The joint network/control design of platooning algorithms can enforce guaranteed safety constraints

Giordano, Giulia; Segata, Michele; Blanchini, Franco; Lo Cigno, Renato

DOI
10.1016/j.adhoc.2019.101962

Publication date
2019

Document Version
Accepted author manuscript

Published in
Ad Hoc Networks

The joint network/control design of platooning algorithms can enforce guaranteed safety constraints

Giulia Giordano[a,*], Michele Segata[b,**], Franco Blanchini[c], Renato Lo Cigno[b]

[a] *Delft Center for Systems and Control, Delft University of Technology, The Netherlands*
[b] *Dept. of Information Engineering and Computer Science, University of Trento, Italy*
[c] *Dept. of Mathematics, Computer Science and Physics, University of Udine, Italy*

Abstract

Vehicular networks supporting cooperative driving are among the most interesting and challenging ad-hoc networks. Platooning, or the act of coordinating a set of vehicles through an ad-hoc network, promises to improve traffic safety, and at the same time reduce congestion and pollution. The design of the control system for this application is challenging, especially because the coordination and cooperation between vehicles is obtained through a wireless network. So far, control and network issues of platooning have been investigated separately, but this is definitely a sub-optimal approach, as constraints of the networked control system impose bounds on the network performance, and network impairments translate into disturbances on the controlled system. In this work we design a cooperative driving system from a joint network and control perspective, determining upper bounds on the error subject to packet losses in the network, so that the actual inter-vehicle gap can be tuned depending on vehicle or network performance. Extensive simulations show that the system is very robust to packet losses and that the derived bounds are never violated. In addition, since the leader control law is part of the proposed control approach, we show that, besides taking into account external events and reacting within the given constraints to ensure the overall road safety, the system can be easily integrated into global traffic optimization tools that mandate the platoon behavior.

Keywords: platooning, networked-control design

1. Introduction

Cooperative driving is a promising solution to reduce traffic congestion and increase safety, thus addressing at once two major problems of modern transportation on roads. Wireless communication enables vehicles to share information about their status and the sensed surrounding environment: this drastically increases their perception of what happens around them and enables cooperation. Using only standard in-car sensors, as currently done by prototype self-driving vehicles, does not empower this ability, thus in many ways self-driving vehicles share the same limitations of human drivers. Instead, as an example, a wireless link can let a vehicle know the *future* intended trajectory of another one (at an intersection, as a long term destination or cruising speed, etc). Of course, this feat cannot be achieved with on-board sensors only.

Cooperative Adaptive Cruise Control (CACC) is a communication enhanced version of a standard Adaptive Cruise Control (ACC), capable of maintaining a very small inter-vehicle spacing while ensuring passengers' safety. While commercial ACC systems implement a time-headway spacing policy with a time-headway not smaller than 1 s [1] (meaning a distance not smaller than 36 m at 130 km/h), CACC systems can implement smaller time headways or even spacing policies in which the distance is constant (e.g., 5 m) independent of the cruising speed [2]. The CACC forms trains of vehicles, called *platoons*, so this application is also known with the name of *cooperative automatic driving*, or *platooning*. Platooning provides benefits in terms of efficiency, safety, and driving comfort [3, 4]. A smaller inter-vehicle gap allows for a better use of the road infrastructure (where most of the space is now simply wasted due to safety distances), improves traffic flow, reduces congestion and, at the same time, the waste of fuel due to start and stop dynamics caused by congestion itself. Since statistics show that human driving is the cause of more than 90 % of the accidents [5], we can expect that an automated system taking control over driving tasks would also improve safety. Finally, comfort is improved: the "former driver" does no longer need to focus on driving and is free to engage in other activities.

When designing a cooperative driving system, we need to face a control-theoretical problem that is inevitably intertwined with networking problems. The control algorithm receives as input information about the other vehicles in the platoon (e.g., speed, position, or acceleration), which is conveyed via wireless links, through periodic broadcast (or beaconing), as well as via local sensors that can improve the precision of distance and relative speed measures. Data packets can be lost due to the

*Dr. Giordano was partially supported by the DTF grant at the Delft University of Technology.

**Dr. Segata was partially supported by the University of Trento within the framework of young researcher support (Bando Giovani Ricercatori 2018).

Email addresses: g.giordano@tudelft.nl (Giulia Giordano), msegata@disi.unitn.it (Michele Segata), blanchini@uniud.it (Franco Blanchini), locigno@disi.unitn.it (Renato Lo Cigno)

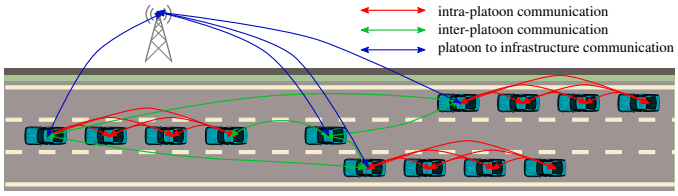


Figure 1: Platooning ad-hoc network.

inherent nature of wireless links, and this in turn has a dramatic impact on the performance of the application. Bad performance of autonomous driving can result in injuries or loss of life. Figure 1 depicts the typical ad-hoc network supporting platooning with the Dedicated Short Range Communications (DSRC) approach [6]. The network is established via 802.11p [7] beacons and moves with the platoon. Different networking approaches as C-V2X [8] may require an adaptation of the design, but the basic principle we propose remains the same. The ad-hoc network is needed both for intra-platoon communication (for control purposes) and for inter-platoon communication (for coordination and maneuvering). Vehicles participating in the ad-hoc network might get support (e.g., for finding a nearby platoon) or advises (e.g., for traffic regulation and safety) from a fixed infrastructure. Ad-hoc networks are the key enabler of future Intelligent Transportation Systems (ITSs), independent of the actual application [9, 10, 11]. Clearly, different applications have different requirements. Cooperative driving and safety applications are among the most challenging from the point of view of the ad-hoc network, as delays and/or loss of information can result in poor application performance. The design of proper protocols is thus extremely challenging [12, 13], but very often only network performance metrics are taken into account.

The impact of wireless impairments on the control performance is not considered, or considered only partially, in most of the works in the field. In this paper we propose a cooperative driving algorithm that specifically takes into account error dynamics due to loss of data and ensures that a predefined safety bound is never violated, given a particular worst-case scenario. This paper extends our previous work [14] including detailed derivations and proofs that were omitted for the sake of brevity, introducing possible extensions, and especially proposing a new mode, the *override mode*, capable of handling external events, such as the presence of slower vehicles ahead or infrastructure-mandated changes of the cruising speed. This additional feature is crucial, since it makes the proposed algorithm a complete platooning control system.

To the best of our knowledge, this is the first attempt to jointly design a control algorithm and a dedicated communication protocol that takes into account packet losses. Our main contributions can be summarized as follows:

- The design jointly considers control and network performance. The controller parameters can be tuned to obey predefined bounds on the position error, given an upper bound on the input error caused by network impairments (Sections 3 and 5). Simulations show that the controller never violates the imposed safety constraints (Section 7);
- The algorithm can maintain a constant spacing policy thanks to a leader plus bidirectional control topology, which comes with no additional network overhead with respect to a commonly assumed leader- plus predecessor-following scheme (Sections 2 and 7);
- The algorithm handles the presence of slower vehicles ahead or infrastructure-based speed advises and reacts to them within given constraints (i.e., a safety distance or a target position). In case the given constraints are missed by the proposed standard controller, the system switches to a modified, more aggressive control mode for a short period of time. This mechanism is detailed in Section 6 and analyzed by means of simulations in Section 7.

2. Background and Related Work

The design of a cooperative automatic driving (or platooning) system is definitely a challenging task, addressed by a large body of literature. Different solutions have been proposed, with different design assumptions and thus characteristics. The main goal is to keep the inter-vehicle gap as small as possible, while ensuring passengers' safety. The key difference to standard ACC solutions is the use of wireless communication for sharing control data with potentially all the vehicles in the platoon. Wireless communication allows a vehicle to "see" behind and ahead other vehicles; this is not possible by using standard sensors, which can only detect objects that are in direct line-of-sight. Even if a sensor could detect objects that are not within the line-of-sight, it could only detect events *after* their occurrence. Communication enables a vehicle to inform the others about what it is going to do, letting them "know the future". Communications for platooning are achieved by means of Cooperative Awareness Messages (CAMs) or *beacons*, broadcast messages sent periodically every T s; when a new beacon is generated it supersedes the previous one, so that stale information is lost rather than propagated with delay. Propagation and processing delays are negligible compared to T.

A key design choice is the *logical control topology*, indicating from which members each vehicle is considering data to compute the control action. This is different from the actual network topology, which is typically broadcast-like. Even if the network topology is a full mesh, the control algorithm may simply exploit a subset of the received information. In fact, the real network topology is unknown in the design phase, as it depends on the context, the technology, and the higher level protocols. A conservative, "technology agnostic" choice may be to use the information received from the front vehicle only: information received from other vehicles (for instance, in a broadcast-like communication technology such as IEEE 802.11p) will simply be ignored. Possible control topologies proposed in the literature are: predecessor-following [1, 15]; leader- and predecessor-following [16, 17, 18, 19], which considers in addition the information of the first vehicle; bi-directional [20] and potentially all-to-all [21]. The choice of the control topology affects the system performance, in particular with respect to the gap policy. Predecessor-following control topologies are proven to be string-stable only under a constant time headway

gap policy [1, 16]: the distance is constant in time, hence the faster the vehicles, the larger the gap. If this policy is not respected, then the string-stability property is violated: distance errors at the head of the platoon may be propagated and amplified towards the end, potentially leading to collisions. By adding a link to the leader, instead, the system can be string-stabilized with respect to a constant spacing gap: the distance is fixed and it is not related to cruising speed [16].

String-stability, however, is not generally related to the distance (or the time headway) vehicles should maintain to avoid collisions in case of packet losses. The performance of a cooperative automatic driving system is typically analyzed with a pure control-theoretic approach, so that a quantitative characterization of the safety gap as a function of the network conditions is hard to find in the literature [22]. A contribution in this direction is [23], which proposes a different control framework (event-triggered control) that deals with the variable sampling time induced by network delays and losses, showing string-stability of the proposed control system.

In our work, instead, we propose a *joint* network and control design of a cooperative automatic driving system, which allows us to compute the minimum inter-vehicle distance depending on worst-case network condition. If the constraints considered for the worst-case analysis are respected, then the distance error cannot be larger than the computed bound. To the best of our knowledge, this is the first attempt to realize such kind of joint network-control system.

3. Control Algorithm

We propose a distributed controller (inspired by an analogy with spring-damper mechanical systems, or impedance-matched transmission lines [24]) that ensures string stability, as proven by Eq. (27) in Section 3.3. As in [14], the control action on each vehicle relies on information about the vehicle in front and the one behind (bidirectional topology); all vehicles share a common dynamic reference speed $v(t)$, which can be either imposed by the first vehicle of the platoon, resulting in a control topology similar to [16], or decided by any other vehicle, or taken from an external source (e.g, speed indications coming from the infrastructure). Even the leader follows the reference speed with a transient. The common reference speed is shared through the communication link, which is local (and for technologies such as IEEE 802.11p or Cellular-V2X even broadcast); hence, the propagation delay mainly comes from medium access control procedures, and is negligible (a few milliseconds) compared to the system dynamics. Table 1 reports the main notation used throughout the paper.

All vehicles are governed by the dynamic equation

$$\ddot{y}_i = u_i + \delta_i$$

where $\ddot{y}_i$ is the acceleration resulting from the controlled input u_i and the disturbance δ_i. The control action u_i depends on the position and speed of the vehicle i, the positions and speeds of the vehicles in front and behind (if any), the desired distance, and the reference speed:

$$u_i = -k(y_i - y_{i+1} - d) - k(y_i - y_{i-1} + d) - h(\dot{y}_i - \dot{y}_{i+1}) - h(\dot{y}_i - \dot{y}_{i-1}) - r(\dot{y}_i - v).$$

$v(t)$	reference speed (equal for all vehicles)
$a(t)$	average position of the platoon, it can be interpreted as the barycenter of the platoon
$y_i, \dot{y}_i, \ddot{y}_i$	position, speed, and acceleration of vehicle i
$z_i, \dot{z}_i, \ddot{z}_i$	differential position, speed, and acceleration of vehicle i w.r.t. vehicle $i-1$, $i = 2, \ldots, N$
u_i	controller input to vehicle i
d	desired distance between vehicles
k	elastic coefficient
h	inter-vehicle friction coefficient
r	vehicle-reference friction coefficient
δ_i	communication-induced disturbance term
N_L	maximum number of consecutive packets lost
T	beacon interval

Table 1: Main notation used in the paper.

The resulting dynamic system has the following equations. For vehicle 1 (the leader),

$$\ddot{y}_1 = -k(y_1 - y_2 - d) - h(\dot{y}_1 - \dot{y}_2) - r(\dot{y}_1 - v) + \delta_1, \tag{1}$$

for vehicles $i = 2, \ldots N-1$,

$$\begin{aligned}\ddot{y}_i = &-k(y_i - y_{i+1} - d) - k(y_i - y_{i-1} + d) \\ &- h(\dot{y}_i - \dot{y}_{i+1}) - h(\dot{y}_i - \dot{y}_{i-1}) - r(\dot{y}_i - v) + \delta_i\end{aligned} \tag{2}$$

and, for vehicle N,

$$\ddot{y}_N = -k(y_N - y_{N-1} + d) - h(\dot{y}_N - \dot{y}_{N-1}) - r(\dot{y}_N - v) + \delta_N. \tag{3}$$

The control algorithm is designed by choosing the coefficients h, k, and r. The noise term δ_i is essentially due to packet losses because delays are negligible in DSRC systems. Furthermore, as beacons are sent by default ever $T = 100\,\text{ms}$, which is a fairly large sampling time for control systems, the loss of packets means that the information available at the controller can be grossly wrong, introducing uncertainties much larger than sensor or GPS errors. Section 5 discusses in detail the relationship between packet losses and δ_i.

3.1. Analysis

Consider the model in Eqs. (1) to (3) with $d = 0$. This is equivalent to changing the variables as $\hat{y}_i = y_i + d(1-i)$, $i = 1, \ldots, N$, so that the condition $\hat{y}_1 = \hat{y}_2 = \cdots = \hat{y}_N$ is achieved when the true distance between consecutive vehicles is d as desired; we drop the hat to keep the notation simpler.

Let $\bar{\mathbf{1}}$ be the all-one vector $\bar{\mathbf{1}}^\top = [\,1\ 1\ \ldots\ 1\,]$ and define the average position as

$$a(t) = \frac{\sum_{i=1}^{N} y_i}{N} = \frac{\bar{\mathbf{1}}^\top y}{N}. \tag{4}$$

Then we introduce a new vector $z(t)$ whose components are the differences $z_i = y_{i-1} - y_i$, $i = 2, \dots, N$:

$$\begin{bmatrix} z_2(t) \\ z_3(t) \\ \vdots \\ z_N(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_N(t) \end{bmatrix}, \quad (5)$$

which can be synthetically written as

$$z(t) = Dy(t), \quad (6)$$

where $D \in \mathbb{R}^{(N-1)\times N}$ is the matrix in Eq. (5). Note that the vector $[\ a(t)\ \ z^\top(t)\ ]^\top$, including the average position and the differences, is in one-to-one correspondence with $y(t)$.

Let us now define the Laplacian matrix $L \doteq D^\top D \in \mathbb{R}^{N\times N}$ and matrix $M \doteq DD^\top \in \mathbb{R}^{(N-1)\times(N-1)}$. Consider the Singular Value Decomposition of matrix D:

$$D = P[\ \bar{\mathbf{0}}_{\mathbf{N-1}}\ \Omega\ ]Q^\top, \quad (7)$$

where $\bar{\mathbf{0}}_{\mathbf{N-1}}$ is an all-zero vector of size $N-1$, matrix $\Omega \in \mathbb{R}^{(N-1)\times(N-1)}$ is diagonal and its positive diagonal entries are the singular values of D, while $P \in \mathbb{R}^{(N-1)\times(N-1)}$ and $Q \in \mathbb{R}^{N\times N}$ are orthonormal matrices [25], hence $P^\top P = I$ and $Q^\top Q = I$ (where I denotes the identity matrix of the appropriate size). Recall that the Laplacian matrix of a connected undirected graph has a single zero eigenvalue, while all other eigenvalues are real and positive. Then we can express our Laplacian matrix L as

$$\begin{aligned} L = D^\top D &= Q[\ \bar{\mathbf{0}}_{\mathbf{N-1}}\ \Omega\ ]^\top[\ \bar{\mathbf{0}}_{\mathbf{N-1}}\ \Omega\ ]Q^\top \\ &= Q\begin{bmatrix} 0 & \bar{\mathbf{0}}_{\mathbf{N-1}}^\top \\ \bar{\mathbf{0}}_{\mathbf{N-1}} & \Omega^2 \end{bmatrix} Q^\top \doteq Q\Lambda^2 Q^\top, \end{aligned} \quad (8)$$

where $\Lambda^2 \in \mathbb{R}^{N\times N}$ is a diagonal matrix whose diagonal entries are the eigenvalues of the Laplacian matrix L, given by 0 and the diagonal entries of Ω^2. Since it will be useful later in Section 3.4, we recall that the first column Q_1 of matrix Q is the normalized eigenvector associated with the zero eigenvalue of the Laplacian matrix, hence $Q_1 = \bar{\mathbf{1}}/\sqrt{N}$.

We can also write

$$M = DD^\top = P[\ 0\ \Omega\ ][\ 0\ \Omega\ ]^\top P^\top = P\Omega^2 P^\top, \quad (9)$$

hence the eigenvalues of matrix M are all the nonzero eigenvalues of L (i.e., the diagonal entries of Ω^2).

With a few algebraic manipulations, the overall system can be written in matrix form as

$$\ddot{y} = -kLy - hL\dot{y} - r\dot{y} + r\bar{\mathbf{1}}v(t) + \Delta, \quad (10)$$

where $\Delta = \begin{bmatrix}\delta_1 \dots \delta_N\end{bmatrix}^\top$ couples the control system with the network performance.

To derive the dynamics of the average position a, we pre-multiply Eq. (10) by $\bar{\mathbf{1}}^\top/N$:

$$\frac{1}{N}\bar{\mathbf{1}}^\top\ddot{y} = -\frac{k}{N}\bar{\mathbf{1}}^\top Ly - \frac{h}{N}\bar{\mathbf{1}}^\top L\dot{y} - \frac{r}{N}\bar{\mathbf{1}}^\top \dot{y} + \frac{r}{N}\bar{\mathbf{1}}^\top\bar{\mathbf{1}}v(t) + \frac{1}{N}\bar{\mathbf{1}}^\top\Delta \quad (11)$$

Then, since it can be shown that $L\bar{\mathbf{1}} = 0$ and $\bar{\mathbf{1}}^\top\bar{\mathbf{1}} = N$, we get

$$\ddot{a}(t) = -r\dot{a}(t) + rv(t) + \frac{1}{N}\bar{\mathbf{1}}^\top\Delta, \quad (12)$$

which does not depend on k and h, while it does depend on the average components $\Delta_{av} \doteq \frac{1}{N}\bar{\mathbf{1}}^\top\Delta$ of the disturbance.

To derive the dynamics of the differences z, let us pre-multiply Eq. (10) by matrix D and exploit the fact that $L = D^\top D$:

$$(D\ddot{y}) = -kDD^\top(Dy) - hDD^\top(D\dot{y}) - r(D\dot{y}) + rD\bar{\mathbf{1}}v(t) + D\Delta.$$

Then, since $D\bar{\mathbf{1}} = 0$ and $M = DD^\top$, we have

$$\ddot{z} = -kMz - hM\dot{z} - r\dot{z} + D\Delta. \quad (13)$$

Eq. (13) does not depend on the reference speed $v(t)$, which can be changed as needed, without altering the system dynamics or hampering safety. Also, Eq. (13) only depends on the disturbance component $D\Delta$ orthogonal to the average (the average disturbance Δ_{av} changes the average position: it moves all vehicles of the same amount and does not affect their spacing).

The overall system can now be analyzed by separately studying the evolution of Eq. (12) and of Eq. (13). Interestingly, the choice of the design parameters can be decoupled: r only affects the dynamics of the average a, while h and k only affect the dynamics of the differences z. In Section 3.2 we investigate the average properties of the platoon, while in Section 3.3 we explore the performance in terms of the differential dynamics. We also briefly discuss the error dynamics in Section 3.4.

3.2. The average dynamics

Analyzing the average platoon dynamics provides a design criterion for parameter r. To study the transient from zero speed to the desired speed $v(t)$, we consider the system in Eq. (12) with initial conditions $a(0) = \dot{a}(0) = 0$, meaning that the platoon is at rest in an initial position (assumed to be 0, without loss of generality). Its solution yields the average position

$$a(t) = vt - \frac{v}{r} + \frac{v}{r}e^{-rt}, \quad (14)$$

with average speed $\dot{a}(t) = v - ve^{-rt}$ and average acceleration $\ddot{a}(t) = rve^{-rt}$. The acceleration is maximal at the beginning and equal to $a_{max} = rv$. The time constant

$$\tau_a = \frac{1}{r} \quad (15)$$

can be selected by choosing r based on the trade-off between promptness and comfort. Increasing the value of r clearly speeds up the convergence, but has two counter-effects. First, a high r might cause undesirable accelerations, perceived as uncomfortable by the passengers [26]; so, the parameter r should guarantee that a_{max} is compatible with the comfort standards. Second, r determines the overall platoon convergence speed to new conditions and increasing it requires to increase the inter-vehicle distance for safety reasons; in fact, a faster overall response entails that, in presence of information loss, the intra-platoon errors are larger, which is one of the fundamental results of this paper (see Section 5).

3.3. The difference dynamics

A smooth average behavior of a platoon is important, but the dynamics of the differences z_i is fundamental for safety and group behavior: $z_i = d$ means that two vehicles are at the double of the desired distance d, while $z_i = -d$ means collision. The key design specification is therefore

$$|z_i| \leq \alpha d, \tag{16}$$

where $0 < \alpha < 1$ is a safety coefficient.

We show in this section that the system in Eq. (13) with $\Delta = 0$ is asymptotically stable: in the absence of disturbances, $z(t)$ converges to 0 as desired; in the presence of disturbances due to packet losses, instead, z can grow and must be kept bounded to prevent collisions. Indeed, in the following we will give bounds of the form $\|z\| \leq K$, where $\|z\| = \sqrt{\sum_i z_i^2}$ is the Euclidean norm of z; this implies that the bound on all distances is $|z_i| \leq K$.

In wireless ad-hoc networks, disturbances are essentially originated by packet losses, as delays are negligible. If a packet is not received by a vehicle, then there is a lack of information on the positions of the preceding and/or following vehicles. The typical (and, given the small beaconing time, the only reasonable) assumption in this case is that the vehicles are at the same distance with the same differential speed as the last transmitted information. The discrepancy between the actual relative position and speed and the estimated ones introduces a disturbance. Denoting by $\underline{y}_i$ the stale old information, Eq. (2) yields

$$\begin{aligned} \ddot{y}_i = &-k(y_i - \underline{y}_{i+1} - d) - k(y_i - \underline{y}_{i-1} + d) \\ &- h(\dot{y}_i - \dot{\underline{y}}_{i+1}) - h(\dot{y}_i - \dot{\underline{y}}_{i-1}) - r(\dot{y}_i - \underline{v}). \end{aligned} \tag{17}$$

Now define the error terms $\delta y_i = \underline{y}_i - y_i$, the derivative error $\delta \dot{y}_i = \dot{\underline{y}}_i - \dot{y}_i$, and $\delta v = \underline{v} - v$ to rewrite the dynamics as in (2), where the term δ_i is now re-defined as

$$\delta_i = h\delta\dot{y}_{i+1} + h\delta\dot{y}_{i-1} + k\delta y_{i+1} + k\delta y_{i-1} + r\delta v. \tag{18}$$

Equation (18) gives a clear criterion to co-design the constants h and k and the communication system to keep the error within safe boundaries. Once a packet loss has occurred, we can investigate how the system recovers after the occurrence and how the system behaves if the packet losses occur repeatedly in a burst leading to a potentially larger difference between the true information and the last received one.

To diagonalize the system, so that it is easier to study its stability, let us introduce the new variable

$$x = P^\top z, \tag{19}$$

where P is the orthonormal matrix such that $M = P\Omega^2 P^\top$, as in Eq. (9), where the diagonal entries of $\Omega^2 = \mathrm{diag}\{\Omega_1^2, \ldots, \Omega_{N-1}^2\}$ are the eigenvalues of M (i.e., the nonzero eigenvalues of L). Then, Eq. (13) can be rewritten as

$$\ddot{x} = -k\Omega^2 x - h\Omega^2 \dot{x} - r\dot{x} + \hat{\delta}, \tag{20}$$

with $\hat{\delta} = P^\top D\Delta$. Since P is orthonormal, it does not change the Euclidean norm: $\|x\| = \|P^\top z\| = \|z\|$.

If we apply the Laplace transform, with zero initial conditions, we have

$$X(s) = [s^2 I + (h\Omega^2 + rI)s + k\Omega^2]^{-1}\hat{\Delta}(s) = \Gamma(s)\hat{\Delta}(s), \tag{21}$$

where $\Gamma(s)$ is a diagonal matrix of transfer functions

$$\Gamma(s) = \mathrm{diag}\left\{\frac{1}{s^2 + (h\Omega_i^2 + r)s + k\Omega_i^2}\right\}. \tag{22}$$

The denominators of the transfer functions $\Gamma_i(s)$ are second order polynomials with positive coefficients, hence stability is ensured because their roots (the poles of the transfer functions) have a negative real part. With a suitable choice of the design parameters, we can rule out even damped oscillations. In fact, we can prove the following result.

Proposition 1. *The poles of the transfer functions $\Gamma_i(s)$ are real and negative if*

$$h > \frac{k}{r}. \tag{23}$$

Proof. The discriminants of the second order polynomials at the denominator of $\Gamma_i(s)$ are

$$\Delta_{G_i} = (h\Omega_i^2 + r)^2 - 4k\Omega_i^2 = h^2\Omega_i^4 + r^2 + 2rh\Omega_i^2 - 4k\Omega_i^2.$$

The roots of these polynomials are real provided that $\Delta_{G_i} > 0$. If $rh > k$, then

$$\Delta_{G_i} > h^2\Omega_i^4 + r^2 + 2rh\Omega_i^2 - 4rh\Omega_i^2 = (h\Omega_i^2 - r)^2 > 0.$$

■

We assume that Eq. (23) holds, hence all poles are real and negative, and we consider two problems:

1. The reaction of the platoon to an erroneous position of one of more vehicles (with no disturbances);
2. The reaction of the platoon to disturbances that are bounded in norm as $\|\hat{\delta}\| \leq \rho$.

For the first problem, we assume that $D\Delta = 0$ and that at some time ($t = 0$ without loss of generality) there is a mismatch in the position: $z(0) = z_0$, with zero speed. Then, we consider the Laplace transform: Since $\mathcal{L}[z(t)] = Z(s)$, $\mathcal{L}[\dot{z}(t)] = sZ(s) - z_0$ and $\mathcal{L}[\ddot{z}(t)] = s^2 Z(s) - sz_0$, from Eq. (13) we get

$$[s^2 I + (hM + rI)s + kM]Z(s) = [sI + (hM + rI)]z_0. \tag{24}$$

Since $X(s) = P^\top Z(s)$ and $x_0 = P^\top z_0$, while $I = PP^\top$ and $M = P\Omega^2 P^\top$, we get

$$\begin{aligned} X(s) &= [s^2 I + (h\Omega^2 + rI)s + k\Omega^2]^{-1}[sI + (h\Omega^2 + rI)]x_0 \\ &\doteq \Phi(s)x_0 = \mathrm{diag}\left\{\frac{s + (h\Omega_i^2 + r)}{s^2 + (h\Omega_i^2 + r)s + k\Omega_i^2}\right\}x_0. \end{aligned} \tag{25}$$

Then, the components of x evolve independently. Let us consider the inverse transform $\phi(t) = \mathrm{diag}\{\phi_i(t)\} = \mathcal{L}^{-1}[\Phi(s)]$. We have that $\phi_i(0) = 1$, from the initial value theorem ($\lim_{t\to 0} \phi_i(t) =$

$\lim_{s\to\infty} s\Phi_i(s)$). Hence $\phi(0) = I$. Moreover, all $\phi_i(t)$ are strictly decreasing, as can be shown by considering their derivative:

$$\begin{aligned}\mathcal{L}[\dot{\phi}_i(t)] &= s\Phi_i(s) - \phi_i(0)\\ &= s\frac{s + (h\Omega_i^2 + r)}{s^2 + (h\Omega_i^2 + r)s + k\Omega_i^2} - 1\\ &= \frac{-k\Omega_i^2}{s^2 + (h\Omega_i^2 + r)s + k\Omega_i^2}.\end{aligned} \tag{26}$$

This transfer function has real poles only, no zeros, and a negative coefficient at the numerator, so its inverse Laplace transform $\dot{\phi}_i(t)$ is negative [27, 28]. Hence, all $\phi_i(t)$'s are equal to 1 at $t = 0$ and converge to 0 for $t \to \infty$ (since the poles of the transfer function are real and negative). Therefore, they must be always positive and bounded as $\|\phi(t)\| \leq 1$ for all t. Hence, $|x_i(t)| < |x_{0,i}|$ for $t > 0$. Coming back to z, the inverse transform of $Z(s)$ is $z(t) = P\phi(t)P^\top z_0$. So, for a perturbation of size $\|z_0\|$,

$$\|z(t)\| = \|P\phi(t)P^\top z_0\| \leq \|\phi(t)\|\|z_0\| \leq \|z_0\|, \quad \text{for } t > 0. \tag{27}$$

The previous inequality ensures string-stability, namely, the propagation of the perturbation has effects that do not exceed in size the perturbation itself. Intuitively, string-stability means that the closed-loop system behaves as a transmission line with no reflections, where any propagating wave is suitably damped. Assume there is a misplacement (error) measured by $|z_i(0)| = \zeta$, then $\|z_0\| = \zeta$, this implies that $\|z(t)\| < \zeta$. Since the norm is greater or equal than the magnitude of any component, $|z_k(t)| \leq \zeta$ for all the components k, hence no component will exceed the initial size ζ. More formally:

Proposition 2. *If $z_i(0) = \zeta \neq 0$ and $z_j(0) = 0$ for $j \neq i$, then $|z_k(t)| \leq \zeta$ for all $t > 0$ and for all the components k.*

To determine the effect of a nonzero disturbance Δ, we can consider Eqs. (13) and (20) indifferently, since the transformation $P^\top$ is norm-preserving (the norms of z and of $x = P^\top z$ are equal). Consider Eq. (20) with $\|\hat{\delta}(t)\| \leq \rho$. Then, the transfer function is $\Gamma(s)$: $X(s) = \Gamma(s)\hat{\Delta}(s)$.

If we assume zero initial conditions and consider the inverse Laplace transform $\gamma(t) = \mathcal{L}^{-1}[\Gamma(s)]$, the solution is given by the convolution $x(t) = \int_0^t \gamma(\sigma)\,\hat{\delta}(t-\sigma)d\sigma$. Then

$$\begin{aligned}\|x(t)\| &= \left\| \int_0^t \gamma(\sigma)\ \hat{\delta}(t-\sigma)d\sigma \right\|\\ &\leq \int_0^t \|\gamma(\sigma)\|\|\hat{\delta}(t-\sigma)\|d\sigma \leq \rho \int_0^t \|\gamma(\sigma)\|d\sigma\\ &\leq \rho \int_0^\infty \|\gamma(\sigma)\|d\sigma = \rho \max_k \int_0^\infty |\gamma_k(\sigma)|d\sigma = \rho \max_k \int_0^\infty \gamma_k(\sigma)d\sigma.\end{aligned}$$

We removed the absolute value because $\gamma_k(\sigma)$ is a positive function. In fact, it has real poles only, no zeros and a positive coefficient at the numerator [27, 28]. The value of the integral can be computed by means of the final value theorem:

$$\int_0^\infty \gamma_k(\sigma)d\sigma = \left.\frac{1}{s^2 + (h\Omega_i^2 + rI)s + k\Omega_i^2}\right|_{s=0} = \frac{1}{k\Omega_i^2}.$$

This results in the bound

$$\|x(t)\| \leq \rho\frac{1}{k\Omega_1^2}, \tag{28}$$

where Ω_1^2 is the smallest eigenvalue of M (i.e., the smallest nonzero eigenvalue of L). Recall that $\|x(t)\| = \|z(t)\|$.

The error given by Eq. (18) scales with k, h and r, if we assume that v is fixed and exactly known. On the other hand, Eq. (23) is assumed to hold, hence $hr > k$. If we take $h/k = (1+\epsilon)/r$, with $\epsilon > 0$, the overall error scales linearly with k:

$$\begin{aligned}\|\delta_i\| &= k\left\| \frac{1+\epsilon}{r}\frac{d}{dt}\delta y_{i+1} + \frac{1+\epsilon}{r}\frac{d}{dt}\delta y_{i-1} + \delta y_{i+1} + \delta y_{i-1} \right\|\\ &\leq k\delta_{M_i},\end{aligned} \tag{29}$$

hence, since $\|\hat{\delta}\| = \|P^\top D\Delta\|$ and $\|D\| \leq 2$,

$$\|\hat{\delta}(t)\| \leq 2k\delta_M \doteq \rho, \tag{30}$$

where δ_M is a bound for the cumulative error of position and speed (according to some norm). Then, we get the bound

$$\|x(t)\| \leq \frac{2\delta_M}{\Omega_1^2}, \tag{31}$$

which depends uniquely on the eigenvalue Ω_1^2.

3.4. The error dynamics

We briefly discuss here the dynamics of the error vector

$$e = y - \bar{\mathbf{1}}\frac{\bar{\mathbf{1}}^\top y}{N} = \left[I - \frac{\bar{\mathbf{1}}\bar{\mathbf{1}}^\top}{N}\right]y,$$

whose ith component is the difference between y_i and the average of y:

$$e_i = y_i - \frac{\bar{\mathbf{1}}^\top y}{N} = y_i - \frac{1}{N}\sum_{j=1}^N y_j.$$

Consider Eq. (10) and pre-multiply it by $B \doteq [I - \frac{\bar{\mathbf{1}}\bar{\mathbf{1}}^\top}{N}]$. Since $BL = LB$ and $B\bar{\mathbf{1}} = 0$, we get

$$\begin{aligned}\ddot{e} &= B\ddot{y} = -kBLy - hBL\dot{y} - rB\dot{y} + Br\bar{\mathbf{1}}v(t) + B\Delta\\ &= -kLe - (hL + rI)\dot{e} + \Delta_{err},\end{aligned}$$

where $\Delta_{err} \doteq B\Delta$. Note that the evolution of the error variable does not depend on v.

We can show that the variance $\|e\|/\sqrt{N}$ is decreasing over time. Let e_0 be a nonzero initial condition and $\Delta_{err} = 0$. Then, in the Laplace transform domain,

$$[s^2 I + (hL + rI)s + kL]E(s) = [sI + (hL + rI)]e_0. \tag{32}$$

Adopting the decomposition in Eq. (8), we can write

$$E(s) = Q[s^2 I + (h\Lambda^2 + rI)s + k\Lambda^2]^{-1}[sI + (h\Lambda^2 + rI)]Q^\top e_0,$$

where $\Lambda^2 = \text{diag}\{0, \Omega^2\}$. Define the diagonal transfer function matrix

$$\Psi(s) = \text{diag}\left\{\frac{s + (h\Lambda_i^2 + r)}{s^2 + (h\Lambda_i^2 + r)s + k\Lambda_i^2}\right\},$$

whose first diagonal term is equal to $1/s$, because the first eigenvalue of the Laplacian L is $\Lambda_1 = 0$, while the other diagonal terms are the same as those in $\Phi(s)$:

$$\Psi(s) = \text{diag}\left\{\frac{1}{s}, \Phi(s)\right\}.$$

Denoting by $\psi(t)$ the inverse Laplace transform of $\Psi(s)$, we get

$$e(t) = Q\psi(t)Q^\top e_0. \tag{33}$$

Note that $\psi(t)$ has the same diagonal entries as $\phi(t)$ and an extra diagonal entry (the first) that is equal to 1: $\psi_{11}(t) = 1$. Hence, we can conclude that $\|e(t)\|$ is non-increasing. Now, we observe that an initial condition e_0 is meaningful if it has 0 mean: in fact, in view of the error definition, its mean must be

$$\frac{\bar{\mathbf{1}}^\top}{N} e = \frac{\bar{\mathbf{1}}^\top}{N} y - \bar{\mathbf{1}}^\top \bar{\mathbf{1}} \frac{\bar{\mathbf{1}}^\top y}{N^2} = 0,$$

since $\bar{\mathbf{1}}^\top \bar{\mathbf{1}} = N$. Also, the first row of $Q^\top$ is equal to $\bar{\mathbf{1}}^\top / \sqrt{N}$, the eigenvector associated with the 0 eigenvalue of L, hence the constant term associated with the mode $\psi_{11}(t) = 1$ disappears in Eq. (33). This proves that the variance $\|e(t)\|/\sqrt{N}$ is indeed decreasing and asymptotically converges to 0 (for $t \to \infty$).

4. Actuator dynamics, delays and asymmetric control

We briefly discuss here some potential extensions of our model. In particular, we have assumed that the transmission delay τ_{dl} is small enough compared to the time scale of vehicle dynamics. We have also assumed homogeneity in the vehicles. This assumption is legitimate as long as *acceleration* is the control input. In the case of non-homogeneous vehicles, the practical implementation of our scheme simply requires a low-level control loop to actuate the desired acceleration profile. However, this loop may have an actuation lag associated with a time constant τ_{act}. We can explicitly consider both the transmission delay τ_{dl} and the actuation lag τ_{act} in the system dynamics in Eq. (17), thus obtaining the Laplace-transformed expression:

$$\begin{aligned} s^2 y_i &= \frac{e^{-\tau_{dl}s}}{1+\tau_{act}s}\left[-k(y_i - y_{i+1}) - k(y_i - y_{i-1})\right. \\ &- \left. sh(y_i - y_{i+1}) - sh(y_i - y_{i-1}) - r(sy_i - v) + \delta_i(s)\right] \end{aligned} \tag{34}$$

Adopting *exactly* the same algebraic steps as before, we find

$$\Gamma(s) = \text{diag}\left\{\frac{\frac{e^{-\tau_{dl}s}}{1+\tau_{act}s}}{s^2 + \frac{e^{-\tau_{dl}s}}{1+\tau_{act}s}(h\Omega_i^2 + r)s + \frac{e^{-\tau_{dl}s}}{1+\tau_{act}s}k\Omega_i^2}\right\} \tag{35}$$

of which (22) is a special case for $\tau_{dl} = 0$ and $\tau_{act} = 0$. Then we can easily check the level of delay and of lag tolerated by the network to avoid instability, by analyzing the non-rational transfer functions in (35) adopting standard control tools, such as the Nyquist diagram.

Another interesting extension is the asymmetric control action suggested in [29, 30, 31], where each vehicle "trusts" the preceding vehicle more than the following one (or the other way round). We can enforce this asymmetry by introducing a positive factor λ^2 in the system equations:

$$\begin{aligned} \ddot{y}_i &= \left[-\lambda^2 k(y_i - y_{i+1}) - k(y_i - y_{i-1})\right. \\ &- \left. \lambda^2 h(\dot{y}_i - \dot{y}_{i+1}) - h(\dot{y}_i - \dot{y}_{i-1}) - r(\dot{y}_i - v) + \delta_i(s)\right] \end{aligned}$$

This case can be handled in our proposed setup: if we consider again $z_i = y_{i-1} - y_i$ and $d = 0$, we get a system of the same form as (13), where now, unfortunately, M is non-symmetric in general. However, to fix this issue, it is enough to scale the variables: we redefine z_i as $z_i = \lambda^{i-1}(y_{i-1} - y_i)$ and we get

$$\begin{aligned} \ddot{z}_i &= -k\left[-\lambda z_{i+1} + (1+\lambda^2)z_i - \lambda z_{i-1}\right] \\ &- h\left[-\lambda \dot{z}_{i+1} + (1+\lambda^2)\dot{z}_i - \lambda \dot{z}_{i-1}\right] - r\dot{z}_i + \tilde{\delta}_i(s) \end{aligned}$$

which has the same form as (13), with M still being a symmetric positive definite matrix. All our considerations then hold unchanged in this more general case.

5. Mapping Packet Losses to Error Bounds

In an ad-hoc mobile network, the loss of packets is by far the major source of disturbance: Delays are negligible with direct communications, and sensor errors are limited. The loss of consecutive packets instead means that the controller is "blinded" for hundreds of milliseconds. Let N_L be the maximum number of consecutive losses (burst) than can occur in the channel with a certain probability bound. Above this value the network is faulty, and the system should enter a disaster recovery phase, which is out of the scope of this paper. Here we do not focus on a specific technology: our contribution is technology independent, and it is not important whether we are using IEEE 802.11p, 5G Cellular-V2X, or Visible Light Communication (VLC) (or a combination of them, as envisioned in [32]): Depending on the technology we will have different values for N_L, but this does not affect the analysis.

For the worst-case analysis we want to compute the bound imposed by the loss of N_L consecutive packets on the disturbance term δ_i. We consider the error in Eq. (18). The error is expressed as the sum of the position, speed, and reference speed errors multiplied by their coefficients. With respect to the position and the speed error, the upper bound can be computed by considering the maximum jerk $\bar{J}$ (the derivative of acceleration) a vehicle can implement. We compute the bounds on position and speed error as

$$\bar{\delta}^{\dot{y}} = \int_0^{(N_L+1)T} \int_0^t \bar{J} dt\, dt = \frac{\bar{J}}{2}((N_L+1)T)^2 \tag{36}$$

$$\bar{\delta}^{y} = \int_0^{(N_L+1)T} \int_0^t \int_0^t \bar{J} dt\, dt\, dt = \frac{\bar{J}}{6}((N_L+1)T)^3, \tag{37}$$

where T is the packet transmission interval. With respect to the reference speed error, the bound depends on how much the reference can change. In cruising conditions sharp changes of the reference are not needed and we set a maximum allowed

change in reference speed named $\bar{v}$ between consecutive packets. Combining Eqs. (18), (36) and (37) yields the error bound

$$\delta_M = 2\left(h\frac{\bar{J}}{2}T_{NL}^2 + k\frac{\bar{J}}{6}T_{NL}^3\right) + r\bar{v}\cdot(N_L+1). \tag{38}$$

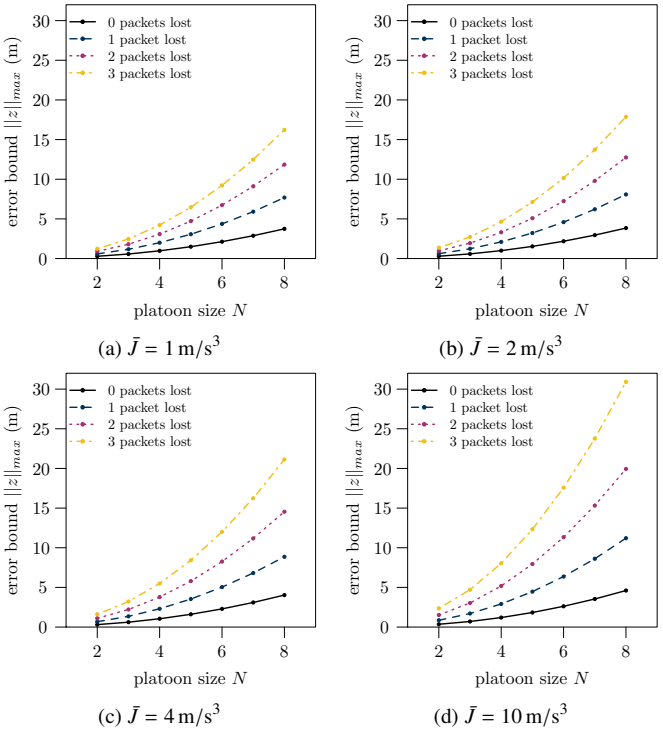
where $T_{NL} = ((N_L+1)T)$. It is necessary to double the position and speed error bounds to consider both preceding and following vehicles. Finally, to compute the maximum possible error, we consider the smallest non-zero eigenvalue Ω_1^2 of $L = D^\top D$, computed using the singular value decomposition of matrix D and exploiting the fact that $\|z\| \leq \frac{2\delta_M}{\Omega_1^2}$, in view of Eq. (31) and of the fact that $\|x\| = \|z\|$. Note that the value Ω_1^2 depends on the number of vehicles: The larger the number of vehicles, the smaller Ω_1^2. Finally, we set the inter-vehicle distance to

$$d > \frac{2\delta_M}{\Omega_1^2}c_s, \tag{39}$$

where $c_s \geq 1$ is a safety coefficient. Since $\frac{2\delta_M}{\Omega_1^2}$ represents the upper bound on the error, Eq. (39) represents a lower bound on the distance to guarantee 100 % safety under the given network and vehicle dynamics constraints. This bound is fully consistent with the bound (17) in [31], which also involves the smallest non-zero eigenvalue of the Laplacian matrix: for large platoons, large errors δ_M due to communication faults require a larger distance margin. The bound we found, as the one in [31], can probably be reduced with heuristics and with model-predictive tools; however, as our bound refers in this case to the complete lack of information, for instance due to jamming, and not to the loss of information at individual receivers, we believe that the correct way to design the system is through the introduction of redundant communication technologies.

Figure 2 plots the bound $\|z\|_{max} = \frac{2\delta_M}{\Omega_1^2}$ and, thus, the minimum safety distance d as a function of the platoon size N, for different maximum jerks $\bar{J}$ and number of consecutive losses N_L. The remaining parameters are fixed: $T = 100\,\text{ms}$, $\bar{v} = 1\,\text{km/h}$ per packet[1], $k = 0.5$, $h = 0.71$, $r = 1$. The choice of 100 ms is common within platooning control literature, both in research and in real world experiments [1]. The platoon size N has the largest impact, as the bound grows more than linearly with N. The parameters N_L and $\bar{J}$ also play a significant role, but the impact is not as large. In good network conditions the control system is definitely performing well, as the worst-case upper bound is below 3 m even with 8 vehicles. In non-ideal network conditions, instead, there is an important trade-off in the choice of the parameters. To have small inter-vehicle distances, we either need to ensure a high network reliability (thus, a low N_L) or limit the size of the platoon. Indeed, this allows the easy regulation of d and its dynamic adaptation to the network conditions. Otherwise, the performance of the vehicle can also be considered and, if needed, altered for system tuning. For example, by limiting the maximum jerk to $4\,\text{m/s}^3$ the system can

[1]This corresponds to 10 km/h per second with the given T, which is much more than the normal speed change we expect while cruising.



(a) $\bar{J} = 1\,\text{m/s}^3$ (b) $\bar{J} = 2\,\text{m/s}^3$

(c) $\bar{J} = 4\,\text{m/s}^3$ (d) $\bar{J} = 10\,\text{m/s}^3$

Figure 2: Error bound $\|z\|_{max}$ as function of the platoon size N, for different maximum jerks $\bar{J}$ and burst size N_L.

maintain a relatively small distance while being robust to heavy packet losses. It is important to remember that the bound $\|z\|_{max}$ is computed as a worst-case which, in reality, might never occur: N_L packets per vehicle are "jammed" so that no vehicle receives any information for $(N_L+1)T$ s. In Section 7, we show that the norm of the distance errors in realistic conditions is much smaller than the bound $\|z\|_{max}$.

6. Handling External Events

The design of the controller considers standard cruising only, which is the main purpose of a platooning control algorithm. A platoon, however, is also required to react to emergencies and external events. One example is an emergency braking maneuver [33], but an external event might not necessarily trigger an emergency mechanism. For instance, the platoon might encounter a slower vehicle upfront or it might receive a speed advise from the road infrastructure. Both examples require a deceleration, although we can refer to an "emergency maneuver" only when the deceleration is perceived as uncomfortable by a passenger (more than $4\,\text{m/s}^2$ [26]).

Differently from conventional CACC systems, where the leader is controlled by an independent law, our design controls leader's behavior as well. If we are required to change the cruising speed in reaction to an external input, setting the reference speed v to such value might not be enough, as the algorithm smoothly (i.e., exponentially) converges to the desired speed.

Proper handling of external events depends on the event itself. Here we consider three instances. The first one is an emergency braking maneuver, where the platoon simply needs to come to a complete stop as quickly as possible. The second

one is maintaining a safe distance to a vehicle in front. The last one is adapting the cruising speed within a given distance after a road infrastructure speed advise.

6.1. Emergency Braking

As discussed, to realize an emergency braking, setting the reference speed $v = 0$ is not enough, as the algorithm smoothly converges to the desired speed with a comfortable deceleration and not in "emergency mode". To realize an emergency braking maneuver we thus need to modify controller parameters "on the fly", in particular by acting on the desired speed v and the vehicle-reference friction coefficient r. Let us assume that the vehicle initiating the maneuver is traveling at speed v_0. To implement the maneuver, we introduce a new control mode, namely the "override mode", in which the leader (or any vehicle in the platoon) follows a desired acceleration towards a reference speed instead of the one computed by our proposed control system. In this mode, given a desired acceleration $\ddot{y}_{\text{des}}$ and a desired speed $\dot{y}_{\text{des}}$, the parameter r is continuously adapted using the following formula:

$$r = \left| \frac{\ddot{y}_{\text{des}}}{\dot{y}_i - \dot{y}_{\text{des}}} \right|. \tag{40}$$

In addition, the reference speed v is set to $\dot{y}_{\text{des}}$. The vehicle requesting the override continuously updates the value of r and broadcasts it, together with the desired speed $\dot{y}_{\text{des}}$, to other vehicles. In the same beacon, the vehicle also advertises the override command, so the others can use the values r and $\dot{y}_{\text{des}}$ inside the message instead of the default ones. Notice that the override mode might cause the violation of the minimum safe distance $\|z\|_{max}$, as we are changing the parameter r and the reference speed v by an amount that can potentially be larger than $\bar{v}$. This mode needs thus to be used only for small periods of times in case of emergency situations.

Implementing an emergency braking with a given deceleration $\ddot{y}_{\text{des}}$ simply requires setting $\dot{y}_{\text{des}} = 0$ in Eq. (40), thus setting $r = \frac{\ddot{y}_{\text{dec}}}{\dot{y}_i}$.

6.2. Maintaining a Safe Distance

Monitoring the presence of other vehicles requires the leader of the platoon to analyze the surrounding environment with onboard sensors, such as radars or LiDARs. These sensors detect objects within their detection range together with their distance d and relative speed $\dot{d}$. Normally, the leader is controlled by an independent control law that exploits such information and implements, for example, an ACC. To control the leader, we can run an ACC in parallel and use its computed acceleration $\ddot{y}_{acc}$ only if required.

In particular, we can consider two cases. Let $\dot{y}_f$ be the speed of the vehicle detected by leader's sensors. In the first case, the vehicle ahead is slower but, progressively reducing the reference speed v to $\dot{y}_f$ without violating the rate of change $\bar{v}$ (in km/h per packet) is enough to maintain the proper safety distance. This permits us to use the proposed control law without violating the error bound $\|z\|_{max}$.

In the second case, instead, the reaction of the proposed control law can not bring the platoon into a safe situation quickly enough. This can be the result, for example, of a vehicle cutting in, i.e., changing lane without respecting the backward safety distance. Another example is when the platoon approaches a much slower vehicle. In this case, the acceleration value $\ddot{y}_{acc}$ of the ACC system needs to be used instead. In particular, $\ddot{y}_{acc}$ can be plugged in the formula used for the emergency braking case, together with the front vehicle speed $\dot{y}_f$.

The reaction of the proposed control system to a slower vehicle ahead requires to change the reference speed v. For this reason, we implement the following simple control loop:

$$\dot{y}_{\text{target}} = \min(\dot{y}_f, \dot{y}_{des}) \tag{41}$$

$$v = v + \min(\bar{v}, \max(-\bar{v}, \dot{y}_{\text{target}} - v)). \tag{42}$$

Equation (41) computes the speed the platoon should converge to, i.e., the minimum between the speed of the vehicle ahead ($\dot{y}_f$) and the desired cruising speed ($\dot{y}_{des}$). Equation (42) instead causes the reference speed v to approach the target, but the rate of change of v can not be larger than $\bar{v}$. To respect the limit of $\bar{v}$ km/h per beacon, Eq. (42) sampling frequency must be the same as the beacon rate.

Finally, we need to compute both the proposed control algorithm and an ACC in parallel to obtain two acceleration values. The acceleration to be actuated is simply chosen as the minimum between the two. If the ACC acceleration is chosen, then the system needs to use the override mode defined in Section 6.1. The condition which triggers the override mode is

$$\ddot{y}_{acc} < \ddot{y}_0 - \epsilon. \tag{43}$$

This condition avoids continuously switching between standard and override mode when the acceleration values are comparable. In here, we set $\epsilon = 0.5\,\text{m/s}^2$.

The ACC control algorithm we consider here is the one defined in [16], that is:

$$\ddot{y}_{acc} = -\frac{1}{T}\left(\dot{y} - \dot{y}_f + \lambda\,(T\dot{y} - d)\right). \tag{44}$$

Equation (44) implements a proportional-derivative control with a time headway spacing policy. In particular, $T = 1.2\,\text{s}$ is the time headway, d is the radar-measured distance to the front vehicle, and λ a design parameter set to 0.1.

6.3. Infrastructure-based Speed Advises

In this scenario we assume that the infrastructure can communicate to the platoon a change in the cruising speed for safety or traffic smoothness reasons. In addition, we assume that the infrastructure also mandates that the change in speed should be realized in a given distance y_{des}. Given the constraints, we can compute the constant acceleration required to change the cruising speed. In particular, given the current cruising speed v and the desired cruising speed $\dot{y}_{des}$, the time and the distance required to perform the maneuver with an acceleration $\ddot{y}$ are simply

$$t = \frac{\dot{y}_{des} - v}{\ddot{y}}, \quad y_{des} = \frac{\dot{y}_{des} + v}{2}t. \tag{45}$$

By combining the equations, we obtain

$$\ddot{y} = \frac{\dot{y}_{des}^2 - v^2}{2y_{des}}. \tag{46}$$

The proposed control system can adjust the cruising speed by an amount which is at most $\bar{v}$ per beacon. The resulting acceleration is thus $\bar{v} \cdot b_r$, where b_r is the beacon rate. If $|\bar{v} \cdot b_r| \leq |\ddot{y}|$, then the system is required to use the override mode. The absolute value accounts for speed advises mandating an increase of the cruising speed, so a positive acceleration. If instead $|\bar{v} \cdot b_r| > |\ddot{y}|$, the reference speed can be adjusted by $\frac{\ddot{y}}{b_r}$ every beacon. This does not violate the maximum change in reference speed $\bar{v}$ per beacon and satisfies the distance requirement y_{des}. In fact, the distance requirement might be violated by a small amount due to vehicle dynamics, as the control law does not consider the actual amount of driven meters. This implies that the acceleration computed when the speed advise is received is not re-evaluated but simply kept constant until the speed reaches the target.

7. Performance Evaluation

We implement the proposed control system in the platooning simulator Plexe [34], which allows testing the performance of platooning control algorithms under realistic vehicle dynamics and communication models. It is especially valuable for assessing implementation-related issues as, e.g., the effect of asynchronous packets' transmission times. As the data exchange rate (10 Hz) between vehicles is slower than the actuation control loop (100 Hz [1]) and vehicles might not be synchronized, the data provided to the algorithm might be incoherent from a time perspective. As an example, the own GPS position might be up to date, while the position of the front and back vehicles is "frozen" to the value included within the last received beacon.

To cope with this issue the control system includes a predictor, which computes missing values by interpolation. More formally, assume that $\ddot{y}_{t_0}$, $\dot{y}_{t_0}$, and y_{t_0} are the acceleration, speed, and position of a vehicle at time t_0. To estimate speed and position of such vehicle at time t, the control system computes

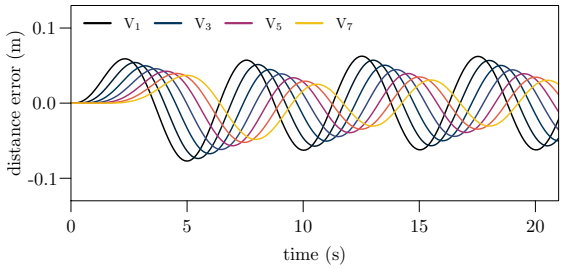
$$\dot{y}_t = \dot{y}_{t_0} + \ddot{y}_{t_0}(t - t_0), \quad y_t = y_{t_0} + \frac{t - t_0}{2}(\dot{y}_t + \dot{y}_{t_0}). \tag{47}$$

The use of Eq. (47) makes Plexe simulation extremely realistic as this is what on-board controllers are expected to do.

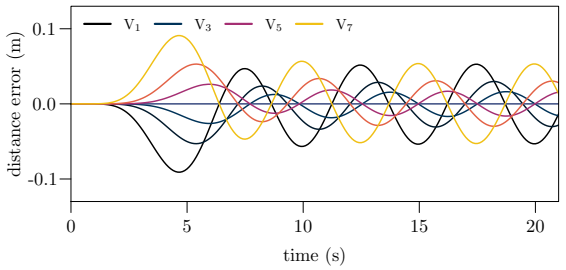
7.1. Error Dynamics Comparison

We first show the dynamics of the vehicles without network impairments. The goal is to understand the behavior of the controller, which is qualitatively different from the solutions proposed in the literature. We compare our algorithm with the controller designed in [1], which is a well-known CACC using a time headway spacing policy.

Figure 3 shows the distance error dynamics between vehicles V_i and V_{i-1} for a platoon of 8 cars under a sinusoidal disturbance. For the CACC designed by Ploeg et. al. [1], the leader changes its speed following the sinusoidal pattern, while



(a) CACC in [1]



(b) Our proposal

Figure 3: Qualitative comparison between a classic algorithm and the proposed solution (distance errors under a sinusoidal disturbance).

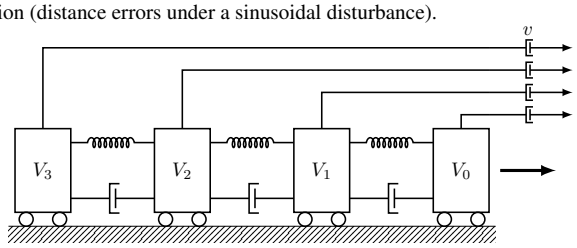


Figure 4: Spring-damper representation of the proposed control system.

for our controller we change the reference speed v. Figure 3a shows the classical attenuation of the error dynamics towards the tail of the platoon, thanks to the string-stability property. Our approach (Fig. 3b) is string stable as well, but the maximum attenuation occurs at the middle of the platoon and the dynamics are symmetric with respect to the center.

We can make an analogy between our algorithm and a spring-damper system (Fig. 4). We can imagine that consecutive vehicles are connected through a spring-damper, and an additional damper representing the reference speed v. The controller coefficient k refers to the spring elastic module, h is instead the damping factor between cars, while r describes the rigidity of the dampers connecting cars to the "virtual body" that moves at speed $v(t)$, thus setting the platoon reference speed. When changing the reference speed the vehicles are pushed back/pulled forward all at the same time, and the "inner" springs take care of attenuating the internal errors. A non trivial consequence of this controller structure is that position errors are compensated balancing the control effort between the front and rear vehicle, while in most other controllers the effort is all on the rear vehicle. This is in line with the "philosophy" of an autonomous driving platoon and not of a human-driven vehicle followed by partially automated vehicles. Further discussion on this topic is beyond the scope of this paper.

7.2. Error Bound Analysis.

As a second analysis we perform a set of simulations to empirically show that the error bound computed in Section 3.3 is always respected. To achieve this, we implement a scenario where the leader vehicle continuously changes the reference speed v by an amount $\bar{v}$ for each packet (i.e., every T seconds). In addition, we consider a channel causing burst losses at the receivers. In particular, each received packet has a certain probability of triggering a burst of losses. If a burst is triggered, the vehicle discards all the incoming packets received until the time $n_L T$ has elapsed, loosing n_L consecutive packets for each vehicle. n_L is drawn from a discrete uniform distribution $\mathcal{U}(1, N_L)$. After the end of a burst, each receiver waits a minimum amount of time before starting the next one. The analysis on the bound is indeed valid when considering the system at steady state. After a burst of losses, the system needs a certain amount of time to converge (cf. Eq. (15)) to eliminate the accumulated error. However, we also consider very small network up-times (as small as 100 ms) to show the robustness of our approach. Finally, we consider a first order actuation lag with a time constant $\tau = 0.5$ s, i.e., the response of the engine and the braking system to actuation commands $\ddot{y}$ is modeled by the transfer function $\ddot{y}_{\text{real}} = \frac{1}{\tau s + 1}\ddot{y}$, which is a common assumption confirmed by field operational tests [1, 16, 18, 17]. In the analysis we consider homogeneous vehicles, i.e., τ is equal for all the vehicles. The extension to non-homogeneous vehicles is possible, as discussed in Section 4. In practice, the same strategy can be implemented directly also on a string of non-homogeneous vehicles, provided that τ is large enough to account for the dynamics of the less-performing vehicle [1]. Table 2 summarizes simulation parameters. Even though the burst loss process to simulate interference is synthetic, the underlying network model emulates a DSRC link using the IEEE 802.11p and 1609.4 standards.

For each simulation s, we compute the norm of the error vector as

$$\|z_s\| = \max_k \sqrt{\sum_{i=1}^{N} (d_{k,i} - d)^2}, \tag{48}$$

where $d_{k,i}$ is the distance between vehicles V_i and V_{i-1} at simulation step k and d is the target distance. We then verify that $\|z_s\| \leq \|z\|_{max}$ for all the simulations, where $\|z\|_{max}$ is the theoretic bound for the norm, computed upon the parameters chosen for that particular simulation.

In the computation of the theoretic bound, however, the maximum jerk $\bar{J}$ is not clearly defined. In the real world it can either be a physical limit of the engine or the braking system, or a design parameter. In the simulations there is no such limit. For this reason, we post-analyze the maximum jerks obtained in the simulations. Figure 5 shows an histogram of the maximum jerk value of each simulation. Small maximum jerks (1.5 m/s^3 to 3.5 m/s^3) occur when packet loss events are unlikely and for small values of the r parameter. Recall that r balances the trade-off between settling time and driving comfort, so a higher value is more likely to cause large acceleration changes. Medium jerk values (5.5 m/s^3 to 8 m/s^3) are caused by a large value of the r parameter ($r = 4$), or a small r value combined with moderate packet losses. Finally, heavy losses cause large maximum jerk values, as the system obtains control data after long periods of silence, requiring strong actions to compensate the error. To compute the theoretic error bounds we use the minimum of the values shown in Fig. 5, i.e., 1.5 m/s^3.

Parameter	Value
k, h, T, τ	0.5, 0.71, 100 ms, 0.5 s
r	$\sqrt{0.5}$, 1, 4
n_L	1, $\sim \mathcal{U}(1,3)$, $\sim \mathcal{U}(1,5)$
Start burst probability	1, 5, 10, 20, 30, 40, and 50 %
Minimum no-burst time	0.1 s, 0.3 s, 0.5 s, 1 s, and 3 s
$\bar{v}$	1 km/h per packet
Repetitions	10
Path loss model	Free space ($\alpha = 2.0$)
PHY model	IEEE 802.11p
MAC model	1609.4 single channel (CCH)
Frequency	5.89 GHz
Bitrate	6 Mbit/s (QPSK R = 1/2)
Access category	AC_VI
MSDU size	200 B (byte)
Transmit power	20 dBm

Table 2: Simulation parameters.

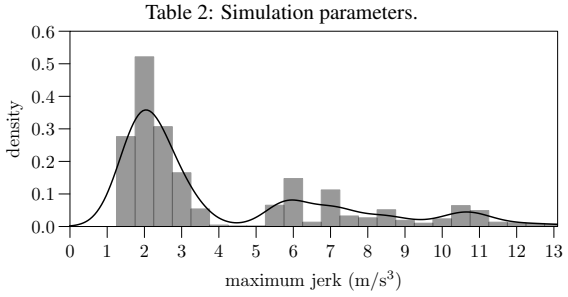


Figure 5: Distribution of maximum jerks measured over all simulation runs.

Figure 6 plots the simulation and theoretic bounds for different combinations of the r and N_L parameters. Simulation bounds are marked with points, theoretic bounds with crosses. The graph clearly shows that the theoretic bounds are respected. The margin between simulation and theory is large and this is due to two facts.

First, the bound $\|z\|_{max}$ is computed on the worst case: A change in the reference speed, a burst loss of N_L packets, and a change in the dynamics with the maximum jerk should occur at the same time. This is very unlikely even in a synthetic sce-

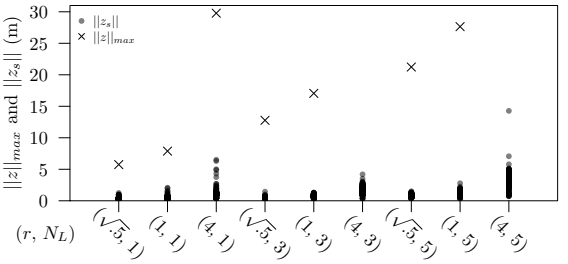


Figure 6: Plot of the simulation ($\|z_s\|$) and the theoretic ($\|z\|_{max}$) bounds, for different combinations of the r and N_L parameters. The $\|z\|_{max}$ values for $(r, N_L) = (4, 3)$ and $(4, 5)$ are out of scale and are not shown for the sake of clarity.

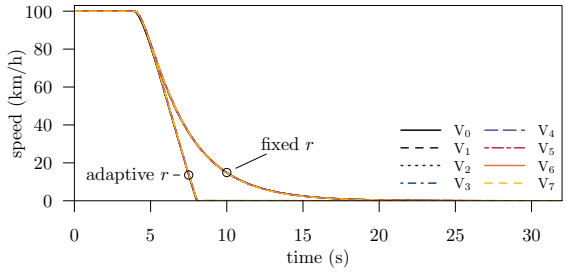


Figure 7: Comparison of the speed dynamics when setting $v = 0\,\text{km/h}$ from $v = 100\,\text{km/h}$ with and w/o adaptive r.

nario like the one we consider, especially because the jerk is a consequence of the control action computed by the algorithm.

Second, the predictor implemented within the control system counteracts the effects of packet losses, estimating the position and the speed of other vehicles during network down time. The effectiveness of the predictor is evident, as the impact of the burst length is smaller compared to the impact of r.

7.3. Emergency Braking

In this and in the following sections we analyze the behavior of our control system in the presence of an external input, as described in Section 6.

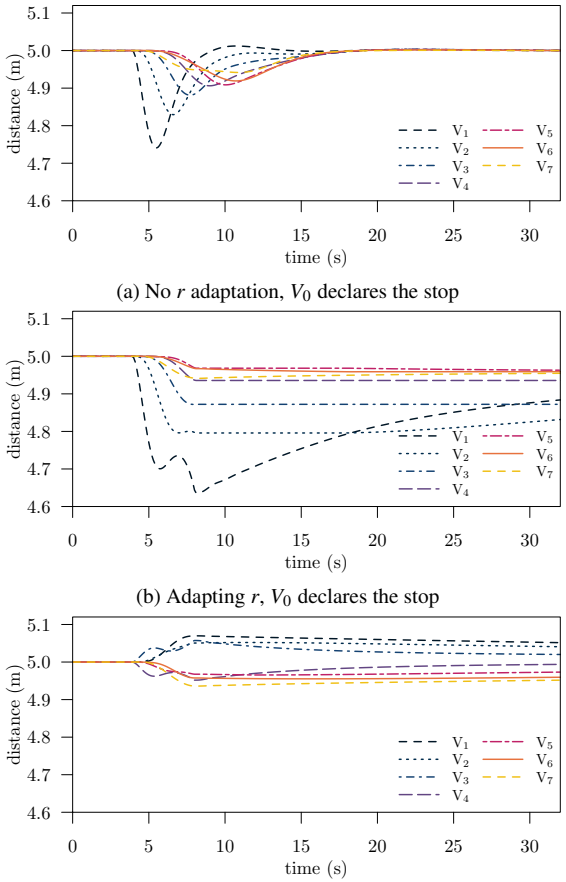
Figure 7 shows the behavior of the system in an emergency braking maneuver, comparing the standard cruising mode (i.e., simply setting the reference speed $v = 0\,\text{km/h}$) and the override mode for a desired deceleration of $8\,\text{m/s}^2$. When the leader sets the reference speed $v = 0$ but does not adapt r, the platoon takes 15 s to come down to a complete stop, while when r is adapted to the situation of a sudden unforeseen stop the platoon comes to a complete stop in 3 s to 4 s. The average behavior is always smooth and depends only on how r is changed.

Figure 8 shows the differential dynamics of the maneuver in terms of relative vehicles distance in the same conditions of Fig. 7 in three different conditions: Without adapting r (Fig. 8a); adapting r (Fig. 8b); and adapting r when the maneuver is initiated by the fourth vehicle in the platoon V_3 and not by the first one V_0 as usual (Fig. 8c). As expected, dynamically changing r allows a faster deceleration, but ends in a larger spacing error, that remains in any case in the order of tens of cm. Interestingly, if the stop is declared by a vehicle in the middle of the platoon, a feature this controller enables, distance errors are smaller. After the platoon comes to a complete stop, the vehicles keep moving very slowly to bring the inter-vehicle distance exactly to d, but these are movements of centimeters and vehicles can be conveniently stopped at any distance if desired.

One observation to make in this scenario is that the theoretic bound $\|z\|_{max}$ is not valid during the emergency maneuvers, as the parameters of the controller change and the scenario is no more a standard cruise, but an emergency stop. The platoon, however, remains very stable and distances, as shown by results, remain well within safety, and indeed within the "cruising bound", even if it is not theoretically valid.

7.4. Maintaining a Safe Distance

To show the reaction of the proposed control system to "external" vehicles we add one additional car in our simulation that



(a) No r adaptation, V_0 declares the stop

(b) Adapting r, V_0 declares the stop

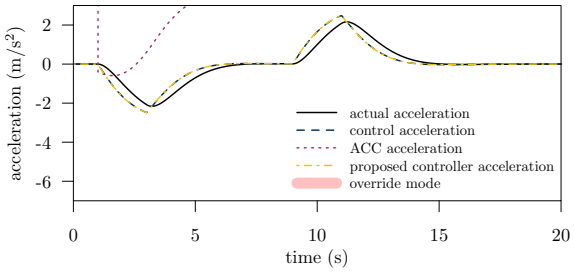
(c) Adapting r, V_3 declares the stop

Figure 8: Comparison of the relative vehicles' position when setting $v = 0\,\text{km/h}$ from $v = 100\,\text{km/h}$ without adapting r (a), adapting r (b), and adapting r (c) when the stop is declared by V_3.

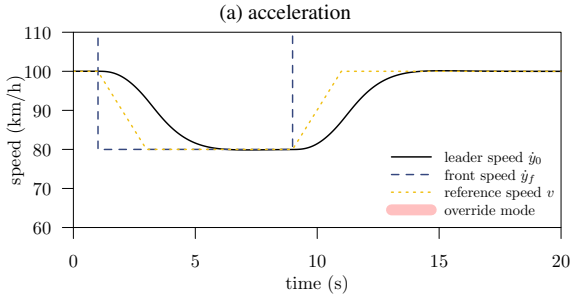
performs a cut in maneuver. The car performs the maneuver at different distances ahead of the platoon and relative speeds. In particular the cut in vehicle performs the maneuver 50 m, 100 m, 200 m, and 300 m ahead of the platoon and with a negative relative speed of 10 km/h, 20 km/h, 30 km/h, and 40 km/h, i.e., the cut in vehicle travels slower than the platoon. After a certain amount of seconds, the cut in vehicle leaves the lane, so the platoon is free to accelerate to the desired cruising speed.

For the sake of brevity we show the results for the three most significant scenarios. In particular, Figs. 9 to 11 show the acceleration and the speed of the leading vehicle. Acceleration plots show the proposed controller and ACC computed accelerations (control inputs), as well as the one chosen between the two (control acceleration). In addition they show the actual acceleration (i.e., post actuation dynamics). Speed plots show the speed of the front vehicle $\dot{y}_f$, which is set to infinity if no vehicle ahead is detected, the reference speed v of the proposed control system, and the actual leader speed $\dot{y}_0$. If the override mode is used, both plots show a shaded box indicating the amount of time the mode was active.

In all the simulation scenarios, the cut in vehicle changes lane and moves in front of the platoon after 1 s of simulation time, and leaves the lane after 9 s. On speed plots, this is shown by two vertical lines, representing the change in the speed de-

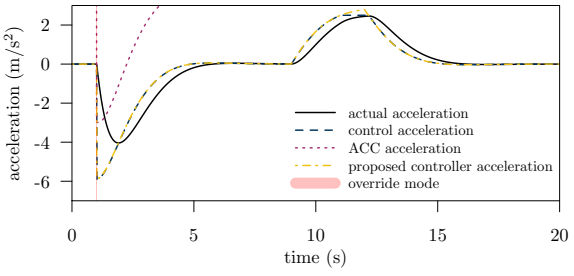


(a) acceleration

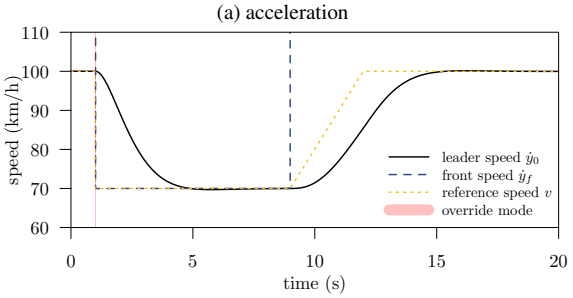


(b) speed

Figure 9: Acceleration and speed plots showing the response to a slower vehicle cut in maneuver for a distance of 100 m and a relative speed of 20 km/h. In this scenario the override mode was never triggered.



(a) acceleration



(b) speed

Figure 10: Acceleration and speed plots showing the response to a slower vehicle cut in maneuver for a distance of 100 m and a relative speed of 30 km/h. In this scenario the override mode was never triggered.

tected by the radar. For example, in Fig. 9b, at 1 s the front vehicle speed $\dot{y}_f$ changes from infinity (no vehicle ahead) to 80 km/h, while at 9 s, $\dot{y}_f$ changes from 80 km/h to infinity, indicating that the cut in vehicle left the lane and it is no more ahead of the platoon.

The first scenario (Figs. 9a and 9b) shows the case where the smooth adaptation of the reference speed v is enough to respect the safety gap. In Fig. 9a, this is shown by the fact that ACC acceleration is larger than the one computed by our proposed control algorithm. The fact that the ACC acceleration becomes positive is simply because the algorithm tries to converge to the exact safety distance. Given that the deceleration imposed by our control system causes the safety distance to be larger than required, the ACC outputs a positive acceleration. In the speed plot (Fig. 9b), it can be seen that the reference speed is smoothly decreased and converges to the front vehicle speed $\dot{y}_f$. At 9 s, when the cut in vehicle leaves the platoon lane, the leader accelerates to converge to the desired cruising speed.

The second scenario (Figs. 10a and 10b) shows a limit case. The system computes that there is the need to use the override mode. This causes the reference speed v to be immediately set equal to the front speed $\dot{y}_f$. However, as soon as this happens, the acceleration value computed by the proposed control system becomes smaller than the ACC-computed one, causing the override mode to be immediately disabled.

The third scenario (Figs. 11a and 11b) shows the use of the override mode. As a vehicle with a relative speed of 20 km/h is detected only 50 m ahead, the ACC requires a strong deceleration. The override mode, however, is only activated for a small period of time (roughly 1 s in this scenario). The control is the handed back to the standard cruising mode. For this scenario, we also plot the complete platoon dynamics (Fig. 12). The control algorithm is robust to the change of the driving mode, resulting in a distance error smaller than 10 cm.

7.5. Infrastructure-based Speed Advises

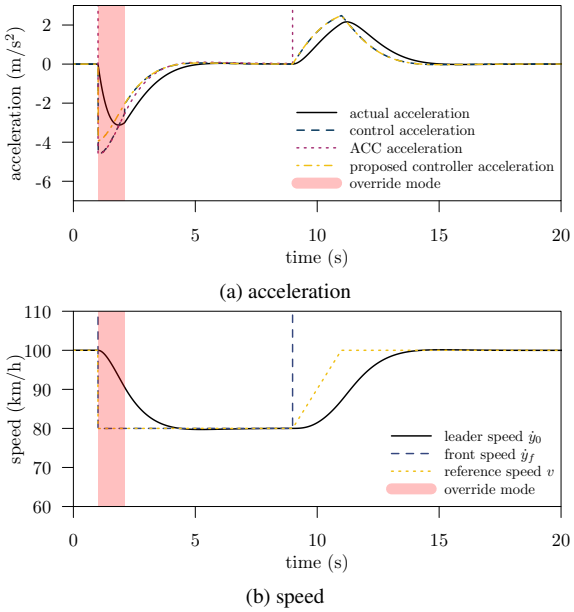
Figures 13 to 15 show the acceleration and the speed dynamics of the leading vehicle subject to speed advises. We show the results of three significant scenarios. In the first one (Figs. 13a and 13b) the infrastructure requires an abrupt change in speed, i.e., from 100 km/h to 40 km/h in just 100 m. This triggers the override mode that brings the speed to the required target in roughly 5 s.

In the second scenario (Figs. 14a and 14b) the required speed change is smaller (i.e., from 100 km/h to 60 km/h in 100 m) and it is thus possible to adapt the reference speed v without violating the rate of change $\bar{v}$. Here, the proposed control algorithm brings the speed down to the target value in roughly 6 s. As the override mode is not used, the convergence is smoother, causing the target distance to be violated by a small amount. A change from 100 km/h to 60 km/h in 6 s corresponds to a travelled distance of roughly 130 m.

The final example (Figs. 15a and 15b), instead, shows an extremely smooth case, where the infrastructure mandates a change in speed from 100 km/h to 60 km/h within 500 m. This is the typical case of road works, where vehicles are required to slow down for the safety of the workers. Such a smooth speed transition over a long time period can not be accomplished by human drivers, which would rather drive with a constant speed of 100 km/h and then suddenly slow down at the speed sign. The control system, instead, can implement any infrastructure-mandated slow down procedure, as shown by the results.

8. Concluding Discussion

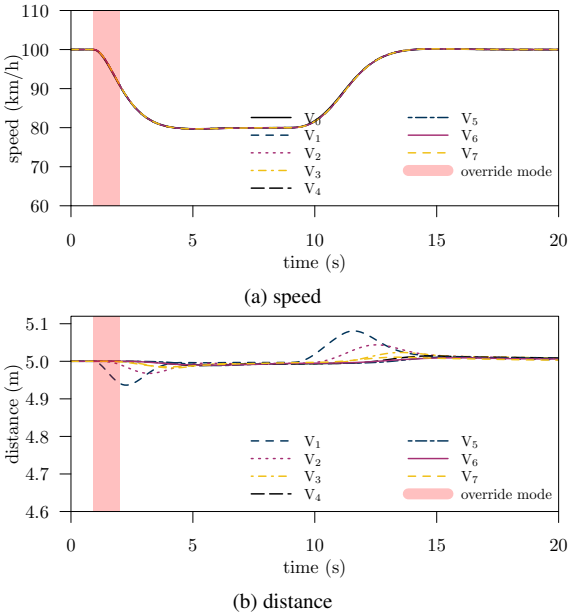
In this work we designed a cooperative automatic driving algorithm from a joint network and control perspective. We derived safety bounds on the inter-vehicle distance depending



(a) acceleration

(b) speed

Figure 11: Acceleration and speed plots showing the response to a slower vehicle cut in maneuver for a distance of 50 m and a relative speed of 20 km/h.



(a) speed

(b) distance

Figure 12: Dynamics of the platoon in response to a slower vehicle cut in maneuver, for a cut in distance of 50 m and a relative speed of 20 km/h.

on vehicle dynamics and packet losses caused by network impairments, showing by means of simulations that such bounds are never violated. On the contrary, the bounds are respected with a large margin due to the robustness of the algorithm to packet losses. In addition, we have shown that the proposed algorithm can be extended to account for external inputs, such as emergency braking scenarios, slower vehicles ahead, and infrastructure-mandated speed changes. This is a fundamental improvement that makes our algorithm practically usable as a complete platooning control system.

Future work includes extending the proposed approach to 2-dimensional platooning formations (such as vehicles moving in parallel lanes) with an arbitrary connection topology, exploiting the fact that the string topology represents the worst-case scenario when computing the smallest eigenvalue of generalised Laplacian matrices [35].

[1] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, H. Nijmeijer, Design and Experimental Evaluation of Cooperative Adaptive Cruise Control, in: IEEE International Conference on Intelligent Transportation Systems (ITSC 2011), Washington, DC, 2011, pp. 260–265.

[2] R. Rajamani, H.-S. Tan, B. K. Law, W.-B. Zhang, Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons, IEEE Trans. on Control Systems Technology 8 (4) (2000) 695–708.

[3] B. van Arem, C. van Driel, R. Visser, The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics, IEEE Trans. on Intelligent Transportation Systems 7 (4) (2006) 429–436.

[4] P. S. Jootel, SAfe Road TRains for the Environment, Final project report, SARTRE Project (Oct. 2012).

[5] S. Singh, Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey, Technical report DOT HS 812 115, National Center for Statistics and Analysis (NHTSA) (Feb. 2015).

[6] SAE Int., Dedicated Short Range Communications (DSRC) Message Set Dictionary, Tech. Rep. J2735-200911, SAE (November 2009).

[7] Wireless Access in Vehicular Environments, Std 802.11p-2010, IEEE (Jul 2010).

[8] 5G Automotive Association, The case for cellular V2X for safety and cooperative driving, White Paper, Tech. rep. (Nov. 2016). URL `http://5gaa.org/pdfs/5GAA-whitepaper-23-Nov-2016.pdf`.

[9] I. Rubin, A. Baiocchi, Y. Sunyoto, I. Turcanu, Traffic management and networking for autonomous vehicular highway systems, Ad Hoc Networks 83 (2019) 125–148.

[10] F. Cunha, L. Villas, A. Boukerche, G. Maia, A. Viana, R. A. Mini, A. A. Loureiro, Data communication in VANETs: Protocols, applications and challenges, Ad Hoc Networks 44 (2016) 90–103.

[11] A. Torres, C. T. Calafate, J.-C. Cano, P. Manzoni, Y. Ji, Evaluation of flooding schemes for real-time video transmission in VANETs, Elsevier Ad Hoc Networks 24 (B) (2015) 3–20.

[12] M. Segata, F. Dressler, R. Lo Cigno, Let's Talk in Groups: A Distributed Bursting Scheme for Cluster-based Vehicular Applications, Elsevier Vehicular Communications 8 (2017) 2–12.

[13] V. Dragonas, K. Oikonomou, K. Giannakis, I. Stavrakakis, A disjoint frame topology-independent TDMA MAC policy for safety applications in vehicular networks, Ad Hoc Networks 79 (2018) 43–52.

[14] G. Giordano, M. Segata, F. Blanchini, R. Lo Cigno, A Joint Network/Control Design for Cooperative Automatic Driving, in: 9th IEEE Vehicular Networking Conference (VNC 2017), IEEE, Torino, Italy, 2017, pp. 167–174.

[15] R. Kianfar, P. Falcone, J. Fredriksson, A Receding Horizon Approach to String Stable Cooperative Adaptive Cruise Control, in: 14th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011), Washington, DC, 2011, pp. 734–739.

[16] R. Rajamani, Vehicle Dynamics and Control, 2nd Edition, 2012.

[17] A. Ali, G. Garcia, P. Martinet, The Flatbed Platoon Towing Model for Safe and Dense Platooning on Highways, IEEE Intelligent Transportation Systems Magazine 7 (1) (2015) 58–68.

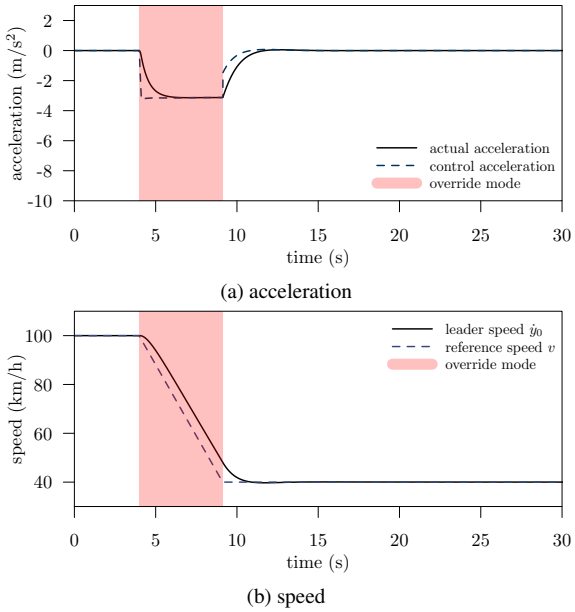
[18] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, M. Nakamura, Cooperative Adaptive Cruise Control in Real Traffic Situations, IEEE Trans. on Intelligent Transportation Systems 15 (1) (2014) 296–305.

[19] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, F. Dressler, Towards Communication Strategies for Platooning: Simulative and Experimental Evaluation, IEEE Trans. on Vehicular Technology 64 (12) (2015) 5411–5423.

[20] J. C. Zegers, E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw, H. Nijmeijer, Consensus-based Bi-directional CACC for Vehicular Platooning, in: American Control Conference (ACC 2016), Boston, MA, 2016, pp. 2578–2584.

[21] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, R. Lo Cigno, A Consensus-based Approach for Platooning with Inter-Vehicular Communications and its Validation in Realistic Scenarios, IEEE Trans. on Vehicular Technology 66 (3) (2017) 1985–1999.

[22] S. Oncu, J. Ploeg, N. Van De Wouw, H. Nijmeijer, Cooperative Adaptive Cruise Control: Network-Aware Analysis of String Stability, IEEE Trans.



(a) acceleration

(b) speed

Figure 13: Acceleration and speed plots showing the response to infrastructure speed advises for a target distance of 100 m and a target speed of 40 km/h.

on Intelligent Transportation Systems 15 (4) (2014) 1527–1537.

[23] V. S. Dolk, J. Ploeg, W. P. M. H. Heemels, Event-Triggered Control for String-Stable Vehicle Platooning, IEEE Trans. on Intelligent Transportation Systems 18 (12) (2017) 3486–3500.

[24] G. Giordano, Structural Analysis and Control of Dynamical Networks, Ph.D. thesis, Università degli Studi di Udine (2016).

[25] R. A. Horn, C. R. Johnson, Matrix Analysis, 2nd Edition, Cambridge University Press, 2012.

[26] P. Greibe, Braking distance, friction and behavior, Technical report, Trafitec, Scion-DTU (Jul. 2007).

[27] S. Jayasuriya, M. Franchek, A Class of Transfer Functions with Non-negative Impulse Response, Journal of dynamic systems, measurement, and control 113 (2) (1991) 313–315.

[28] Y. Liu, P. H. Bauer, Sufficient Conditions for Non-negative Impulse Response of Arbitrary-order Systems, in: IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2008), Macao, China, 2008, pp. 1410–1413.

[29] P. Barooah, P. G. Mehta, J. P. Hespanha, Mistuning-based control design to improve closed-loop stability margin of vehicular platoons, IEEE Trans. on Automatic Control 54 (2009) 21002113.

[30] Y. Zheng, S. E. Li, K. Li, L.-Y. Wang, Stability margin improvement of vehicular platoon considering undirected topology and asymmetric control, IEEE Trans. on Control Systems Technology 24 (2016) 12531265.

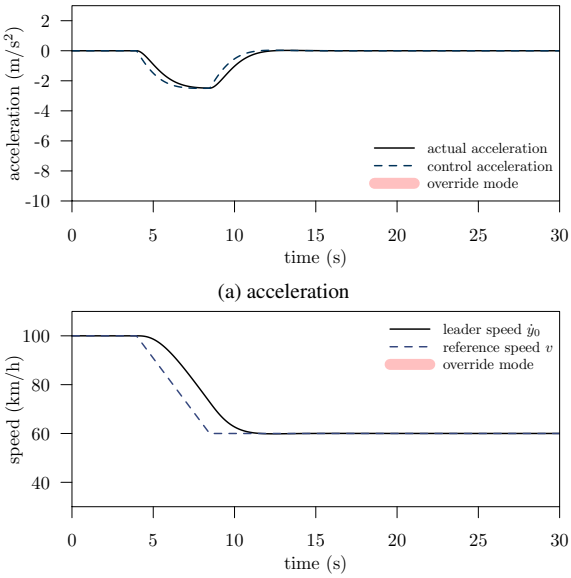
[31] Y. Zheng, S. E. Li, K. Li, W. Ren, Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed H-infinity controller synthesis, IEEE Trans. on Intelligent Transportation Systems 19 (2018) 13531364.

[32] F. Dressler, F. Klingler, M. Segata, R. Lo Cigno, Cooperative Driving and the Tactile Internet, Proceedings of the IEEE 107 (2) (2019) 436–446.

[33] M. Segata, R. Lo Cigno, Automatic Emergency Braking - Realistic Analysis of Car Dynamics and Network Performance, IEEE Trans. on Vehicular Technology 62 (9) (2013) 4150–4161.

[34] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, R. Lo Cigno, PLEXE: A Platooning Extension for Veins, in: 6th IEEE Vehicular Networking Conference (VNC 2014), Paderborn, Germany, 2014, pp. 53–60.

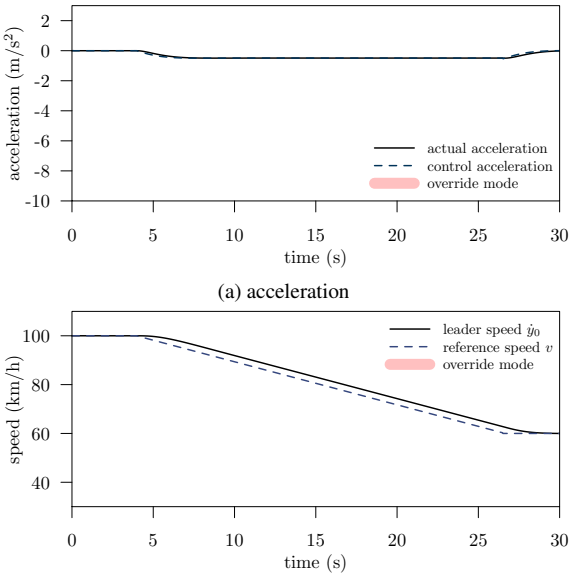
[35] G. Giordano, F. Blanchini, E. Franco, V. Mardanlou, P. L. Montessoro, The Smallest Eigenvalue of the Generalized Laplacian Matrix, with Application to Network-Decentralized Estimation for Homogeneous Systems, IEEE Trans. on Network Science and Engineering 3 (4) (2016) 312–324.



(a) acceleration

(b) speed

Figure 14: Acceleration and speed plots showing the response to infrastructure speed advises for a target distance of 100 m and a target speed of 60 km/h. In this scenario the override mode was never triggered.



(a) acceleration

(b) speed

Figure 15: Acceleration and speed plots showing the response to infrastructure speed advises for a target distance of 500 m and a target speed of 60 km/h. In this scenario the override mode was never triggered.