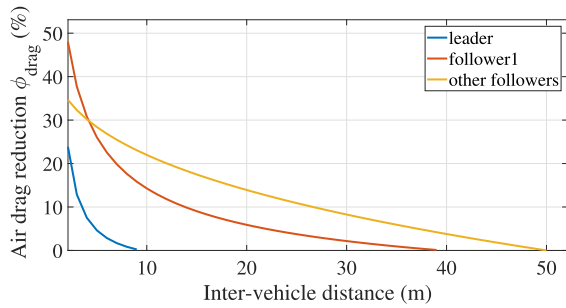
	\mathfrak{A}	\mathfrak{B}	\mathfrak{C}	\mathfrak{D}
leader	-0.7538	-1.407	-0.032	10
first follower	-0.9004	-0.4909	-0.148	40
other followers	0.1487	0.3278	0.5361	50

Therefore, a lower inter-vehicle gap leads to a higher percentage of air-drag reduction and a lower aerodynamic drag force (and fuel consumption). ϕ_{drag}^i can be obtained for passenger cars from Fig. 12 for the leading vehicle (blue curve), first follower (red curve) and other followers (yellow curve). It is noted from Fig. 12 that the air-drag reduction ϕ_{drag}^i is inversely related to the inter-vehicle distance d^i . Curves in Fig. 12 are fitted to the data points taken from [24]. We used curve fitting tools to approximate those data points into a power function of the following form,

$$\phi_{drag}^i = \begin{cases} \mathfrak{C} - \mathfrak{A}(d^i)^{\mathfrak{B}} & 0 \leq d^i \leq \mathfrak{D} \\ 0 & d^i > \mathfrak{D} \end{cases} \quad (48)$$

TABLE 7. Consumed fuel in litre for the platoon vehicles for the scenarios with 600 & 800 vehicles with/without DCC for headway 0.2s, 0.4s, 0.6s, 1s.

	τ_h		Consumed fuel $\times 10^{-2}$				
			veh0	veh1	veh2	veh3	veh4
600 vehicles	0.2s	DCC	7.0639	6.1153	5.8363	5.8407	5.8422
		no DCC	7.1256	6.1673	5.8848	5.8833	5.8851
	0.4s	DCC	7.0639	6.5448	6.0918	6.0772	6.0654
		no DCC	7.1256	6.5998	6.1466	6.1300	6.1124
	0.6s	DCC	7.0639	6.7314	6.2484	6.2174	6.1900
		no DCC	7.1256	6.7984	6.3073	6.2770	6.2468
	1s	DCC	7.0639	6.9343	6.4586	6.4054	6.3521
		no DCC	7.1256	7.0007	6.5259	6.4773	6.4322
800 vehicles	0.2s	DCC	6.9275	6.0031	5.7321	5.7409	5.7518
		no DCC	7.1256	6.1811	5.8896	5.8924	5.9065
	0.4s	DCC	6.9275	6.4152	5.9824	5.9701	5.9587
		no DCC	7.1256	6.5988	6.1430	6.1322	6.1121
	0.6s	DCC	6.9275	6.6092	6.1234	6.0894	6.0621
		no DCC	7.1256	6.8011	6.3057	6.2612	6.2371
	1s	DCC	6.9275	6.7876	6.3101	6.2431	6.1842
		no DCC	7.1256	6.9857	6.5119	6.4590	6.3961

**FIGURE 12.** Air drag reduction ϕ_{drag}^i .**TABLE 8.** Normalized fuel savings.

	τ_h	Normalized fuel savings (%)				
		veh0	veh1	veh2	veh3	veh4
600 vehicles	0.2s	0.8657	0.8440	0.8251	0.7241	0.7288
	0.4s	0.8657	0.8329	0.8908	0.8610	0.7678
	0.6s	0.8657	0.9866	0.9326	0.9490	0.9105
	1s	0.8657	0.9493	1.0307	1.1092	1.2455
800 vehicles	0.2s	2.7802	2.8809	2.6750	2.5706	2.6190
	0.4s	2.7802	2.7828	2.6135	2.6439	2.5098
	0.6s	2.7802	2.8214	2.8906	2.7435	2.8055
	1s	2.7802	2.8348	3.0996	3.3433	3.3122

The parameters \mathfrak{A} , \mathfrak{B} , \mathfrak{C} are obtained using curve fitting tool **cfTool** from Matlab. The parameters \mathfrak{A} , \mathfrak{B} , \mathfrak{C} , \mathfrak{D} are shown in Table 6.

The fuel consumption model of Eq. 47 uses, besides the ϕ_{drag}^i values of Table 6, the following parameters: $m^i = 1700\text{kg}$, $\rho = 1.29\text{kg/m}^3$, $A = 3.12\text{m}^2$, $C = 0.367$, $g = 9.8\text{m/s}^2$, $\varpi = 0.4$, $H = 36\text{MJ/L}$. q is 1 if the engine is active and zero otherwise. ϖ is the average engine efficiency. H is calorific power of the fuel.

2) CONSUMED FUEL

The fuel consumption of the platoon vehicles for the DCC-active and DCC-inactive cases are shown in Table 7. The fuel savings of the platoon vehicles of the DCC-active cases are shown in Table 8 which are normalized with respect to the DCC-inactive cases. We notice that DCC-active cases outperform the DCC-inactive cases in terms of fuel savings, which reaches 3% of savings in case of 800 vehicles. For > 800 vehicles the savings are expected to be more

than 3% but it is hard to stabilize the system in case of DCC-inactive scenarios.

IX. CONCLUSION

In this paper, we presented a multi-rate multi-layered vehicle-platoon model and control under high levels of network and traffic congestion taking communication delay into account. We analysed the performance with and without considering communication delay and we showed that safety and string stability are compromised if delay is not considered in the platoon modeling and control design. Moreover, we evaluated and validated the control scheme with extensive simulation of network and traffic behavior. It can be concluded that the platooning system cannot operate under high congestion without message-rate congestion control (DCC). Therefore, DCC is an enabling technology that directly affects the performance of the platoon in high congestion situations. The simulations moreover show a 3% fuel savings if a DCC is used. In future work, we plan to extend the approach to consider time-varying delay to study platoon control with heterogeneous communication rates, and to study the stability of MPC under fast switching.

REFERENCES

- [1] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Veh. Commun.*, vol. 2, no. 2, pp. 110–123, Apr. 2015.
- [2] T. Abbas, K. Sjöberg, J. Karedal, and F. Tufvesson, "A measurement based shadow fading model for vehicle-to-vehicle network simulations," *Int. J. Antennas Propag.*, vol. 2015, pp. 1–12, May 2015.
- [3] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "EMBARC: Error model based adaptive rate control for vehicle-to-vehicle communications," in *Proc. 10th ACM Int. Workshop Veh. Inter-Netw., Syst., Appl. (VANET)*, Jun. 2013, pp. 41–50.
- [4] M. S. Branicky, S. M. Phillips, and W. Zhang, "Stability of networked control systems: Explicit analysis of delay," in *Proc. Amer. Control Conf. (ACC)*, vol. 4, Jun. 2000, pp. 2352–2357.
- [5] F. Borrelli, A. Bemporad, and U. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, Jun. 2017.
- [6] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of platooning on public motorways," in *Proc. 17th World Congr. Intell. Transp. Syst.*, Oct. 2010, pp. 1–12.
- [7] A. Chandramohan and G. Heijenk, "Modelling activation of congestion control for estimating channel load in vehicular networks," in *Proc. Wireless Days (WD)*, Jun. 2021, pp. 1–8.

- [8] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1575–1580, Jul. 2009.
- [9] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct. 2007.
- [10] M. di Bernardo, A. Salvi, and S. Santini, "Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 102–112, Feb. 2015.
- [11] Q. Deng, "A general simulation framework for modeling and analysis of heavy-duty vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3252–3262, Nov. 2016.
- [12] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, document ETSI EN 302 637-2, 2019.
- [13] *Intelligent Transport Systems (ITS); Users and Applications Requirements; Part 2: Applications and Facilities Layer Common Data Dictionary*, document ETSI TS 102 894-2, 2014.
- [14] *Intelligent Transport Systems (ITS); V2X Applications; Part 1: Road Hazard Signalling (RHS) Application Requirements Specification*, document ETSI TS 101 539-1, 2013.
- [15] *Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for Operation in the ITS G5A and ITS G5B Medium; Report on Cross layer DCC Algorithms and Performance Evaluation*, document ETSI TR 101 612, 2014.
- [16] *Intelligent Transport Systems (ITS); Cross Layer DCC Management Entity for Operation in the ITS G5A and ITS G5B Medium; Validation Set-Up and Results*, document ETSI TR 101 613, 2015.
- [17] *Intelligent Transport Systems (ITS); On the Recommended Parameter Settings for Using STDMA for Cooperative ITS; Access Layer Part*, document ETSI TR 102 861, 2012.
- [18] *Intelligent Transport Systems (ITS); Access Layer Specification for Intelligent Transport Systems Operating in the 5 GHz Frequency Band*, document ETSI EN 302 663, 2019.
- [19] G. Fiengo, D. G. Lui, A. Petrillo, S. Santini, and M. Tufo, "Distributed robust PID control for leader tracking in uncertain connected ground vehicles with V2V communication delay," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 3, pp. 1153–1165, Jun. 2019.
- [20] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intell. Transp. Syst.*, vol. 10, no. 7, pp. 503–513, Sep. 2016.
- [21] J. Gong, Y. Zhao, and Z. Lu, "Sampled-data vehicular platoon control with communication delay," *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 232, no. 1, pp. 39–49, Jan. 2018.
- [22] D. Goswami, A. Masrur, R. Schneider, C. J. Xue, and S. Chakraborty, "Multirate controller design for resource- and schedule-constrained automotive ECUs," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2013, pp. 1123–1126.
- [23] T. R. Gonçalves, V. S. Varma, and S. E. Elayoubi, "Vehicle platooning schemes considering V2V communications: A joint communication/control approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, May 2020, pp. 1–6.
- [24] A. A. Hussein and H. A. Rakha, "Vehicle platooning impact on drag coefficients and energy/fuel saving implications," Jan. 2020, *arXiv:2001.00560*.
- [25] A. Ibrahim, D. Goswami, H. Li, I. M. Soroa, and T. Basten, "Multi-layer multi-rate model predictive control for vehicle platooning under IEEE 802.11p," *Transp. Res. C, Emerg. Technol.*, vol. 124, Mar. 2021, Art. no. 102905, doi: [10.1016/j.trc.2020.102905](https://doi.org/10.1016/j.trc.2020.102905).
- [26] A. Ibrahim, C. B. Math, D. Goswami, T. Basten, and H. Li, "Co-simulation framework for control, communication and traffic for vehicle platoons," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2018, pp. 352–356.
- [27] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012 (Revision of IEEE Standard 802.11-2007), Mar. 2012, pp. 1–2793, doi: [10.1109/IEEEESTD.2012.6178212](https://doi.org/10.1109/IEEEESTD.2012.6178212).
- [28] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standard 802.11p-2010 (Amendment to IEEE Standard 802.11-2007), IEEE Standard 802.11k-2008, IEEE Standard 802.11r-2008, IEEE Standard 802.11y-2008, IEEE Standard 802.11n-2009, and IEEE Standard 802.11w-2009), Jul. 2010, pp. 1–51, doi: [10.1109/IEEEESTD.2010.5514475](https://doi.org/10.1109/IEEEESTD.2010.5514475).
- [29] D. Jiang and L. Delgrossi, "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, May 2008, pp. 2036–2040.
- [30] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and inter-vehicle communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 155–161, Sep. 2002.
- [31] R. Kianfar, P. Falcone, and J. Fredriksson, "A control matching model predictive control approach to string stable vehicle platooning," *Control Eng. Pract.*, vol. 45, pp. 163–173, Dec. 2015.
- [32] J. B. Kenney, G. Bansal, and C. E. Rohrs, "LIMERIC: A linear message rate control algorithm for vehicular DSRC systems," in *Proc. 8th ACM Int. Workshop Veh. Inter-Netw. (VANET)*, Sep. 2011, pp. 21–30.
- [33] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2804–2821, Jul. 2018.
- [34] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, Dec. 1976.
- [35] W. Y. Lin, M. W. Li, K. Lan, and C. H. Hsu, "A comparison of 802.11 a and 802.11 p for V-to-I communication: A measurement study," in *Proc. Int. Conf. Heterogeneous Netw. Qual., Rel., Secur. Robustness*. Berlin, Germany: Springer, Nov. 2010, pp. 559–570.
- [36] R. Lu, Y. Xu, and R. Zhang, "A new design of model predictive tracking control for networked control system under random packet loss and uncertainties," *IEEE Trans. Ind. Electron.*, vol. 63, no. 11, pp. 6999–7007, Nov. 2016.
- [37] D. Limon, T. Alamo, F. Salas, and E. F. Camacho, "On the stability of constrained MPC without terminal constraint," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 832–836, May 2006.
- [38] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Aug. 2001, pp. 625–630.
- [39] W. Liu and Z. Li, "Comprehensive predictive control method for automated vehicles with delays," *IEEE Access*, vol. 7, pp. 81923–81933, 2019.
- [40] J. Lan and D. Zhao, "Min-max model predictive vehicle platooning with communication delay," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12570–12584, Nov. 2020.
- [41] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2004, pp. 284–289.
- [42] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, Feb. 2009.
- [43] J. M. Maciejowski, *Predictive Control: With Constraints*. London, U.K.: Pearson, 2002.
- [44] R. Makowitz and C. Temple, "FlexRay—A communication network for automotive control systems," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, Jun. 2006, pp. 207–212.
- [45] G. Marsden, M. McDonald, and M. Brackstone, "Towards an understanding of adaptive cruise control," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 1, pp. 33–51, Feb. 2001.
- [46] H. N. Mahjoub, A. Tahmasbi-Sarvestani, S. M. O. Gani, and Y. P. Fallah, "Composite $\alpha - \mu$ based DSRC channel model using large data set of RSSI measurements," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 205–217, Mar. 2018.
- [47] G. Naus, R. Vugts, J. Ploeg, D. M. M. Van, and M. Steinbuch, "Towards on-the-road implementation of cooperative adaptive cruise control," in *Proc. 16th World Congr. Exhib. Intell. Transp. Syst. Serv.*, Sep. 2009, pp. 2567–2578.
- [48] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain approach," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4268–4279, Nov. 2010.
- [49] "OSEK/VDX operating system," OSEK Group, Osek, Czech Republic, Tech. Rep., Feb. 2005. [Online]. Available: <https://www.osek-idx.org/mirror/os223.pdf>

- [50] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, “Cooperative adaptive cruise control: Network-aware analysis of string stability,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1527–1537, Aug. 2014.
- [51] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, “Design and experimental evaluation of cooperative adaptive cruise control,” in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 260–265.
- [52] W. B. Qin, M. M. Gomez, and G. Orosz, “Stability and frequency response under stochastic communication delays with applications to connected cruise control design,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 388–403, Feb. 2017.
- [53] M. Sepulcre, M. Gonzalez-Martin, J. Gozalvez, R. Molina-Masegosa, and B. Coll-Perales, “Analytical models of the performance of IEEE 802.11 p vehicle to vehicle communications,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 713–724, Jan. 2021.
- [54] D. V. Swaroop, “String stability of interconnected systems: An application to platooning in automated highway systems,” Ph.D. dissertation, Univ. California, Berkeley, Berkeley, CA, USA, 1997.
- [55] A. Tiganasu, C. Lazar, and C. F. Caruntu, “Cyber physical systems-oriented design of cooperative control for vehicle platooning,” in *Proc. 21st Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2017, pp. 465–470.
- [56] M. Tsujii, H. Takeuchi, K. Oda, and M. Ohba, “Application of self-tuning to automotive cruise control,” in *Proc. Amer. Control Conf.*, May 1990, pp. 1843–1848.
- [57] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, “Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3684–3703, Sep. 2009.
- [58] A. G. Ulsoy, H. Peng, and M. Çakmakci, *Automotive Control Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [59] B. van Arem, C. J. G. van Driel, and R. Visser, “The impact of cooperative adaptive cruise control on traffic-flow characteristics,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 429–436, Dec. 2006.
- [60] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. van de Wouw, “Robust model predictive cooperative adaptive cruise control subject to V2V impairments,” in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.
- [61] W. Viriyasitavat, M. Boban, H.-M. Tsai, and A. Vasilakos, “Vehicular communications: Survey and challenges of channel and propagation models,” *IEEE Veh. Technol. Mag.*, vol. 10, no. 2, pp. 55–66, Jun. 2015.
- [62] J. Zhang and P. Ioannou, “Longitudinal control of heavy trucks in mixed traffic: Environmental and fuel economy considerations,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 92–104, Mar. 2006.
- [63] T. Zeng, O. Semiari, W. Saad, and M. Bennis, “Joint communication and control for wireless autonomous vehicular platoon systems,” *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7907–7922, Nov. 2019.
- [64] H. Zhou, R. Saigal, F. Dion, and L. Yang, “Vehicle platoon control in high-latency wireless communications environment: Model predictive control method,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2324, no. 1, pp. 81–90, Jan. 2012.



AMR IBRAHIM received the bachelor's degree in computer engineering and systems from Mansoura University, Egypt, in 2010, the Diploma degree in mechatronics from the Information Technology Institute, Egypt, in 2011, the master's degree in mathematical engineering, and the joint M.Sc. degree in mathematical modeling in life sciences from L'Aquila University, Italy, and Hamburg University, Germany, in 2016. From 2011 to 2014, he was appointed as a Teaching Assistant with Mansoura University, during this period, he participated in teaching different undergraduate courses and he finished his coursework towards his master's degree in mathematical engineering. In 2014, he joined the International Erasmus Mundus Program “Mathmods.” He is currently a Researcher with the Electronic Systems Group, Department of Electrical Engineering, Eindhoven University of Technology (TU/e). His research interests include model predictive control, intelligent transport systems, and networked control systems.



DIP GOSWAMI (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the National University of Singapore (NUS), in 2009. From 2010 to 2012, he was an Alexander von Humboldt Postdoctoral Fellow at TU Munich, Germany. He is currently an Assistant Professor with the Electronic Systems Group, Department of Electrical Engineering, Eindhoven University of Technology (TU/e). His research interests include various design aspects of embedded control systems in resource-constrained domains, such as automotive and robotics.



HONG LI received the B.Sc. degree in electronics and the M.Sc. and Ph.D. degrees in dynamical ECG acquisition and analysis from Tsinghua University, Beijing, China, in 1988, 1992, and 1995, respectively. He joined the Department of Engineering Physics, Tsinghua University, as a Lecturer, in 1995. From 1996 to 1998, he did a postdoctoral research on image representation and recognition at the Delft University of Technology, The Netherlands. He joined Philips Research, Eindhoven, in 1999, worked as a Senior Scientist on personal infotainment content management systems, including digital rights management, privacy protection, and 3D video storage. Since 2006, he moved to NXP Research and worked on personal radio systems, audio content analysis, and software defined radio. He is currently a Senior Principal in RF Processing of NXP, with research interests covering signal processing, automotive digital radio, wireless communications, radar signal processing, intelligent transport systems, and autonomous driving.



TWAN BASTEN (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computing science from the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. He is currently a Professor with the Department of Electrical Engineering, TU/e. He is also a Senior Research Fellow with the ESI, TNO, Eindhoven. His current research interests include the design of embedded and cyber-physical systems, dependable computing, and computational models.

...