Mechatronics 74 (2021) 102512

Contents lists available at ScienceDirect

Mechatronics

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

# Connected vehicle driving safety enhancement via dynamic communication channel selection

Zejiang Wang [a], Yunhao Bai [b], Jingqiang Zha [a], Junmin Wang [a,*], Xiaorui Wang [b]

[a] Walker Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX 78712, United States
[b] Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, United States

ARTICLE INFO

*Keywords:*
Collision avoidance
Cooperative adaptive cruise control
Vehicle-to-vehicle communication

ABSTRACT

Vehicle-to-Vehicle communication can cogently improve traffic safety because it grants drivers better situational awareness and strengthens inter-cooperation among them. However, the current policy obliges the safety-critical messages, i.e., Basic Safety Messages (BSMs) and Event Safety Messages (ESMs), to be transmitted exclusively in one single channel over the Dedicated Short-Range Communication (DSRC) spectrum, which may incur severe channel congestions, intolerable communication delays, and higher accident probabilities. To alleviate the channel congestion, popular measures focus on adaptively adjusting the transmission parameters, e.g., packet generating rate. However, due to the narrow bandwidth of a single channel, these methods can hardly ensure timely delivery of critical safety-related messages when the density of DSRC-enabled vehicles becomes high. Instead of sticking to only one single channel, this paper applies a dynamic channel selection algorithm to thoroughly exploit the entire DSRC band resource and reduce the transmission delay. To demonstrate the effectiveness of the utilized algorithm, we conduct experiments under two representative scenarios, i.e., a cooperative adaptive cruise control scenario and a run-the-red-light scenario at an intersection. Experimental results show that the transmission delays of both the BSMs and the ESMs could be effectively reduced, yielding an improved vehicle platoon control accuracy, string stability, and collision avoidance performance.

## 1. Introduction

Vehicle-to-vehicle (V2V) communication recently attracts attention from both academia and industry as it has the potential to substantially reduce traffic-related casualties and improve commuting efficiencies. Typical applications of V2V communication include city intersection management [1, 2], vehicle lane-change and collision warnings [3, 4], and cooperative traffic congestion detection [5]. Particularly, integrating V2V communication into the Adaptive Cruise Control (ACC) system [6] leads to the Cooperative Adaptive Cruise Control (CACC) system [7], which can attenuate speed fluctuations along a vehicle platoon [8] and further reduce collision hazards. Moreover, V2V-based platoon emergency braking can be found in [9].

To increase the benefits of vehicular communications, the V2V networks need to be deployed on a large scale [10]. Nonetheless, as the penetration rate of the connected vehicles soars, the intrinsic vulnerabilities of vehicular networks [11], e.g., fragile fading channels, hidden-terminal issue, and drastic channel interferences will become more pronounced. In parallel, the Federal Communication Commission (FCC) in the United States obliges that only one unique channel—-principally the Common Control Channel (CCH)—among the seven in total available channels within the Dedicated Short-Range Communication (DSRC) spectrum is allowed for transmitting both the Basic Safety Messages (BSMs), e.g., vehicle dynamic states, and the Event Safety Messages (ESMs), e.g., warning flags [12]. The other six channels are reserved mainly for non-safety-related applications, such as advertisements or infotainments [13]. As revealed in [14], this restriction on channel utilization can entail severe congestion in the CCH, resulting in a huge transmission delay. Standard methods to mitigate channel congestion concentrate on adaptively adjusting transmission parameters [15], e.g., packet generating rate, transmission power, and carrier sense threshold. Even though these countermeasures can mitigate packet collision, the safety-related messages are howbeit constrained in a single channel whose network capacity will appear deficient if a considerable amount of messages need to be simultaneously exchanged.

To tackle this issue, multichannel operations [16] for disseminating

[*] Corresponding author.
*E-mail addresses:* wangzejiang@utexas.edu (Z. Wang), bai.228@osu.edu (Y. Bai), zhajingqiang@utexas.edu (J. Zha), jwang@austin.utexas.edu (J. Wang), wang.3596@osu.edu (X. Wang).

https://doi.org/10.1016/j.mechatronics.2021.102512
Received 18 May 2020; Received in revised form 16 December 2020; Accepted 31 January 2021
Available online 10 February 2021

the BSMs and the ESMs have also been proposed [17, 18]. However, these approaches typically used one or two *pre-selected* channels for safety message spreading. Besides, they did not consider the real-time communication requirements of the served V2V use case. In stark contrast, this paper employs a *dynamic* channel selection algorithm. A group of involved vehicles of a V2V application locally estimates the transmission delays and Packet Reception Rates (PRRs) of all the seven channels in the DSRC spectrum and then collaboratively decides the most appropriate channel for data transmission. The ultimately selected channel will try to satisfy both the delay and PRR requirements of the specific V2V application. To verify the algorithm, experiments under two representative scenarios were conducted with three scaled cars. Firstly, the capability of the algorithm in reducing the transmission delay of BSMs was verified in a CACC scenario, where the three scaled cars formed a platoon. It was demonstrated that both the inter-vehicle distance tracking performance and the string stability could be improved with the channel selection algorithm. Secondly, a run-the-red-light scenario at an intersection was designed to exhibit that the transmission delay of ESMs could also be effectively reduced with the help of the channel selection algorithm, which facilitated collision avoidance.

The rest of the paper is organized as follows. Section 2 explains the dynamic channel selection algorithm. Section 3 describes the experimental setup. Section 4 introduces the algebraic differentiation technique [19], which was used for deriving the longitudinal speed of each scaled car from the noisy longitudinal position data. Section 5 exhibits the experiment results of both the CACC scenario and the run-the-red-light scenario. Finally, Section 6 concludes this paper.

## 2. Dynamic communication channel selection

Embedded in each vehicle involved in a specific V2V application, e.g., collision avoidance or CACC, the dynamic channel selection algorithm makes all the participants jointly determine the most appropriate channel for timely dissemination of the safety-critical messages.

### 2.1. Overall framework

Each vehicle periodically estimates the communication delays and the packet reception rates (PRRs) of all the seven channels in the DSRC spectrum. Based on the estimation results, each car individually classifies the seven channels into three classes and constructs a continuously updated local Channel Preference List (CPL). Once being triggered by an event flag, these vehicles form a communication group, and the vehicle with the smallest media access control (MAC) address would serve as the coordinator and collect all the CPLs from other vehicles in the same group. According to the received CPLs, the coordinator determines the optimal channel and sends back this decision in an ACK frame to all the other group members. Once the other vehicles receive this ACK frame, they also change the channel to the optimal one and start to send/receive packets. The scheme of the channel selection algorithm can be found in [14]. There are two major components in this channel selection algorithm: 1) Periodical estimation of transmission delays and PRRs, and 2) Channel selection (upon an event flag).

### 2.2. Estimation of transmission delay and prr

The channel employed to relay either the BSMs or the ESMs should satisfy two constraints specified by the V2V application [20]: 1) a packet transmitted on this channel cannot experience a delay beyond the Maximal Allowable Delay (MAD); and 2) the probability of successfully transmitting a single packet within the Maximal Allowable Transfer Interval (MATI) must reach a minimum threshold. The mathematical expression of these two constraints can be summarized as:

$$\begin{cases} d \leq \Delta, \\ 1 - p^{L_m} \geq \delta, \end{cases} \tag{1}$$

where $d$ is the actual communication delay, $\Delta$ represents the maximal allowable delay threshold, $p$ is the packet error rate, $L_m$ exhibits the utmost retransmission times allowed within the MATI, and $\delta$ is the minimum requirement on the overall packet reception rate within MATI. In (1), $\Delta$,$L_m$, and $\delta$ are all determined by the specific V2V application. In contrast, the communication delay $d$ and the packet error rate $p$ must be estimated in real-time. According to the packet collision model in [21], packet collision will not occur if 1) there is no interfering node sending packet at the same time slot, and 2) there is no hidden terminal transmitting messages during the vulnerable period on the same channel. Therefore, the packet error rate $p$ can be calculated as:

$$p = 1 - e^{-N_c\tau} e^{-N_h\tau T_\nu/\sigma}, \tag{2}$$

where $N_c$ represents the number of interfering nodes in the considered channel, $N_h$ implies the number of hidden terminals, $T_\nu$, $\tau$, and $\sigma$ express individually the hidden terminal's vulnerability period, the total transmission probability of one node in a single time slot, and the duration of a time slot. In (2), $T_\nu$,$\tau$, and $\sigma$ can be calculated by following the standard procedures. Moreover, $N_c$ and $N_h$ can be inferred from the beacon messages in the considered channel.

Subsequently, the actual transmission delay $d$ in (1) can be decomposed into two parts, i.e., the queuing delay $d_q$ and the service delay$d_s$.The queuing delay $d_q$ can be directly deduced from the *M/G/1* queue model in [21] and the service delay $d_s$can be approximated as the mathematical expectation of the overall retransmission delay. By modeling the back-off procedure of the current DSRC protocol (*IEEE 802.11p*) as a Markov process and applying the standard binary exponential back-off algorithm, $d_s$is evaluated as:

$$d_s = \sum_{i=0}^{L_u} (1-p)p^i \frac{\min\left(2^i CW_0, CW_{\max}\right) E[X]}{2}. \tag{3}$$

In (3), $L_u$ represents the default threshold on the maximal retransmission count in the *IEEE 802.11p* protocol, $p$ indicates the packet error rate calculated in (2), $CW_0$ and $CW_{\max}$ demonstrate respectively the initial and the upper-bounded contention window size, and $E[X]$ exhibits the mathematical expectation of the time elapse until the back-off counter's first decrement occurs. $E[X]$ can be deduced according to the Markov model of the back-off mechanism.

According to (2), (3), together with the *M/G/1* queue model, each vehicle can locally estimate both the transmission delay $d$, and the global packet reception rate within the MATI $1 - p^{L_m}$,of all the seven channels in the DSRC spectrum. From the estimation results, each car can individually categorize the seven channels in the DSRC band into three classes by verifying the transmission constraints in (1):

*Class 1*: the channel naturally satisfies both the transmission delay and the PRR constraints.

*Class 2*: the channel cannot simultaneously satisfy both the transmission delay and the PRR constraints until the non-safety-related messages from the interfering vehicles within the one-hop distance are temporarily suspended.

*Class 3*: the channel can only simultaneously satisfy both the transmission delay and the PRR constraints until the non-safety-related messages transmitted by 1) the interfering nodes within the one-hop distance and 2) the hidden terminals within the two-hop distance are temporarily suspended.

After the channel categorization, each car constructs its local Channel Preference List (CPL) by accrediting the highest preference to the channels within the *Class 1* and the lowest preference to the channels within the *Class 3*. As the estimations are effectuated periodically, each vehicle maintains the most freshly updated local CPL.

### 2.3. Channel selection

Once triggered by an event flag, the vehicle with the smallest MAC address is selected as the coordinator and the other vehicles dispatch their CPLs to the coordinator through the CCH. To reduce the additional delay caused by the channel selection procedure, the CPL is assigned with the highest priority by placing it at the head of the MAC queue. Once the coordinator receives all the CPLs, it will then compare the received CPLs with its own CPL and determine the target channel for safety-related messages dissemination via the following guidelines:

a If there exist at least one common *Class 1* channels in all the CPLs, the *Class 1* channel in the coordinator's CPL enduring the least interfering nodes (with the smallest $N_c$) will be selected;
b Or, if there merely exist shared *Class 2* channels in the coordinator's CPL and the received CPLs, the suboptimal *Class 2* channel in the coordinator's CPL that requires the least effort to temporarily suspend the non-safety-related messages from the interfering vehicles within the one-hop distance will be used;
c Very rarely, there does not exist either a common *Class 1* or a common *Class 2* channel in the coordinator's CPL and the received CPLs. Under this circumstance, the suboptimal *Class 3* channel in the coordinator's CPL that requires the least effort to momentarily suppress the non-safety-related messages from 1) the interfering nodes within the one-hop distance and 2) the hidden terminals within the two-hop distance will be finally chosen.

The decision of the coordinator would be sent back to other group members in an ACK frame, and others veer to this selected channel.

Note that this finally selected channel may not always be a *Class 1* channel for a specific vehicle in the group. If the selected channel is in *Class 2* with respect to a certain vehicle, the vehicle would then broadcast the transmission probability packets to all the interfering nodes within its one-hop distance. These transmission probability packets trigger the *P-persistence* mechanism of the interfering nodes: when a non-safety-related packet is going to be sent by one interfering node through the MAC layer, it only has a probability of *P* to be delivered to the PHY layer. Otherwise, such a non-safety-related packet will go through another round of back-off delay. In this way, the interference of this *Class 2* channel can be reduced to enforce the MATI/MAD requirements. Likewise, if the selected channel is in *Class 3* with respect to a certain group member, the non-safety-related messages from hidden terminals within its two-hop distance need to be temporarily suspended, as well.

### 2.4. Discussion

The dynamic channel selection algorithm is event-triggered and the activating condition is specified by the served V2V application. For a CACC use case, the algorithm is activated when the vehicle platoon is formed. For a V2V based collision-avoidance system, the algorithm will be triggered when the inter-vehicle distance goes below a predefined threshold.

The extra time-lag owing to the dynamic channel selection algorithm itself consists of four parts: the time for collecting CPLs, the generation overhead and the transmission delay of the coordinator's ACK, plus the channel switching overhead. Interfering severity plays a major role in the transmission delay [14] and the calculation overhead is hardware-related. The extra time lag of the dynamic channel selection algorithm was less than 0.02 s during our experiments.

## 3. Experiment hardware setup

To verify the effectiveness of the dynamic channel selection algorithm, experiments were conducted with three scaled cars, numbered as car1, car2, and car3. Please note that due to the space constraint of the test field, involving more scaled cars in the experiments would be difficult. Simulation results of the dynamic channel selection algorithm in large-scale traffic scenarios can be found in [22].

The 1:18 scaled car was front-steered and four-wheel-drive (4WD) via respectively a DC-servo and a brushed DC-motor. A Raspberry Pi (RPi) control board in the middle of the chassis served as the electronic control unit. The control board sent the throttle and the steering commands, in the form of Pulse Width Modulation (PWM) signals, to the motor and the servo. Moreover, the wireless chip of RPi acted as a virtual comprehensive sensor: to be cost-efficient, no expensive sensor, e.g., high-precision Inertial Measurement Unit (IMU) or laser scanner was mounted on the scaled cars. Instead, all the raw kinematics data, i.e., longitudinal position $X_i(t)$, lateral position $Y_i(t)$, and yaw angle $\psi_i(t)$,$i = 1, 2, 3$, collected by the OptiTrack camera-based indoor GPS, were wirelessly fed back to each corresponding scaled car. Finally, the crucial V2V functionality was implemented via TMOTE modules. TMOTE is a wireless sensor module commonly used for reliable and fast data dissemination in developing Wireless Sensor Networks (WSN) [23]. Onboard each scaled car, a TMOTE module communicated with the control board via the serial port for data exchange. Extra TMOTE modules were placed around, acting as the interfering nodes. The overall hardware setup is shown in Fig. 1.

The control framework of each scaled car is depicted in Fig. 2. The indoor GPS captured and wirelessly fed back the global coordinates $X_i(t)$,$Y_i(t)$, $\psi_i(t)$ to each scaled car in real-time. Based on the lateral coordinate $Y_i(t)$ and the yaw angle $\psi_i(t)$, a steering controller commanded the DC servo to guide the scaled car follow a referential path. At the same time, a speed planner determined the desired forwarding velocity, which was calculated based on the specific V2V application. According to the discrepancy between the actual forwarding speed and its desired value, a sliding-mode speed controller commanded the DC motor to eliminate the speed tracking error. To this end, the actual forwarding velocity $\widehat{\dot{X}}_i(t)$ must be first estimated from the noisy longitudinal position data $X_i(t)$.This task was realized via the algebraic differentiation estimation, which will be illustrated in Section 4.

## 4. Algebraic differentiation estimation

Deducing the longitudinal velocity from the noisy longitudinal position data in real-time can be challenging. It is well known that directly computing derivatives of a noisy signal is ill-posed [24]. In this paper, the algebraic differentiation estimator was adopted to derive $\widehat{\dot{X}}_i(t)$ in real-time.

### 4.1. Algebraic differentiation approach

Rooted in the operational calculus and the differential algebra, the Algebraic Differentiation Estimation (ADE) can explicitly express an

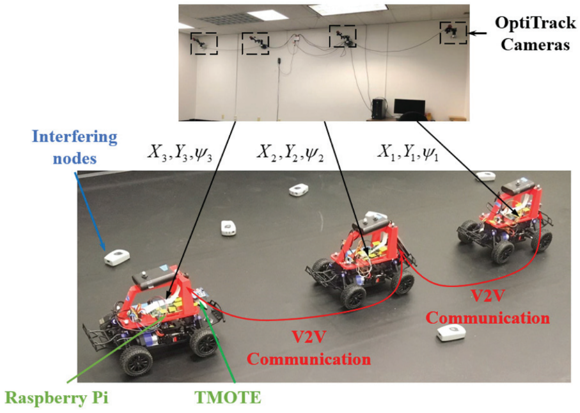


**Fig. 1.** CACC hardware setup.

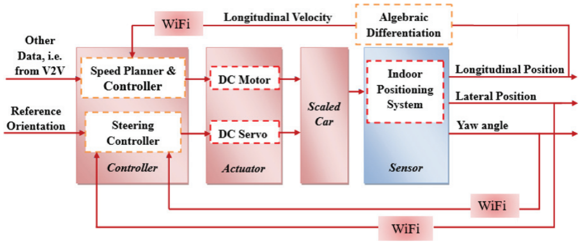


**Fig. 2.** Control framework of the scaled car.

arbitrary-order *derivative* of a smooth signal as its weighted *integral* [19, 24]. To determine the $\nu -$ *th*order derivative, the original signal will firstly be approximated as its truncated Taylor series with order $N \geq \nu$ and then be transformed into the operational domain via the Laplace transform. Therein, the approximated signal goes through pure algebraic manipulations to isolate the coefficient of the $\nu$ $-th$ order term of the truncated Taylor series. Finally, the $\nu - th$ order derivative of the primitive signal can be obtained by transforming the corresponding coefficient of the Taylor series back into the time domain.

When $N = \nu = 1$, the estimation of the first-order derivative of a noisy signal $y(t)$ under a sliding window framework can be expressed as [25]:

$$\hat{\dot{y}}(t) = \frac{6}{T^3}\int (T - 2\tau)y(t - \tau)d\tau, \tag{4}$$

with $T$as the sliding window width.

Specifically for our experiments, the noisy signal $y(t)$ represented the longitudinal position $X(t)$.

### 4.2. Implementation and representative experiment result

To implement (4), we first initiated a data buffer with $m$ +1 zero elements. At each sampling step $k$, the most recent raw data $y(k)$ was retrieved from the indoor GPS and inserted at the head of the data buffer (with index *0*). This action then shifted all the other raw data one-step to the right and the stalest one at the end of the data buffer (with index $m$) would be popped out. This insert-pop action created a First-In-First-Out (FIFO) sliding queue. Assigning the sampling frequency as $f_s$, the sliding window width in (4) becomes $T = m/f_s$. To reduce both the numerical integral error and the estimation delay, a high sampling frequency is preferred [19].

The ADE was validated through a preliminary experiment on a single scaled car. The car was controlled to follow a curved path and its longitudinal position $X(t)$, lateral position $Y(t)$, and yaw angle $\psi(t)$ were collected by the indoor GPS. Sampling frequency was fixed as $f_s = 500Hz$ and the sliding window size $T$ was respectively tuned as 0.04 s, 0.02 s, 0.02 s for $X(t), Y(t), \psi(t)$.The first-order derivative estimations $\hat{\dot{X}}_i(t)$, $\hat{\dot{Y}}(t)$, and $\hat{\dot{\psi}}(t)$ are depicted in Fig. 3.

Therefore, the algebraically estimated derivatives (red solid lines) fitted well with the ground truths (blue dashed lines), which were obtained by directly differentiating the raw data. Since the overall control loop of the scaled car could not run as fast as 500 Hz, the ADE was effectuated inside a PC, and the estimated longitudinal speed $\hat{\dot{X}}_i(t)$ was wirelessly sent to the scaled car's control board.

## 5. Experimental verification with scaled cars

The dynamic channel selection algorithm was verified under both a CACC scenario and a run-the-red-light-scenario. The CACC scenario was designed to show that the transmission timeliness of BSMs as well as the control performance and the string stability of a vehicle platoon can be improved with the dynamic channel selection algorithm. Later, the run-the-red-light scenario demonstrated that the ESMs can be addressed to the target vehicle within a shorter period through the dynamically selected channel, which can enhance the chance of collision avoidance at an intersection.

### 5.1. Cooperative adaptive cruise control

#### 5.1.1. Scenario description

In this scenario, the three-car platoon went along a straight line. The speed planner of the leading car generated the desired speed $v_{x1}^r$ for car1 as depicted in Fig. 4 to produce a series of speed fluctuations.

In the meantime, both the middle car (car2) and the last car (car3) adjusted their longitudinal velocities trying to maintain the referential inter-vehicle distances. As illustrated in Fig. 1, the leading car's BSMs were sent to the middle car, and the middle car's BSMs were sent to the last car. CACC under this communication topology is commonly referred to as 'semi-autonomous' ACC. In contrast to communicating with multiple cars in front, semi-autonomous ACC is easier to be implemented [26]. Besides, the middle car and the last car also had access to the longitudinal position and velocity of their immediate leader from OptiTrack. They utilized this supplementary information to decide the inter-vehicle distance as well as the velocity discrepancy in real-time.

The BSMs from car1 to car2 and from car2 to car3 went through one of the three usable channels: CH22, CH24, and CH26 inside TMOTE. Fifteen extra TMOTE modules were unevenly distributed in the three channels as the interfering nodes. An interfering node periodically sent out garbage packets, which occupied one of the three available channels. The number of interferences in each channel was time-varying.

Two groups of experiments were effectuated, with either the dynamically selected channel or the fixed CH26 (mimicking the CCH in the DSRC spectrum) utilized for BSMs transmissions. For simplicity, the channel selection algorithm remained active if enabled.

#### 5.1.2. Distance tracking error dynamics

Even though three cars formed the platoon, the distance tracking error dynamics can be sufficiently expressed with a leader-follower system, as depicted in Fig. 5.

In **Fig. 5**, $x_i$ and $x_{i-1}$ represent individually the longitudinal positions of the rear bumps of the follower and the leader. $v_i$ and $v_{i-1}$ indicate their corresponding longitudinal velocities. Besides, $L_i$ shows the length of the follower and $D_i$ is the actual inter-vehicle distance.

There exist various spacing policies e.g., [27, 28] to determine the

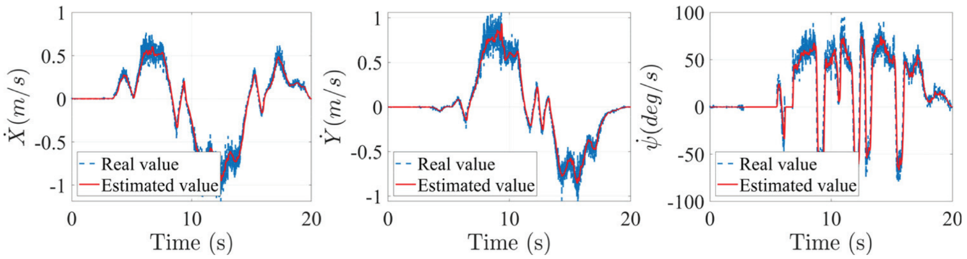


**Fig. 3.** First-order derivative estimations from ADE.

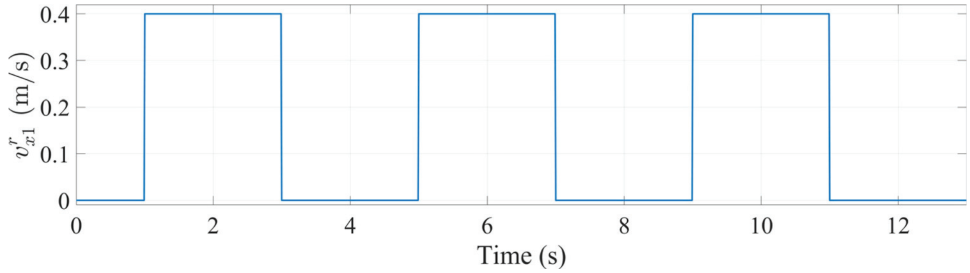


**Fig. 4.** The desired speed of CACC leading car1.

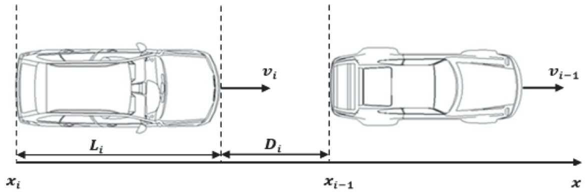


**Fig. 5.** Leader-follower vehicle platoon.

referential inter-vehicle distance $D_i^*$. Here, the constant-time spacing policy was adopted to fix $D_i^*$ as:

$$D_i^* = r_i + h_i v_i, \tag{5}$$

with $r_i$ as the standstill distance and $h_i$ the constant time headway. Therefore, the distance tracking error $e_i$ can be expressed as:

$$e_i = x_{i-1} - x_i - L_i - r_i - h_i v_i. \tag{6}$$

Then, a first-order system was utilized to approximate the longitudinal dynamics of the follower and the leader, as:

$$\begin{cases} v_i = u_i/(1+\tau_i s), \\ v_{i-1} = u_{i-1}/(1+\tau_{i-1} s). \end{cases} \tag{7}$$

In (7), $\tau_i, \tau_{i-1}$ are respectively the time constants of the follower and the leader, $u_i, u_{i-1}$ express their *desired* longitudinal velocities, and $s$ is the Laplace variable. For the three-car platoon, $u_1$ (the desired speed of the leading car) was specified in Fig. 4 whereas $u_2, u_3$ were determined by a CACC speed control law.

Based on (6) and (7) and assuming zero initial conditions, the distance tracking error dynamics can be summarized as:

$$\begin{cases} e_i = x_{i-1} - x_i - L_i - r_i - h_i v_i, \\ \dot{e}_i = v_{i-1} - v_i - h_i(u_i - v_i)\,/\tau_i, \\ \ddot{e}_i = -\dot{e}_i\,/\tau_i - \left(u_i + h_i \dot{u}_i\right)\,/\tau_i + (u_{i-1} - v_{i-1})\,/\tau_{i-1} + v_{i-1}\,/\tau_i. \end{cases} \tag{8}$$

### 5.1.3. *Speed control law design*

According to (8), the desired longitudinal velocity of the follower $u_i$ could be designed as:

$$u_i + h\dot{u}_i = \tau_i((u_{i-1} - v_{i-1})\,/\,\tau_{i-1} + v_{i-1}\,/\,\tau_i) + K_d \dot{e}_i + K_p e_i, \tag{9}$$

with $K_d$ and $K_p$ representing the control gains. Note that $u_i$ corresponds exactly to the output of the speed planner in Fig. 2. Substituting (9) into (8), we can cast the tracking error dynamics into a matrix form as:

$$\begin{bmatrix} \dot{e}_i \\ \ddot{e}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\dfrac{K_p}{\tau_i} & -\dfrac{1+K_d}{\tau_i} \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix}. \tag{10}$$

Hence, by selecting the control gains as:

$$\begin{cases} K_p = \eta^2 \tau_i, \\ K_d = 2\eta\tau_i - 1, \end{cases} \tag{11}$$

with $\forall \eta > 0$, the repeated eigenvalues of (10) become:

$$\lambda_1 = \lambda_2 = -\eta < 0, \tag{12}$$

which indicates that the distance tracking error $e_i$ will ultimately converge into zero. There exist various robust and highly efficient control algorithms in the literature to regulate vehicle platoon [29, 30, 31, 32, 33]. However, it should be noted that controller design is not the focus of this paper. Instead, we demonstrate the relative control performance enhancement brought from the channel selection algorithm.

As implied in (9), the proposed speed planner can readily handle a heterogeneous vehicle platoon since the time constant of the follower can be different from the one of the leader. This preferred characteristic was realized by sending the leader's time constant $\tau_{i-1}$ and desired speed $u_{i-1}$, as the safety-related BSMs, to the follower.

Similar to the approach in [34], the time constants of the three scaled cars were determined offline via the *MATLAB System Identification Toolbox*. As revealed in [35], rather than being permanently static, the time constant was indeed influenced by the longitudinal velocity. For the sake of simplicity, only the time constants for a particular accelerating phase and a decelerating phase were identified. During the acceleration phase, a scaled car speeded up from standstill to $0.4m\,/s$. And during the braking phase, a scaled car decelerated from $0.4m\,/s$ to a full stop. Representative identification results are demonstrated in Fig. 6.

Hence, a first-order model was sufficient to approximate the actual dynamics of a scaled car, and the identified time-constants of all the three scaled cars are summarized in Table 1.

Please note that online time-constant identification for a linear system, as the case in (8), is also possible [36]. However, it would further aggravate the computational burden of the control board.

### 5.1.4. *String stability*

Aside from regulating the inter-vehicle distance tracking error, the tracking error propagated along a vehicle platoon cannot be amplified [37]. This requirement is commonly referred to as 'string stability'. As indicated in [38], there exist various definitions of 'string stability' and the most popular definition is:

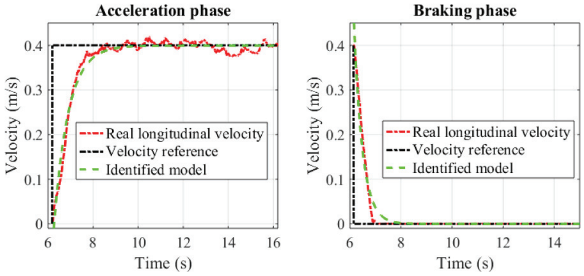


**Fig. 6.** Time constant identification.

**Table 1**
Time-constant Identification Results.

| | Accelerating phase | Braking phase |
|---|---|---|
| Leading car1 | 0.50 s | 0.30 s |
| Middle-car2 | 0.60 s | 0.35 s |
| Last-car3 | 0.50 s | 0.35 s |

$$\left|\frac{E_i(j\omega)}{E_{i-1}(j\omega)}\right| \leq 1, \forall\omega, \forall i \geq 1, \tag{13}$$

where $E_i(j\omega)$ represents the Laplace transform of the tracking error $e_i$ in (8). However, as illustrated in [26], for a linear, *heterogeneous* vehicle platoon with a predecessor-follower communication topology like ours, we should indeed define the string stability condition as:

$$\left|\frac{X_i(j\omega)}{X_{i-1}(j\omega)}\right| \overset{\Delta}{=} \gamma_i(j\omega) \leq 1, \forall\omega, \forall i \geq 1, \tag{14}$$

where $X_i(j\omega), X_{i-1}(j\omega)$ are the Laplace transforms of the vehicle positions $x_i, x_{i-1}$.

We here explain the difference between (13) and (14). By combining (5)-(11), the transfer function from $X_{i-1}(j\omega)$to $X_i(j\omega)$ can be expressed as a block diagram in Fig. 7, with $\rho$ indicating the data transmission delay.

From Fig. 7, the left side of (14) can be calculated as:

$$\left|\frac{X_i(j\omega)}{X_{i-1}(j\omega)}\right| = \left|\frac{K_p + (K_d+1)j\omega - \tau_i\omega^2 e^{-\rho\omega j}}{(1+jh_i\omega)\left(K_p + (K_d+1)j\omega - \tau_i\omega^2\right)}\right|. \tag{15}$$

Instead, omitting the constant term $-(L_i + r_i)$ in (8) and assuming zero initial condition, we can express the Laplace transform of the position tracking error $e_i$ as:

$$E_i(j\omega) = X_{i-1}(j\omega) - X_i(j\omega) - h_i j\omega X_i(j\omega). \tag{16}$$

By substituting (15) into (16), we can formulate (13) as:

$$\begin{aligned} \left|\frac{E_i(j\omega)}{E_{i-1}(j\omega)}\right| &= \left|\frac{X_{i-1}(j\omega) - (1+h_i j\omega)X_i(j\omega)}{X_{i-2}(j\omega) - (1+h_{i-1}j\omega)X_{i-1}(j\omega)}\right| \\ &= \left|\frac{X_{i-1}(j\omega) - (1+h_i j\omega)\gamma_i(j\omega)X_{i-1}(j\omega)}{X_{i-2}(j\omega) - (1+h_{i-1}j\omega)\gamma_{i-1}(j\omega)X_{i-2}(j\omega)}\right| \\ &= \left|\frac{[1-(1+h_i j\omega)\gamma_i(j\omega)]\gamma_{i-1}(j\omega)}{[1-(1+h_{i-1}j\omega)\gamma_{i-1}(j\omega)]}\right|. \end{aligned} \tag{17}$$

Therefore, we have:

$$\lim_{\omega\to 0}\left|\frac{E_i(j\omega)}{E_{i-1}(j\omega)}\right| = \frac{h_i}{h_{i-1}}, \tag{18}$$

where $h_i, h_{i-1}$ are respectively the constant time headway of the follower and the leader. In other words, if we have $h_i > h_{i-1}$, the criterion (13) will never hold, no matter how advanced the V2V communication technique is. On the contrary, based on (15), we have:

$$\lim_{\omega\to 0}\left|\frac{X_i(j\omega)}{X_{i-1}(j\omega)}\right| = 1, \tag{19}$$

which satisfies the string stability condition (14).

Actually, for a *homogenous* platoon where $\rho_i = \rho_{i-1}, \tau_i = \tau_{i-1}, h_i = h_{i-1}$with the control gains fixed by (11), we have:

$$\left|\frac{E_i(j\omega)}{E_{i-1}(j\omega)}\right| = \left|\frac{X_i(j\omega)}{X_{i-1}(j\omega)}\right|, \forall\omega, \forall i \geq 1. \tag{20}$$

In regards to our case, the desired time headways of both the middle car and the last car were uniformly set as $h_i = h_{i-1} = 0.15s$. However, as demonstrated in Table I, the vehicle string was heterogeneous due to the different time constants. Therefore, we adopted (14) to verify the string stability.

Eq. (14) indeed requires that the *amplitude of the oscillation* of the velocity cannot be amplified upstream the vehicle platoon [26] (from the head car1 to the tail car3). From (15), the minimum time headway $h_{\min}(\rho, K_p, K_d, \tau_i) \leq h$, which guarantees (14) can be numerically calculated. Moreover, with fixed control gains $K_p, K_d$ and vehicle time constant $\tau_i$, the minimum time headway $h_{\min}$ can be expressed as a univariate function of the communication delay $\rho$. For instance, the $h_{\min} - \rho$ relationship is demonstrated in Fig. 8 with $\tau_i = 0.5, \eta = 2$ set in (11). Hence, with $h_{\min} = 0.15s$, the maximally tolerable communication delay would be around 0.04 s.

As a final point, if there is no communication delay, $\rho$ in (15) becomes zero, and (15) can be simplified as:

$$\left|\frac{X_i(j\omega)}{X_{i-1}(j\omega)}\right| = \left|\frac{1}{1+h_i s}\right|, \tag{21}$$

which suggests that the string stability condition (14) is inherently maintained for $\forall h_i > 0$.

### 5.1.5. *Front steering control law design*

Since the CACC scenario focused on the vehicle longitudinal speed control, all three scaled cars should run along a straight lane in face of the external disturbances. A simple proportional controller was designed to command the front steering servo of each scaled car, as:

$$\delta(t) = \delta_0 + K_f\left(L_{pre}\sin(\psi(t)) + K_y\Delta Y(t)\right), \tag{22}$$

with $\delta_0$ as the neutral PWM command to zero the front steering, $L_{pre}$ as the preview distance, $\psi(t)$ as the yaw angle, $\Delta Y(t)$ as the lateral offset with respect to the reference straight line, and $K_y, K_f$ being the control gains. Preliminary experimental results validated this simple steering control law as the lateral offsets were less than 40 mm along a 3000 mm straight path.

### 5.1.6. *Experiment result*

In this Section, we demonstrate the CACC control performance

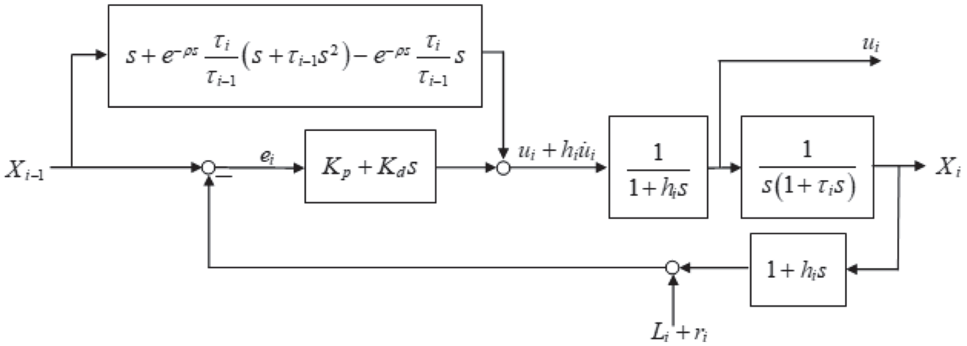

$s + e^{-\rho s}\frac{\tau_i}{\tau_{i-1}}(s+\tau_{i-1}s^2) - e^{-\rho s}\frac{\tau_i}{\tau_{i-1}}s$
$u_i$
$X_{i-1}$
$e_i$
$-$
$K_p + K_d s$
$u_i + h_i\dot{u}_i$
$\frac{1}{1+h_i s}$
$\frac{1}{s(1+\tau_i s)}$
$X_i$
$1 + h_i s$
$L_i + r_i$

**Fig. 7.** Transfer function diagram.

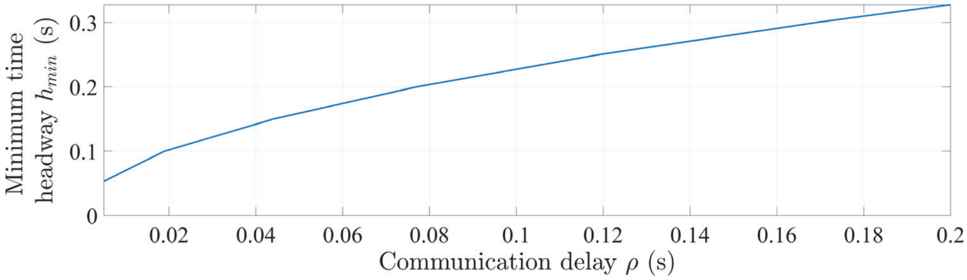


**Fig. 8.** Minimum time headway with respect to the communication delay.

**TABLE 2**
The number of interfering nodes in the three channels.

| | CH22 | CH24 | CH26 |
|---|---|---|---|
| 0–5s | 2 | 5 | 8 |
| 5–8s | 4 | 1 | 10 |
| 8–18s | 1 | 5 | 9 |

enhancement brought by the channel selection algorithm.

We first show the time-varying quantities of the interfering nodes in the three available channels in Table 2.

The channel selection result is demonstrated in Fig. 9, where the selected channel information was derived from TMOTE. Note that due to the relatively small number of the interfering TMOTE modules, the three available channels were all categorized as *Class 1* in Section 2. The main difference among them laid in the number of the interfering nodes in each channel. For the large-scale simulation results with *Class 2* and *Class 3* channel selection, reader can refer to [22].

Therefore, the channel selection algorithm successfully found the optimal channel with the least interfering nodes.

The longitudinal velocities of the three scaled cars with channel selection being active are shown in Fig. 10, and the results without channel selection (when the BSMs went through the default CH26) are shown in Fig. 11.

Comparing Fig. 10 with Fig. 11, we can observe that if the channel selection algorithm was enabled, the velocities of the two following cars: car2, car3, became much smoother and they could follow the speed trend of the leading car1 with more accuracy. Moreover, the speed fluctuations of the leading car1 (blue dashed line) were attenuated by car2 and car3. In contrast, in Fig. 11, if BSMs went through the crowded CH26, the relatively small speed oscillation of the leading car1 could trigger a much-pronounced speed fluctuation of the middle car2. Hence, the channel selection algorithm helped improve the string stability (14), as well.

Another point to note is the oscillatory speeds of car2 and car3 below 0.1 m/s. As we mentioned in Section 3, the longitudinal velocities of both car2 and car3 were controlled by a sliding mode controller, which produced vehicle chattering (moving back and forth) at near-zero speed. Contrasting Fig. 10 with Fig. 11, we can first witness that the magnitudes of such speed oscillations were much larger if the BSMs went through the default CH26. Moreover, the speed oscillation of car2 can be distinguishably reduced in car3 only if the channel selection algorithm was enabled.

Lastly, the following two cars could respond to the movement of the

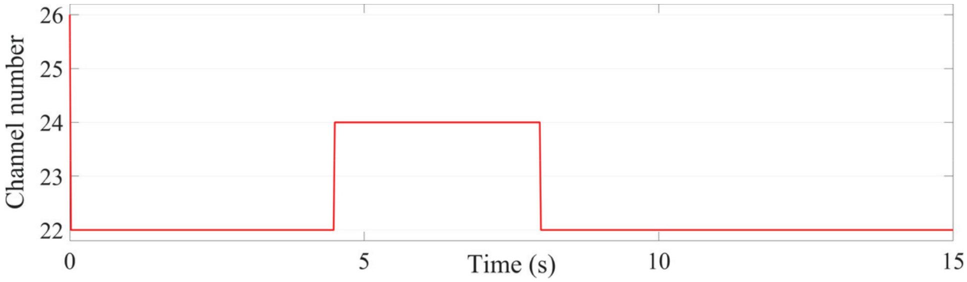


**Fig. 9.** CACC channel selection result.

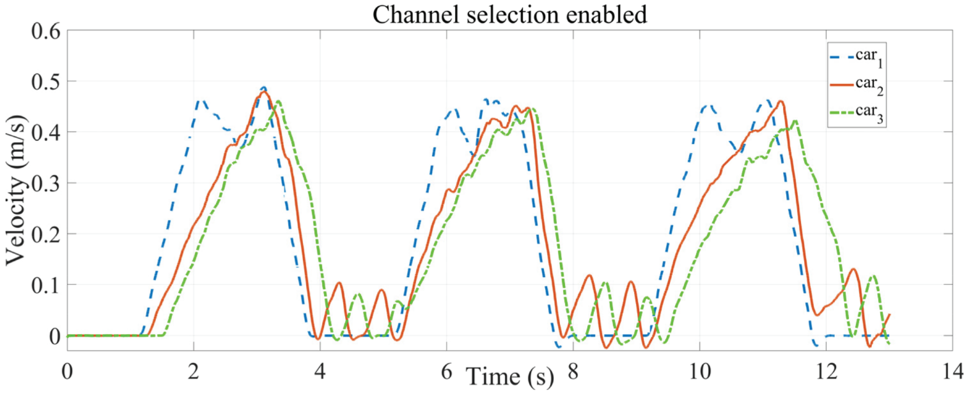


**Fig. 10.** Longitudinal velocities with channel selection enabled.

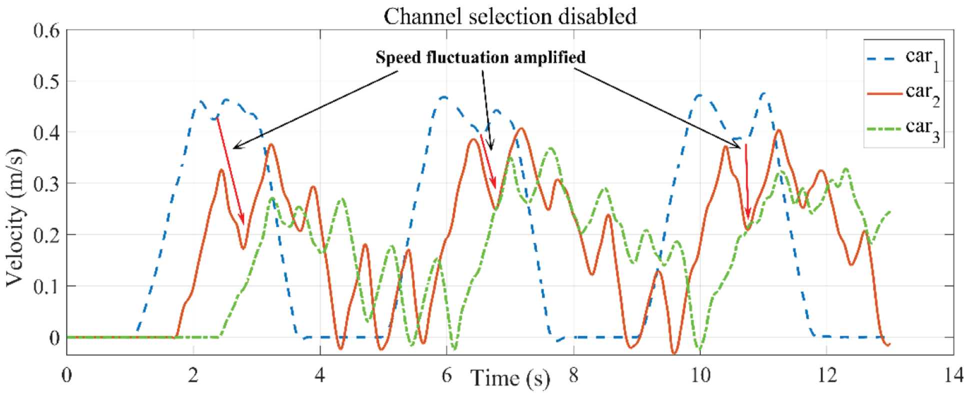


**Fig. 11.** Longitudinal velocities without channel selection.

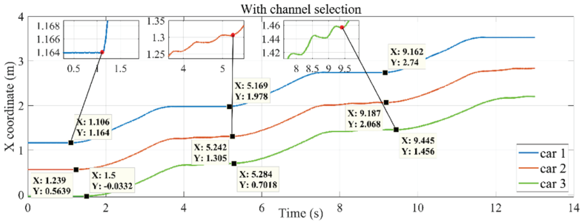


**Fig. 12.** Longitudinal coordinates of scaled cars with channel selection enabled.

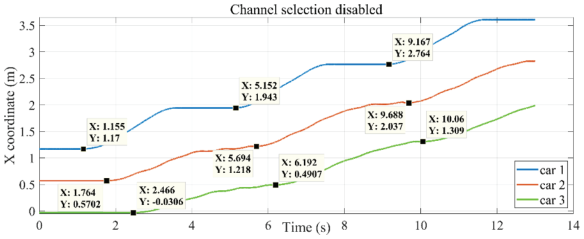


**Fig. 13.** Longitudinal coordinates of scaled cars without channel selection.

leading car1 much faster in Fig. 10 than in Fig. 11. This fact is further validated in Fig. 12 and Fig. 13, which exhibit the longitudinal coordinates $X_{1,2,3}$ of the three scaled cars with/without channel selection, respectively.

The enlarged graphs in Fig. 12 indicate the actual starting moments (after the transitional chattering) of each car. The counterparts of each car without channel selection are marked in Fig. 13.

The averaged time elapse between the start-to-forward moments of car1/car2 and car1/car3 are summarized in Table 3.

Hence, the following cars can react prompter to the movement of the leading car1 with channel selection.

Subsequently, the absolute distance tracking errors between car1/car2 and car2/car3 are demonstrated in Fig. 14, where the left subplot corresponds to the case with channel selection enabled and the right subplot shows the tracking errors when the BSMs passed through the default CH26.

Therefore, with channel selection enabled, the magnitude of the distance tracking error between car2/car3 was always beneath the one between car1/car2. Moreover, both the distance tracking errors between car1/car2 and car2/car3 were substantially reduced with respect to the cases when the channel selection was disabled. The Root Mean Square (RMS) values and the maximum of the absolute distance tracking errors are summarized in Table 4, where $e_2$, $e_3$ indicate respectively the distance tracking error of the middle car2 and the last car3. With the aid of channel selection, the distance tracking accuracy can be cogently improved.

**Table 3**
Vehicle Start Delays.

| | car1/car2 | car1/car3 |
|---|---|---|
| With channel selection | **0.064s** | **0.262s** |
| No channel selection | 0.557s | 1.081s |

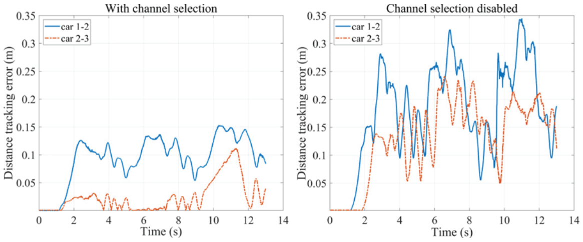


**Fig. 14.** Absolute distance tracking errors.

All in all, Figs. 10-14 show that with the dynamic channel selection algorithm, the following cars can respond more swiftly to their immediate predecessor, the distance tracking errors of a CACC platoon can be substantially reduced, and the string stability could be guaranteed.

## 5.2. *Run-the-Red-Light at an intersection*

### 5.2.1. *Scenario description*

In this scenario, there existed an autonomous-driving CACC platoon with two cars: CACC leader (car2) and CACC follower (car3), passing through an intersection from the south to the north. In the meantime, there was an ambulance (car1) coming from the west, and the ambulance ran the red light. Thus, to avoid the collision and ask the platoon to yield the road, the ambulance would send a "Stop" packet (i.e., an ESM) every 0.02 s to the CACC leader. Upon receiving such a 'Stop' flag, the CACC leader changed its referential speed from a positive value to zero. During the whole process, the CACC leader continuously sent its time constant and referential speed (i.e., BSMs) to the follower, similar to the case in Section 5.1.

Likewise, there were fifteen interfering nodes around the intersection. Each of them occupied one of the two available channels: CH26 or CH24. Unlike the CACC scenario, such a run-the-red-light situation should only last in a short period in practice. Therefore, the numbers of interfering nodes in CH26 and CH24 were set constant. Fig. 15 illustrates this scenario and the numbers of interfering nodes in the two available channels are listed in Table 5.

Akin in Section 5.1, two experiments were conducted with either the dynamically selected channel or the default CH26 utilized for ESMs and BSMs transmissions. When the distance between the ambulance and the CACC leader went below 1 m, the CACC platoon and the ambulance

**Table 4**
Distance Tracking Errors.

| | $RMS(e_2)$ | $RMS(e_3)$ | $\max(\|e_2\|)$ | $\max(\|e_3\|)$ |
|---|---|---|---|---|
| With channel selection | **0.1015m** | **0.0365m** | **0.1534m** | **0.1128m** |
| No channel selection | 0.1973m | 0.1480m | 0.3451m | 0.2416m |

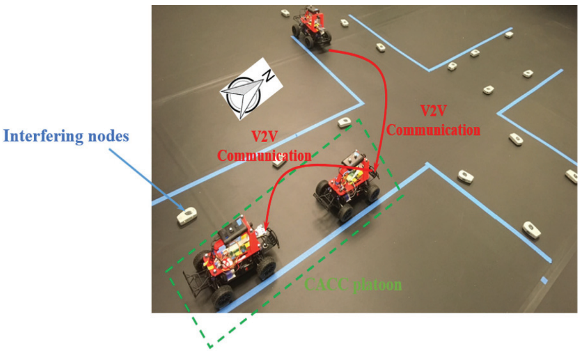


**Fig. 15.** Run the red light at an intersection.

formed a communication group, and the channel selection algorithm would be triggered if it had been enabled. Then, if the distance further decreased to less than 0.98 m, the ambulance would start sending ‘Stop’ flags to the CACC leader.

### 5.2.2. Experimental result

The channel selection result is demonstrated in Fig. 16. Therefore, the better channel with much less interfering nodes was successfully identified and selected.

Fig. 17 displays the longitudinal velocities of both the CACC leader (car2) and the follower (car3).

The upper and the lower subplots of Fig. 17 demonstrate the longitudinal speed profile of the CACC leader and the follower, respectively. Besides, the red dashed lines reflect the results by use of the default CH26, whereas the blue solid lines correspond to the case with the channel selection algorithm being active. Therefore, with the help of the channel selection algorithm, the CACC leader started braking (adopted the zero referential speed) almost immediately (0.022 s) after the ambulance sent out the first ‘Stop’ packet. Instead, if the ‘Stop’ packets needed to pass through the crowded CH26, the CACC leader could not brake until 0.269 s after the first ‘Stop’ was issued. Moreover, as demonstrated in the lower subplot of Fig. 17, if the CACC leader (car2) could swiftly switch to the zero referential speed, then the CACC follower (car3) could decelerate—when its own reference speed calculated from (9) went below its actual longitudinal velocity—0.241 s after the first ‘Stop’ packet from the ambulance was fired. On the contrary, if the channel selection algorithm was not enabled, then the CACC follower could not produce a braking behavior until 0.61 s after the ambulance sent the first ‘Stop’ message.

The tardive response of the CACC platoon due to the delayed alert caused a serious consequence, which is demonstrated in Fig. 18.

Fig. 18 shows the distance between the center of gravities (CGs) of the CACC leader and the ambulance, deduced from OptiTrack. As both the ambulance and the CACC platoon run towards the intersection, this distance decreased at the beginning. If the channel selection algorithm was enabled, the CACC platoon could timely stop in front of the ambulance, which ran the red light. As a consequence, the minimum CG distance between the CACC leader and the ambulance was above 0.365 m. As the ambulance continued moving straight, this distance increased afterward. In contrast, if the ‘Stop’ flags came through the default CH26, the braking response of the CACC leader was largely delayed, which indeed caused a side collision between itself and the ambulance at around 2.35 s. Once the collision occurred, both the CACC leader and the ambulance stopped at the collision point, which explained the flat red dashed line after 2.35 s.

**Table 5**
Number of interfering nodes in the two channels.

| | CH26 | CH24 |
|---|---|---|
| 0–4s | 13 | 2 |

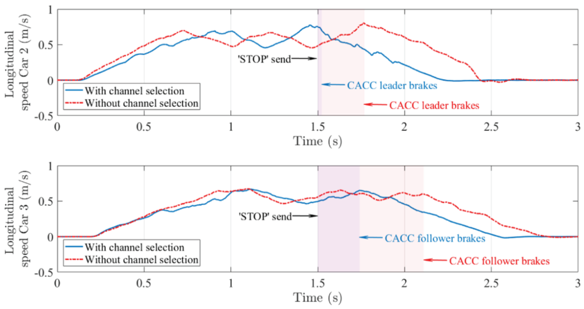


**Fig. 17.** Speed profiles of CACC leader and follower.

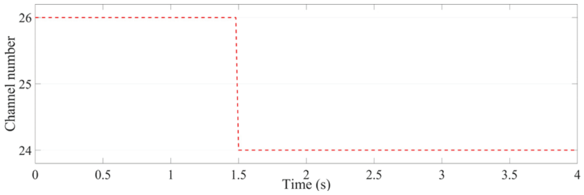


**Fig. 16.** Run-the-red-light channel selection result.

In conclusion, Fig. 17 and Fig. 18 imply that if the critical ESMs passed through the dynamically selected channel, its transmission delay could be cogently reduced. Consequently, the collision between the CACC leader and the ambulance can be avoided.

## 6. Conclusions

Current DSRC policy obliges safety-related messages to be exchanged principally via a unique channel. This restriction can induce a huge transmission delay and endanger drivers. Hence, a dynamic multi-channel selection algorithm is validated in this paper, which makes all the involved vehicles of a specific V2V application work collaboratively to identify the most appropriate channel for transmitting the safety-critical messages. Experiments demonstrated the effectiveness of the algorithm in reducing the transmission delay of both the BSMs in a CACC scenario and the ESMs in a run-the-red-light scenario. The algebraic differentiation technique and a newly designed CACC speed planner were formulated to accomplish the experiments.

Future work will focus on validating the channel selection algorithm in more complex traffic scenarios under various communication topologies.

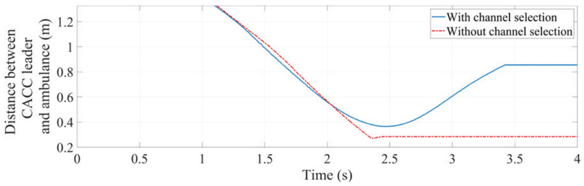


**Fig. 18.** Distance between CACC leader and the ambulance.

## Funding

This work was supported by the National Science Foundation with award number 1901632.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported.

## References

[1] Azimi SR, Bhatia G, Rajkumar RR, Mudalige P. Vehicular networks for collision avoidance at intersections. SAE Int J Passeng Cars Mech Syst 2011;4(2011–1–573):406–16. https://doi.org/10.4271/2011-01-0573.

[2] Li Z, Elefteriadou L, Ranka S. Signal control optimization for automated vehicles at isolated signalized intersections. Transp Res Part C Emerg Technol 2014;49:1–18. https://doi.org/10.1016/j.trc.2014.10.001.

[3] Dang R, Ding J, Su B, Yao Q, Tian Y, Li K. A lane change warning system based on V2V communication. In *Proc IEEE 17th Int ITSC* 2014:1923–8. https://doi.org/10.1109/ITSC.2014.6957987. IEEE.

[4] Tang A, Yip A. Collision avoidance timing analysis of DSRC-based vehicles. Accid Anal Prev 2010;42(1):182–95. https://doi.org/10.1016/j.aap.2009.07.019.

[5] Bauza R, Gozalvez J, Sanchez-Soriano J. Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In *IEEE Local Comput Netw Conf* 2010:606–12. https://doi.org/10.1109/LCN.2010.5735780. IEEE.

[6] Xiao L, Gao F. A comprehensive review of the development of adaptive cruise control systems. Veh Sys Dyn 2010;48(10):1167–92. 10.1080/00423110903365910.

[7] Shladover SE, Nowakowski C, Lu XY, Ferlis R. Cooperative adaptive cruise control: definitions and operating concepts. Transp Res Rec 2015;2489(1):145–52. https://doi.org/10.3141/2489-17.

[8] Milanés V, Shladover SE. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. Transp Res Part C Emerg Technol 2014;48:285–300. https://doi.org/10.1016/j.trc.2014.09.001.

[9] Thunberg J, Lyamin N, Sjöberg K, Vinel A. Vehicle-to-vehicle communications for platooning: safety analysis. IEEE Netw Lett 2019;1(4):168–72. https://doi.org/10.1109/LNET.2019.2929026.

[10] Carpenter SE, Sichitiu ML. Analysis of packet loss in a large-scale DSRC field operational test. In *2016 Int Conf Perform Evaluat Model Wired Wireless Netw (PEMWN)* 2016:1–6. https://doi.org/10.1109/PEMWN.2016.7842909. IEEE.

[11] Cunha F, Villas L, Boukerche A, Maia G, Viana A, Mini RA, et al. Data communication in VANETs: protocols, applications and challenges. Ad Hoc Netw 2016;44:90–103. https://doi.org/10.1016/j.adhoc.2016.02.017.

[12] Kenney JB. Dedicated short-range communications (DSRC) standards in the United States. Proc IEEE. 2011;99(7):1162–82. https://doi.org/10.1109/JPROC.2011.2132790.

[13] Jiang D, Taliwal V, Meier A, Holfelder W, Herrtwich R. Design of 5.9GHz DSRC-based vehicular safety communication. IEEE Wirel Commun 2006;13(5):36–43. https://doi.org/10.1109/WC-M.2006.250356.

[14] Bai Y, Zheng K, Wang Z, Wang X, Wang J. Dynamic Channel Selection for Real-Time Safety Message Communication in Vehicular Networks. In *2018 IEEE Real-Time Syst Sympos (RTSS)* 2018:56–66. https://doi.org/10.1109/RTSS.2018.00016. IEEE.

[15] Sepulcre M, Mittag J, Santi P, Hartenstein H, Gozalvez J. Congestion and awareness control in cooperative vehicular systems. Proc IEEE. 2011;99(7):1260–79. https://doi.org/10.1109/JPROC.2011.2116751.

[16] Campolo C, Molinaro A, Vinel A, Zhang Y. Modeling prioritized broadcasting in multichannel vehicular networks. IEEE Trans Veh Technol 2011;61(2):687–701. https://doi.org/10.1109/TVT.2011.2181440.

[17] Wang Q, Leng S, Fu H, Zhang Y. An IEEE 802.11 p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks. IEEE Trans Intell Transp Syst 2011;13(2):449–58. https://doi.org/10.1109/TITS.2011.2171951.

[18] Yao Y, Zhang K, Zhou X. A flexible multi-channel coordination MAC protocol for vehicular ad hoc networks. IEEE Commun Lett 2017;21(6):1305–8. https://doi.org/10.1109/LCOMM.2017.2681060.

[19] Mboup M, Join C, Fliess M. Numerical differentiation with annihilators in noisy environment. Numer Algorithms 2009;50(4):439–67. https://doi.org/10.1007/s11075-008-9236-1.

[20] Sadi Y, Ergen SC, Park P. Minimum energy data transmission for wireless networked control systems. IEEE Trans Wireless Commun 2014;13(4):2163–75. https://doi.org/10.1109/TWC.2014.0204.131204.

[21] Yao Y, Rao L, Liu X. Performance and reliability analysis of IEEE 802.11 p safety communication in a highway environment. IEEE Trans Veh Technol 2013;62(9):4198–212. https://doi.org/10.1109/TVT.2013.2284594.

[22] Bai Y, Zheng K, Wang Z, Wang X, Wang J. MC-Safe: multi-channel Real-time V2V Communication for Enhancing Driving Safety. ACM Trans Cyber-Phys Syst 2020;4(4):1–27. https://doi.org/10.1145/3394961.

[23] Johnson M, Healy M, Van de Ven P, Hayes MJ, Nelson J, Newe T, Lewis E. A comparative review of wireless sensor network mote technologies. Proc IEEE Sensors 2009:1439–42. https://doi.org/10.1109/ICSENS.2009.5398442. IEEE.

[24] Fliess M, Join C, Sira-Ramirez H. Non-linear estimation is easy. Int J Model Ident Contr 2008;4(1):12–27. https://doi.org/10.1504/IJMIC.2008.020996.

[25] Wang Z, Wang J. Ultra-local model predictive control: a model-free approach and its application on automated vehicle trajectory tracking. Control Eng Pract 2020;101:104482. https://doi.org/10.1016/j.conengprac.2020.104482.

[26] Naus GJ, Vugts RP, Ploeg J, van De Molengraft MJ, Steinbuch M. String-stable CACC design and experimental validation: a frequency-domain approach. IEEE Trans Veh Technol 2010;59(9):4268–79. https://doi.org/10.1109/TVT.2010.2076320.

[27] Wang J, Rajamani R. Should adaptive cruise-control systems be designed to maintain a constant time gap between vehicles? IEEE Trans Veh Technol 2004;53(5):1480–90. https://doi.org/10.1109/TVT.2004.832386.

[28] Rajaram V, Subramanian SC. Heavy vehicle collision avoidance control in heterogeneous traffic using varying time headway. Mechatronics 2018;50:328–40. https://doi.org/10.1016/j.mechatronics.2017.11.010.

[29] Yi SY, Chong KT. Impedance control for a vehicle platoon system. Mechatronics 2005;15(5):627–38. https://doi.org/10.1016/j.mechatronics.2004.12.002.

[30] Ferrara A, Vecchio C. Second order sliding mode control of vehicles with distributed collision avoidance capabilities. Mechatronics 2009;19(4):471–7. https://doi.org/10.1016/j.mechatronics.2008.11.002.

[31] Firooznia A, Ploeg J, van de Wouw N, Zwart H. Co-design of controller and communication topology for vehicular platooning. IEEE Trans Intell Transp Syst 2017;18(10):2728–39. https://doi.org/10.1109/TITS.2017.2660544.

[32] Abou Harfouch Y, Yuan S, Baldi S. An adaptive switched control approach to heterogeneous platooning with intervehicle communication losses. IEEE Control Netw Syst 2017;5(3):1434–44. https://doi.org/10.1109/TCNS.2017.2718359.

[33] Acciani F, Frasca P, Stoorvogel A, Semsar-Kazerooni E, Heijenk G. Cooperative adaptive cruise control over unreliable networks: an observer-based approach to increase robustness to packet loss. In *2018 Eur. Control Conf. (ECC)* 2018:1399–404. https://doi.org/10.23919/ECC.2018.8550585. IEEE.

[34] Milanés V, Shladover SE, Spring J, Nowakowski C, Kawazoe H, Nakamura M. Cooperative adaptive cruise control in real traffic situations. IEEE Trans Intell Transp Syst 2013;15(1):296–305. https://doi.org/10.1109/TITS.2013.2278494.

[35] Lidstrom K, Sjoberg K, Holmberg U, Andersson J, Bergh F, Bjade M, Mak S. A modular CACC system integration and design. IEEE Trans Intell Transp Syst 2012;13(3):1050–61. https://doi.org/10.1109/TITS.2012.2204877.

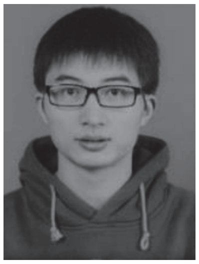
[36] Fliess M, Sira–Ramírez H. An algebraic framework for linear identification. Esaim Contr Optim CA. 2003;9:151–68. https://doi.org/10.1051/cocv:2003008.

[37] Rajamani R. Adaptive Cruise Control. Vehicle dynamics and control. 2nd ed. New York, NY, USA: Springer Science and Business Media; 2012. p. 158–9. https://doi.org/10.1007/978-1-4614-1433-9. ch. 6, sec. 4.

[38] Feng S, Zhang Y, Li SE, Cao Z, Liu HX, Li L. String stability for vehicular platoon control: definitions and analysis methods. Annu Rev Control 2019. https://doi.org/10.1016/j.arcontrol.2019.03.001.

**Zejiang Wang** received the B.E. degree (with honors) in mechanical engineering and automation from Southeast University, Nanjing, China, in 2014, the Dipl. Ing. degree from ENSTA ParisTech, Palaiseau, France, and the M.S. degree in design, modeling, and architecture of complex industrial systems from École Polytechnique, University of Paris-Saclay, Palaiseau, France, both in 2017. He is currently pursuing the Ph.D. degree from the Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, USA. His-research interests include vehicle dynamics and control, combustion engine control, and cyber-physical system.

**Yunhao Bai** received his B.E. and M.S degree in electrical engineering from Shandong University in 2014 and The Ohio State University in 2016, respectively. Currently, he is a PhD student in the Power-Aware Computer System lab at The Ohio State University. His-current research interests include real-time and embedded systems, network sensing, and signal processing.



**Jingqiang Zha** received the B.E. degree in mechanical engineering and automation from Shanghai Jiaotong University, Shanghai, China, in 2013 and the M.S. degree in mechanical engineering from The University of Texas at Austin, Austin, TX, USA, in 2015. He is currently pursuing the Ph.D. degree from the Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, USA. His-research interests include vehicle dynamics and control, robust control, and LPV control.

**Junmin Wang** received the B.E. degree in automotive engineering and the first M.S. degree in power machinery and engineering from the Tsinghua University, Beijing, China, in 1997 and 2000, respectively. He received the second and third M.S. degrees in electrical engineering and mechanical engineering from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2003, and the Ph.D. degree in mechanical engineering from the University of Texas at Austin, Austin, TX, USA, in 2007. Dr. Junmin Wang is the Accenture Endowed Professor in Department of Mechanical Engineering at University of Texas at Austin. In 2008, he started his academic career at Ohio State University, where he founded the Vehicle Systems and Control Laboratory, was early promoted to Associate Professor in September 2013 and then very early promoted to Full Professor in June 2016. He also gained five years of full-time industrial research experience at Southwest Research Institute (San Antonio Texas) from 2003 to 2008. Prof. Wang has a wide range of research interests covering control, modeling, estimation, optimization, and diagnosis of dynamical systems, especially for automotive, smart and sustainable mobility, human-machine, and cyber-physical system applications. Dr. Wang is the author or co-author of more than 330 peer-reviewed publications including 170 journal articles and 13 U.S. patents. Prof. Wang is a recipient of numerous international and national honors and awards. He is an IEEE Vehicular Technology Society Distinguished Lecturer, SAE Fellow, and ASME Fellow.

**Xiaorui Wang** received the D.*Sc.* degree in Computer Science from Washington University in St. Louis in 2006. He is currently a professor in the Department of Electrical and Computer Engineering at The Ohio State University. He is the recipient of the US Office of Naval Research (ONR) Young Investigator (YIP) Award in 2011 and the US NSF CAREER Award in 2009. He also received the Best Paper Award from the 29th IEEE Real-Time Systems Symposium (RTSS) in 2008. He is an author or coauthor of more than 100 refereed publications. From 2006 to 2011, he was an assistant professor at the University of Tennessee, Knoxville, where he received the Chancellor's Award for Professional Promise and the College of Engineering Research Fellow Award. His-research interests include computer architecture and systems, data center power management, and cyber-physical systems. He is the General Chair of the 14th IEEE International Conference on Autonomic Computing (ICAC 2017) and the TPC Co-Chair of the 24th IEEE/ACM International Symposium on Quality of Service (IWQoS 2016). He is also an associate editor of the IEEE Transactions on Parallel and Distributed Systems (TPDS), the IEEE Transactions on Computers (TC), and the IEEE Transactions on Cloud Computing (TCC).