HAL open science

Stochastic String Stability of Vehicle Platoons via Cooperative Adaptive Cruise Control With Lossy Communication

Francesco Acciani, Paolo Frasca, Geert Heijenk, Anton Stoorvogel

▶ To cite this version:

Francesco Acciani, Paolo Frasca, Geert Heijenk, Anton Stoorvogel. Stochastic String Stability of Vehicle Platoons via Cooperative Adaptive Cruise Control With Lossy Communication. IEEE Transactions on Intelligent Transportation Systems, 2022, 23 (8), pp.10912-10922. 10.1109/TITS.2021.3097199 . hal-03427758

HAL Id: hal-03427758

https://hal.science/hal-03427758

Submitted on 14 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Stochastic String Stability of Vehicle Platoons via Cooperative Adaptive Cruise Control with Lossy Communication

Francesco Acciani, Paolo Frasca, Geert Heijenk, Anton Stoorvogel

***Abstract*—This paper is about obtaining stable vehicle platooning by using Cooperative Adaptive Cruise Control when the communication is unreliable and suffers from message losses. We model communication losses as independent random events and we propose an original design for the cooperative controller, which effectively mitigates the effects of the losses. Our design explicitly takes into account the stochastic nature of the losses by considering both the average evolution of the system and the stochastic variations around it. The control design promotes both plant stability and string stability of the average error dynamics by an $\mathscr{H}_\infty$ approach, while minimizing the variance of the trajectories around their average. We show by simulations that the proposed controller is able to compensate losses even for high loss probabilities.**

I. Introduction

A platoon is a group of vehicles that move close together at the same speed: automated vehicle platooning has been heralded since the eighties as an enabler of effective road usage. Crucial to the effectiveness of platoons is their ability to remain *string stable*, that is, to dampen disturbances that may affect the motion of vehicles.

Adaptive Cruise Control systems, which are becoming widespread in the market, allow vehicles to adapt their dynamics to their surroundings by taking measurements of relative distances and speeds. Platoons that are based on ACC systems, however, may not be string stable [1]. Instead, Cooperative Adaptive Cruise Control (CACC) provides a more effective way to stabilize platoons, with performance that is superior to ACC without cooperation. Besides performing relative measurements of distance and speed, CACC allow vehicles to communicate relevant internal variables, such as their current control inputs.

Cooperation in CACC hinges on the ability of the vehicles to communicate effectively and consequently is prone to disruption if communication is problematic, for instance if it is severely affected by delays or packet losses. When CACC is deployed over unreliable communication, researchers have observed that "as the probability of data loss increases, the behavior of the platoon becomes unacceptable [...], unless the time headway constant is also increased" [2]. The importance of communication for string stabilization is supported by a large body of work spanning across at least two decades [3]–[5]: in the context of CACC, researchers have proposed both robustness analyses [6]–[10] and designs to mitigate the disruptions [11]–[13].

F. Acciani, G. Heijenk and A. Stoorvogel are with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, Netherlands.

P. Frasca is with Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, GIPSA-lab, Grenoble, France. E-mail: paolo.frasca@gipsa-lab.fr.

The contribution of this paper is a novel cooperative control scheme that is designed to be robust to communication losses. In defining the design problem, we make well-accepted assumptions about the communication between the vehicles: messages are sent at uniformly-spaced sample times and failure (loss) events are stochastic. On this matter, it is worth noting that even though string stability notions have been studied for half a century [14]–[16], relatively little attention has been devoted to stochasticity in this context. Deterministic models are more popular in the control systems community [17]–[19], with a few recent exceptions: [8] has performed string stability analysis for mean and covariance dynamics, [20] has studied stochastic losses by a deterministic equivalent, and [21] has proposed stochastic string stability notions for platoons with random delays. Recently, event-driven communication has also been proposed in this context [22]. These papers are however focused on the analysis of a given platoon while this paper focuses on the design of a suitable controller. Moreover, the above papers (with the exception [8]) are limited to the average dynamics while our paper also considers the variance.

We therefore model message losses as independent stochastic events and our main contribution is the design of a cooperative controller that mitigates their effects. We define our control objective as the *stability of a string of stochastic systems*: the pursuit of this objective leads us to design a distributed controller such that the average behavior of the platoon is string stable and the variance around the average is minimized. This twofold requirement is meant to ensure that disturbances are dampened along most of the realized trajectories of the stochastic system. This objective is achieved by means of $\mathscr{H}_\infty$ control design methods in the state space, which are applied on the expected dynamics. Despite the large body of literature on communication and CACC, our work is the first to incorporate the stochastic nature of losses in the design of a string-stabilizing controller with the purpose of mitigating their effects. The obtained distributed controller is tested in simulations that show its ability to compensate for high probability of losses while keeping the platoon tightly together. Being distributed, our controller can be scaled up to any number of vehicles without loss of performance. Furthermore, even though its design assumes that the parameters of the vehicle motion and of the communication are known and uniform across the vehicles, the controller is robust against uncertainties in the parameters.

Key to our design solution is the original formulation of the platoon stabilization problem, which lends itself to the application of powerful $\mathscr{H}_\infty$ techniques [23]. The decades-long history of $\mathscr{H}_\infty$ design has already seen applications to platooning in [24], [25] and to other stochastic networked control systems in [26], [27]. Another $\mathscr{H}_\infty$ design for platoons with packet losses can be found in [28], [29]. The latter papers are close in their objectives compared to this work but an advantage of our approach is the ability to distinguish the effect of packet losses on the average behavior and the variance seperately.

Finally, we would like to remind that, even though state-space models (as we use here) very naturally allow for the modeling of packet losses, the issue of string stability on lossy networks can also be approached by a frequency-domain design: in [30] a dynamic controller is used together with an unknown input observer that reconstructs the missing inputs. Related references include [31], [32] on frequency-domain analyses and designs and [18] on using predictors.

Paper structure: The rest of the paper is organized as follows. In Section II we describe the full dynamic model of the platoon. In Section III we discuss a suitable notion of string stability , which we set as design objective. In Section IV we describe our control architecture with all its components. In Section V we test the performance of our controller in simulation, showing its effectiveness. We conclude with some reflections and perspectives in Section VI.

II. Vehicle and platoon models

This section describes our model for the platoon, including both the motion model of the vehicles, the error dynamics with respect to the control objective, and the assumptions about the available measurements and communications.

A. Vehicle model and spacing policy

We assume a well-established [33] dynamical model of the single vehicle, which takes into account only its longitudinal motion: lateral control, though relevant for practical implementations of our methods, is left outside the scope of this work. We let scalar $q_i(t)$ be the position of vehicle i, $v_i(t)$ its speed and $a_i(t)$ its acceleration and we assume the dynamics

$$\dot{a}_i(t) = -\frac{1}{\tau}a_i(t) + \frac{1}{\tau}u_i(t-\phi), \tag{1}$$

where $\tau > 0$ is the vehicle time constant, and $\phi > 0$ the internal delay of the system. As usual, $u_i(t)$ is the input for our system. Following a consolidated tradition, we assume for simplicity that the constants are known and independent of i, thereby making the platoon homogeneous: accounting for heterogeneity and uncertainty in platooning is a relevant issue and an active topic of research [17], [34]–[37].

The objective of the controller is to maintain the distance between vehicles at a certain reference: we adopt here a speed-dependent time spacing policy, according to [38]. The desired distance between vehicle i and $i-1$ is then defined as

$$d_{r,i}(t) = R_i + hv_i(t),$$

where the positive constant h is a time-headway (lower values for h represent shorter distances between vehicles) and R_i is the standstill desired distance. In the remainder of this paper R_i will be neglected, without loss of generality as it is always possible to find a coordinate transformation that is equivalent to choosing $R_i = 0$. We can now define the error for vehicle i as

$$e_i(t) = d_i(t) - d_{r,i}(t) = q_{i-1}(t) - q_i(t) - hv_i(t), \tag{2}$$

where $d_i(t) = q_{i-1}(t) - q_i(t)$ is the distance between vehicles i and $i-1$. Finally, we observe that the vector state

$$x(t) = \begin{bmatrix} e_i(t) \\ \dot{e}_i(t) \\ \ddot{e}_i(t) + \frac{h}{\tau}u_i(t-\phi) \end{bmatrix}$$

follows the closed dynamics

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ -\frac{h}{\tau} \\ \frac{h-\tau}{\tau^2} \end{bmatrix} u_i(t-\phi) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u_{i-1}(t-\phi), \tag{3}$$

which is consistent with (1)-(2) and which we shall refer to as the *error dynamics*.

B. Measurements and communication

We assume that every vehicle is able to measure the distance and relative speed with respect to the preceding vehicle (say, by a radar sensor), as well as its own absolute speed and acceleration. These measurements permit to reconstruct the first and second components of the state, i.e., $e_i(t) = d_i(t) - hv_i(t)$ and $\dot{e}_i(t) = v_{i-1}(t) - v_i(t) - ha_i(t)$. However, we cannot measure the third component of the state as there is no simple way to measure relative acceleration or jerk. Therefore, we define the *measured* output $y(t)$ as:

$$y_i(t) = Cx_i(t-\psi) + w(t)$$

where $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, positive scalar ψ is a measurement delay, and $w(t)$ is measurement noise.

Since we are looking for a cooperative controller, we assume that the vehicles are also able to communicate. We thus define a *communicated* output, which is the input of the previous vehicle: this is a noise-free quantity which however suffers from losses and transmission delay:

$$y_{\text{comm}_i}(t) = f(u_{i-1}(t-\theta))$$

where $f(u_{i-1}(t-\theta))$ models the network unreliability:

$$f(u_{i-1}(t-\theta)) = \begin{cases} u_{i-1}(t-\theta) & \text{if communication available} \\ 0 & \text{otherwise.} \end{cases}$$

A scheme representing which quantities are used by the vehicles can be found in Fig. 1: typical values for the parameters of the continuous time systems are [39]:

$$\tau = 0.1s, \quad \phi = 0.2s, \quad \theta = 0.02s, \quad \psi = 0.05s.$$

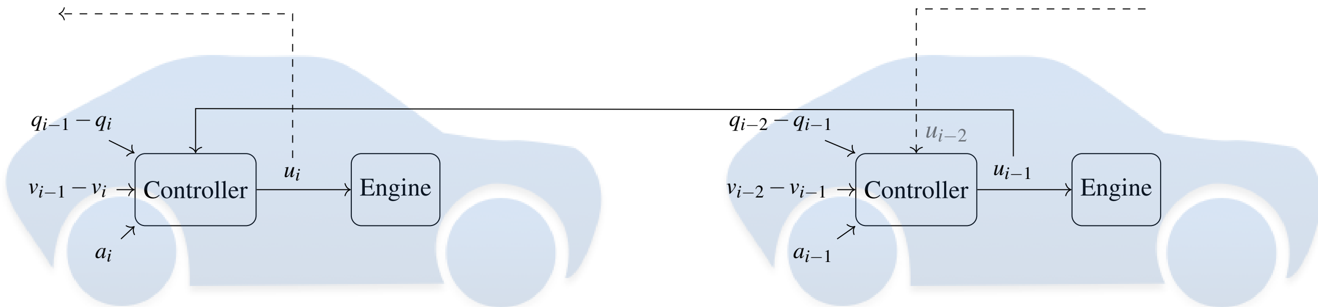


Fig. 1: Details of the interconnection between two vehicles in the platoon.

C. *Communication model and discrete-time dynamics*

After modelling the continuous-time dynamics, we are going to convert it to a discrete-time one by choosing a suitable inter-sampling time T_s. Indeed, the discrete-time framework is more suitable to accommodate for the communication between vehicles, which is inherently a discrete time phenomenon, including the stochastic losses. To model the communication, we assume that transmissions occur at the sampling times and that, upon each transmission, each vehicle is able to receive the information that is sent by the preceding one with probability $1-p$ (thus p is the probability for each sent message to be lost). Transmissions are assumed to be time and space independent, therefore a loss happening at some time for one vehicle does not influence future or current losses for any other vehicle in the platoon. Furthermore, discrete time allows us to easily incorporate delays in our analysis, so long as we assume that delays are multiples of the sampling time.

Based on the above considerations, we choose a inter-sampling time $T_s = 0.01s$ and $d = \frac{\phi}{T_s}$, $m = \frac{\psi}{T_s}$, and $r = \frac{\theta}{T_s}$. We thus obtain the model

$$x_i(k+1) = Ax_i(k) + Bu_i(k-d) + Eu_{i-1}(k-d), \qquad (4)$$

where

$$A = \begin{bmatrix} 1 & T_s & \tau T_s - \tau^2\left(1-e^{\frac{-T_s}{\tau}}\right) \\ 0 & 1 & \tau\left(1-e^{\frac{-T_s}{\tau}}\right) \\ 0 & 0 & e^{\frac{-T_s}{\tau}} \end{bmatrix}$$

$$B = \begin{bmatrix} -\frac{T_s^2}{2} + T_s(\tau-h) - \tau(\tau-h)\left(1-e^{-\frac{T_s}{\tau}}\right) \\ -T_s + (\tau-h)\left(1-e^{-\frac{-T_s}{\tau}}\right) \\ -\frac{\tau-h}{\tau}\left(1-e^{-\frac{T_s}{\tau}}\right) \end{bmatrix}$$

$$E = \begin{bmatrix} \frac{T_s^2}{2} - T_s\tau + \tau^2\left(1-e^{\frac{-T_s}{\tau}}\right) \\ T_s - \tau\left(1-e^{\frac{-T_s}{\tau}}\right) \\ 1-e^{\frac{-T_s}{\tau}} \end{bmatrix},$$

and by a slight notational abuse the discrete-time variables are denoted by the same letters as the corresponding continuous-time variables. The full model then reads

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k-d) + Eu_{i-1}(k-d) \\ y_i(k) &= Cx_i(k-m) \qquad (5) \\ y_{\mathrm{comm}\,i}(k) &= u_{i-1}(k-r)\,\delta_i(k), \end{aligned}$$

where $\delta_i(k)$ is a Bernoulli random variable with mean $1-p$. For controller design purposes, we shall disregard measurement delays (which can be incorporated in the larger input delay) and transmission delays (which we assume to be small). We thus effectively work on

$$\begin{aligned} x_i(k+1) &= Ax_i(k) + Bu_i(k-d) + Eu_{i-1}(k-d) \\ y_i(k) &= Cx_i(k) \qquad (6) \\ y_{\mathrm{comm}\,i}(k) &= u_{i-1}(k)\,\delta_i(k). \end{aligned}$$

The general model (5), however, shall be used to test the controller in simulation.

III. String stability and control objectives

In order to make every vehicle in the platoon smoothly follow the preceding one, our control objective is twofold: stabilizing the error (2) to zero and ensuring a *string stability* property. This latter notion refers to the uniform boundedness of the states or, equivalently, to the dampening of disturbances along the string of vehicles. Indeed, disturbance amplification is not only a risk for a safe operation of the platoon, but it also compromises traffic flow stability and throughput [40]. If the metric to measure disturbances is $\mathscr{L}_2$, then the resulting notion is the so-called $\mathscr{L}_2$-string stability that requires

$$\frac{\|u_i\|_{\mathscr{L}_2}}{\|u_{i-1}\|_{\mathscr{L}_2}} < 1. \qquad (7)$$

For a proper definition of a feasible and useful control objective, we make three choices that correspond to three crucial points to account for. Firstly, we need to ensure *both plant stability* (that is, stabilization of (3) for each vehicle) *and string stability* for the whole platoon. To this purpose, we consider a combination of input and error as

$$z_i(k) = [\varepsilon e_i(k), r\,u_i(k)]^\top$$

(with suitable positive constants ε, r) and replace (7) by

$$\frac{\|z_i\|_{\mathscr{L}_2}}{\|u_{i-1}\|_{\mathscr{L}_2}} < 1, \qquad (8)$$

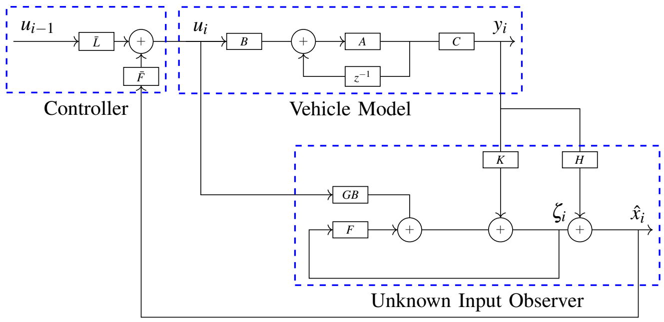


Fig. 2: Control architecture with $\mathscr{H}_\infty$ controller and state observer, that is, $u_i(k) = \bar{F}\hat{x}_i(k) + \bar{L}u_{i-1}(k))$. This controller is applied to the average dynamics. The full control architecture includes the switching logic and the lifting block.

which –contrary to (7)– promotes both plant and string stability. Secondly, instead of imposing the ratio in (8) to be smaller than one, we aim to make it as small as possible:

$$\min \frac{\|z_i\|_{\mathscr{L}_2}}{\|u_{i-1}\|_{\mathscr{L}_2}}. \tag{9}$$

This formulation can take advantage of the solid literature on $\mathscr{H}_\infty$ control design devoted to minimizing this cost.

Thirdly and most importantly, our system is *stochastic* in nature and therefore it requires us to adapt the deterministic definition of string stability that we have given above. Ideally, one would like that disturbances be attenuated along every or almost every trajectories. However, such a requirement would be too restrictive for performance in terms of the necessary headway h. On the contrary, requiring disturbance attenuation in expectation only would allow for a small headway h but would actually be too weak a requirement, since individual trajectories are free to significantly deviate from the average. In view of this robustness/performance trade-off in the definition of string stability in the stochastic setting, in our work we aim at ensuring an acceptable behavior of the stochastic string with good performance by a two-step approach: we first impose string stability of the *expected trajectory* and then we *minimize the variance* of the trajectories around their expectation.

IV. Control design

The design of our cooperative controller is done in several phases, which roughly correspond to the main components of the control architecture, summarized in Fig. 2. The vehicle plant is fed by input $u_i(k)$, which is computed by the corresponding controller. The controller produces $u_i(k)$ by using the input of the preceding vehicle $u_{i-1}(k)$ when available, the estimate of the state from an unknown input observer, and the delayed samples of $u_i(k)$ and $u_{i-1}(k)$. More specifically, to model the effects of delays in the vehicle, we study a lifted system, which increases the size of the state. However, it is possible to design the unknown input observer based on the small, delay-free system, and enrich the output of the unknown input observer with previous measurements of $u_i(k)$ and $u_{i-1}(k)$ when available, to fully reconstruct the lifted state, which is the controller's input. To keep the synthesis and the design of the controller simple, we present all the ingredients for the controller synthesis without taking delays into account, while the actual computation of the controller takes into account the lifted, high-dimensional, system. We shall first design a cooperative controller for the generic vehicle i in the *full information* case, that is, assuming perfect knowledge of $x_i(k)$ (state-feedback) and $u_{i-1}(k)$ (ideal communication without losses). This preliminary controller, in the form

$$u_i(k) = \bar{F}x_i(k) + \bar{L}u_{i-1}(k), \tag{10}$$

has the objective of ensuring both closed-loop stability of the error dynamics and string stability of the platoon. Next, we take into account the communication limitations and use $u_{i-1}(k)$ only when it is available; the control law therefore becomes the following switching one:

$$u_i(k) = \begin{cases} F_1x_i(k) + Lu_{i-1}(k) & \text{if } u_{i-1}(k) \text{ available} \\ F_2x_i(k) & \text{otherwise.} \end{cases} \tag{11}$$

We adapt the controller to the stochastic nature of our system in two steps: (i) we study the average or *expected* dynamics of the system and design the controller in such a way that the expected behavior matches the ideal behavior (without losses); (ii) we include the criterion of minimizing the *variance* to account for the dispersion of the actual trajectories.

Finally, we design an *unknown input observer* to produce an estimate of the state $\hat{x}_i(k)$ that can be used for state-feedback and we incorporate delays by applying the aforementioned steps to a suitably *lifted* system. All these steps are suitably detailed in the following subsections.

A. Lossless communication and state-feedback

To design the controller, we study the single vehicle dynamics in the lossless case. From now on, we drop the subscripts and we use $\xi(k)$ to denote $u_i(k)$, the input of the i^{th} vehicle, and $\nu(k)$ to denote $u_{i-1}(k)$, the input of the previous vehicle $i-1$. The model of the single vehicle that we use to design a controller is the following:

$$\begin{cases} x(k+1) &= Ax(k) + B\xi(k) + E\nu(k) \\ y(k) &= Cx(k) \\ z(k) &= C_zx(k) + D_\xi\xi(k). \end{cases} \tag{12}$$

The variable $y(k)$ is the measured output defined by

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

while $z(k)$ is a *performance* output: it is not measured, but it is used to design the controller. The matrices C_z and D_ξ are:

$$C_z = \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad D_\xi = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

so that the performance output is a combination of error and local input:

$$z(k) = [\varepsilon e(k), \xi(k)]^\top.$$

In order to promote both asymptotic stability and string stability, we define the $\mathscr{H}_\infty$ control objective of minimizing the norm

$$\|H\|_{\mathscr{H}_\infty} = \frac{\|z\|_{\mathscr{L}_2}}{\|\nu\|_{\mathscr{L}_2}}. \tag{13}$$

Comparing (13) with (7), the choice of z becomes clear: we put a small weight ε on $e_i(k)$ so that $z(k)$ is close to $\xi(k)$. In order to ensure string stability, we require this norm to be made smaller than one for some small ε by the $\mathscr{H}_\infty$ design. To solve this design problem, we make the assumption that the full state is available for feedback, thereby focusing on the system

$$\begin{cases} x(k+1) &= Ax(k) + B\xi(k) + E\nu(k) \\ y(k) &= x(k) \\ z(k) &= C_z x(k) + D_\xi \xi(k). \end{cases} \tag{14}$$

On this system, the $\mathscr{H}_\infty$ problem can be solved by applying the following result from [23].

Lemma 1 ($\mathscr{H}_\infty$ design)**.** *Consider the following system*

$$\begin{cases} x(k+1) &= Ax(k) + B\xi(k) + E\nu(k) \\ z(k) &= Cx(k) + D_1\xi(k) \end{cases} \tag{15}$$

and assume that system (A, B, C, D_1) has no invariant zeroes on the unit circle. Then, a controller in the form

$$\xi(k) = \overline{F}x(k) + \bar{L}\nu(k)$$

exists, such that the closed-loop transfer function from ν to z has $\mathscr{H}_\infty$-norm less than one, if and only if a symmetric matrix $P \geq 0$ exists, such that:

1) $V = D_1^\top D_1 + B^\top PB > 0$
2) $R = I - E^\top PE + E^\top PBV^{-1}B^\top PE > 0$
3) *P satisfies the following Riccati equation:*

$$P = A^\top PA + C^\top C - \begin{bmatrix} B^\top PA + D_1^\top C \\ E^\top PA \end{bmatrix}^\top G(P)^{-1} \begin{bmatrix} B^\top PA + D_1^\top C \\ E^\top PA \end{bmatrix},$$

$$\text{where } G(P) = \begin{bmatrix} D_1^\top D_1 & 0 \\ 0 & -I \end{bmatrix} + \begin{bmatrix} B^\top \\ E^\top \end{bmatrix} P \begin{bmatrix} B & E \end{bmatrix}.$$

If such P matrix exists, then the static feedback matrices $\overline{F}$ and $\overline{L}$ can be chosen as

$$\begin{aligned} \overline{F} &= -[D_1^\top D_1 + B^\top PB]^{-1}[B^\top PA + D_1^\top C] \\ \overline{L} &= -[D_1^\top D_1 + B^\top PB]^{-1}B^\top PE. \end{aligned}$$

A general procedure to compute matrix P can be found in [41]. In the next subsections we shall see more precisely how this deterministic design problem can be useful in our stochastic lossy system (6).

B. Controlling the expectation

By using the model (12) for the single vehicle dynamics, and assuming that the communication between vehicles is modelled by a Bernoulli process, our system consists of n subsystems, each of which can be described by the dynamics (6):

$$\begin{cases} x(k+1) &= Ax(k) + B\xi(k) + E\nu(k) \\ y(k) &= Cx(k) \\ y_{\text{comm}} &= \delta(k)\nu(k), \end{cases} \tag{16}$$

where $\delta(k) = 0$ when the communication from the preceding vehicle is lost, and $\mathbb{P}[\delta(k) = 1] = 1 - p$. We recall that the losses, i.e. $\delta(k)$, are time and space independent: the losses experienced by different vehicles are uncorrelated, as well as subsequent losses experienced by a single vehicle. In our design, we assume the loss probability p to be known, even though in practice it has to be estimated from sample ratios of successful transmissions.

The following simple result gives us the average behavior of the lossy system (16) interconnected with a switching controller.

Lemma 2 (Expected dynamics)**.** *The expected value $\mathbb{E}[x(k)]$ of the state of system* (16) *interconnected with a stochastic switching controller in the form:*

$$\xi(k) = \delta(k)\big(F_1 x(k) + Ly_{\text{comm}}(k)\big) + \big(1 - \delta(k)\big)F_2 x(k) \tag{17}$$

has the same dynamics as the state of system (16) *when interconnected with a non-switching deterministic controller in the form:*

$$\xi(k) = \overline{F}x(k) + \overline{L}\nu(k),$$

where

$$\begin{aligned} \overline{F} &= (1-p)F_1 + pF_2 \\ \overline{L} &= (1-p)L. \end{aligned} \tag{18}$$

This lemma can be used to find a switching controller such that the expected dynamics of the lossy system is the same as a given deterministic system. In our design, we shall impose that *the expected system coincides with the nominal system without losses* (14). In order to exploit the state-feedback controller provided by Lemma 1, we compute $\overline{L}$ and $\overline{F}$ by applying Lemma 1 with $C = C_z$, $D_1 = D_\xi$. Matrix F_1 will be chosen later in such a way to minimize covariance, as detailed in the next subsection. Matrix F_2 is then uniquely determined by (18).

C. Covariance minimization

Thanks to Lemma 2, we are able to guarantee the expected behaviour of the platoon to be string stable, provided that it is possible to find a state feedback $\xi = \overline{F}x + \overline{L}\nu$ for the lossless system (14) such that the $\mathscr{H}_\infty$ gain is less than one. Now, we focus on minimizing the covariance of the error, in order to keep the trajectories to be close to the expected, string stable, behavior. Therefore we want to minimise the variance

$$\mathbb{E}\left[\|(x - \mathbb{E}[x])\|_2^2\right]. \tag{19}$$

Theorem 1 (Controller design)**.** *Consider the dynamics*

$$x(k+1) = Ax(k) + B\xi(k) + E\nu(k),$$

with the stochastic control law

$$\xi(k) = \begin{cases} F_1 x(k) + L\nu(k) & \text{with probability } 1-p \\ F_2 x(k) & \text{with probability } p, \end{cases}$$

and the same dynamics with the nominal deterministic control law $\xi(k) = \bar{F}x(k) + \bar{L}v(k)$. Assume that

$$F_1 = \overline{F} - pL\mathbb{E}[v]\mathbb{E}[x]^\top\left(\mathbb{E}[\tilde{x}\tilde{x}^\top] + \mathbb{E}[x]\mathbb{E}[x]^\top\right)^{-1} \quad (20)$$

$$F_2 = \frac{1}{p}\left(\overline{F} - (1-p)F_1\right) \quad (21)$$

$$L = \frac{1}{1-p}\overline{L}, \quad (22)$$

where $\tilde{x}(k) = x(k) - \mathbb{E}[x(k)]$. Then, the expectation of the stochastic dynamics follows the nominal dynamics

$$\begin{cases} \mathbb{E}[x(k+1)] & = A\mathbb{E}[x(k)] + B\mathbb{E}[\xi(k)] + E\mathbb{E}[v(k)] \\ \mathbb{E}[\xi(k)] & = \overline{F}\mathbb{E}[x(k)] + \overline{L}\mathbb{E}[v(k)], \end{cases} \quad (23)$$

and cost (19) *is minimized over all F_1 given $\bar{F}$.*

Proof. We can compute the dynamics of the expected state

$$\begin{aligned}\mathbb{E}[x(k+1)] =& A\mathbb{E}[x(k)] + B\left((1-p)F_1 + pF_2\right)\mathbb{E}[x(k)] \\ &+ (1-p)BL\mathbb{E}[v(k)] \end{aligned} \quad (24)$$

and, by some lengthy manipulations using (21) , which are not reported, the dynamics of its covariance

$$\begin{aligned}\mathbb{E}\left[\tilde{x}_+\tilde{x}^\top{}_+\right] =& A\mathbb{E}[\tilde{x}\tilde{x}^\top]A^\top + E\mathbb{E}[\tilde{v}\tilde{v}^\top]E^\top + 2(1-p)BL\mathbb{E}[\tilde{v}\tilde{v}^\top]E^\top \\ &+ (1-p)BL\mathbb{E}[\tilde{v}\tilde{v}^\top]L^\top B^\top \\ &+ p(1-p)BL\mathbb{E}[v]\mathbb{E}[v]^\top L^\top B^\top \\ &- B\overline{F}\mathbb{E}[x]\mathbb{E}[x]^\top\overline{F}^\top B^\top + 2B\overline{F}\mathbb{E}[\tilde{x}\tilde{x}^\top]A^\top \\ &- 2(1-p)B\overline{F}\mathbb{E}[x]\mathbb{E}[v]^\top L^\top B^\top \\ &+ \frac{1}{p}B\overline{F}\left(\mathbb{E}[\tilde{x}\tilde{x}^\top] + \mathbb{E}[x]\mathbb{E}[x]^\top\right)\overline{F}^\top B^\top \\ &+ \frac{1-p}{p}BF_1\left(\mathbb{E}[\tilde{x}\tilde{x}^\top] + \mathbb{E}[x]\mathbb{E}[x]^\top\right)F_1^\top B^\top \\ &+ 2(1-p)BF_1\mathbb{E}[x]\mathbb{E}[v]^\top L^\top B^\top \\ &- 2\frac{1-p}{p}[BF_1\left(\mathbb{E}[\tilde{x}\tilde{x}^\top] + \mathbb{E}[x]\mathbb{E}[x]^\top\right)\overline{F}^\top B^\top, \end{aligned}$$

where we have dropped the dependence on k to increase the readability and denoted $\tilde{x} = \tilde{x}(k)$ and $\tilde{x}_+ = \tilde{x}(k+1)$.

We want to find F_1 to minimise (19), therefore we compute

$$\begin{aligned}\frac{\partial Tr\left[\tilde{x}_+\tilde{x}^\top{}_+\right]}{\partial BF_1} =& 2\frac{1-p}{p}BF_1\mathbb{E}[xx^\top] + 2(1-p)BL\mathbb{E}[v]\mathbb{E}[x]^\top \\ &- 2\frac{1-p}{p}B\overline{F}\mathbb{E}[xx^\top], \end{aligned}$$

and conclude that matrix F_1 in (20) minimises variance (19). Given F_1, matrices F_2 and L descend from (21) and (22). □

The time-varying matrix gain in (20) should be used in the controller, but its computation is problematic because it requires the knowledge of expectation and variance of v. Therefore, we propose an approximation.

D. Approximate covariance minimization controller

The controller gain in (20) is time varying: to minimise the covariance of the error $x(k+1)$ the controller needs to know the expected value of the input at time k, i.e. $\mathbb{E}[v(k)]$, the expected value of the error itself $\mathbb{E}[x(k)]$ and its covariance $\mathbb{E}[\tilde{x}(k)\tilde{x}(k)^\top]$. In order to derive a handier relation, we look for a static matrix to approximate the time varying F_1 in (20). This approximation will be derived by looking at expected dynamics. Even though F_1 is time-varying, the average controlled dynamics (23) is time-invariant and the corresponding transfer function is

$$G_{v\to\xi} = \frac{\Xi(z)}{N(z)} = \overline{F}(zI - A - B\overline{F})^{-1}(E + B\overline{L}) + \overline{L}.$$

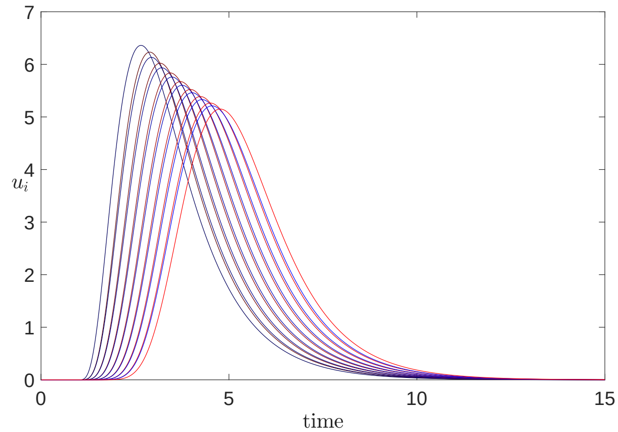


Fig. 3: Real and estimated inputs for a platoon of 8 vehicles. The real inputs are displayed in red, while the output of $G_{v\to\xi}$ in blue. See Section V for more information on the simulations.

Fig. 3 shows by an example that this relation is effective in reconstructing the inputs, up to a delay that was not considered in the model. Next, we further simplify the relation $v \to \xi$ by approximating it by its mere low frequency gain $g := \lim_{z\to 1} G_{v\to\xi}$ and therefore we approximate:

$$\mathbb{E}[v(k)] \approx \frac{1}{g}\mathbb{E}[\xi(k)]. \quad (25)$$

By using the approximation (25) in (23) we get

$$\mathbb{E}[v(k)] \approx \frac{1}{g}\mathbb{E}[\xi(k)] = \frac{1}{g}\overline{F}\mathbb{E}[x(k)] + \frac{1}{g}\overline{L}\mathbb{E}[v(k)],$$

and finally

$$\mathbb{E}[v(k)] \approx \left(1 - \frac{1}{g}\overline{L}\right)\frac{1}{g}\overline{F}\mathbb{E}[x(k)]. \quad (26)$$

Now we can use approximation (26) in (20), which becomes

$$F_1 \approx \overline{F} - \frac{p}{1-p}\overline{L}\left(1 - \frac{1}{g}\overline{L}\right)\frac{1}{g}\overline{F}\mathbb{E}[x]\mathbb{E}[x]^\top\mathbb{E}[xx^\top]^{-1}.$$

Finally, we disregard the statistical dispersion by approximating $\mathbb{E}[x]\mathbb{E}[x]^\top\mathbb{E}[xx^\top]^{-1}$ by the identity matrix. We thus obtain

$$F_1 \approx \left(1 - \frac{p}{1-p}\overline{L}\left(1 - \frac{1}{g}\overline{L}\right)\frac{1}{g}\right)\overline{F}, \quad (27)$$

which is the constant gain used in our implementation.

E. State observer design

Even though in Section IV-A we have designed a state-feedback controller, the output of system (12) is not the full state. Since one input is unknown, we need to estimate the state, which we do via an Unknown Input Observer [42], [43]. Because of the measurement delay, we can only estimate the delayed state

$$x_d(k) = x(k-m), \tag{28}$$

and by substituting (28) in (12) we get

$$\begin{aligned} x_d(k+1) &= Ax_d(k) + B\xi_d(k) + E\nu_d(k) \\ y(k) &= Cx_d(k), \end{aligned}$$

where $\xi_d(k) = \xi(k-d-m)$ and $\nu_d(k) = \nu(k-d-m)$ are the delayed inputs.

As observer for the delayed state $x_d(k)$ it is possible to use the following dynamical system:

$$\begin{aligned} \zeta(k+1) &= F\zeta(k) + GB\xi_d(k) + Ky(k) \\ \hat{x}(k) &= \zeta(k) + Hy(k). \end{aligned} \tag{29}$$

After simple algebraic manipulations, it can be shown that the estimate error $\varepsilon(k) = x_d(k) - \hat{x}(k)$ follows the dynamics:

$$\begin{aligned} \varepsilon(k{+}1) = {} & Ax(k) + B\xi_d(k) + E\nu_d(k) - F\zeta(k) - GB\xi_d(k) \\ & - K_1Cx(k) - K_2y(k) - Hy(k+1) \\ = {} & Ax(k) + B\xi_d(k) + E\nu_d(k) - F(\hat{x}(k) - Hy(k)) \\ & - GB\xi_d(k) - K_1Cx(k) - K_2y(k) \\ & - HCAx(k) - HCB\xi_d(k) - HCE\nu_d(k) \\ = {} & (A - K_1C - HCA)(x(k) - \hat{x}(k)) + (A - K_1C - HCA - F)\hat{x}(k) \\ & + (I - G - HC)B\xi_d(k) + (I - HC)E\nu_d(k) + (FH - K_2)y(k), \end{aligned}$$

where $K = K_1 + K_2$. Since CE is injective, we can choose

$$\begin{aligned} F &= A - K_1C - HCA \\ K_2 &= FH \\ H &= E[(CE)^\top CE]^{-1}(CE)^\top \\ G &= I - HC \end{aligned}$$

to obtain $\varepsilon(k+1) = (A - K_1C - HCA)\varepsilon(k)$. In order to bring the estimate error to zero, we can choose K_1 to have the desired poles for F, because the couple $(A - HCA, C_1)$ is observable. A straightforward solution is choosing the gain matrix K_1 to impose a deadbeat response with all the poles of F in 0. In our implementation, we use $\hat{x}(k)$ as estimate for $x(k)$, effectively disregarding the measurement delay.

F. Modeling input delays: Lifted system

So far, our design has disregarded delays. Indeed, delays can be accounted for by writing a standard lifted system. Let us recall that the system to control, for the full information case and including the input delay, is

$$\begin{cases} x(k+1) &= Ax(k) + B\xi(k-d) + E\nu(k) \\ z(k) &= C_zx(k) + R\xi(k), \end{cases} \tag{30}$$

where $C_z = \begin{bmatrix} \varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $R = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. By defining

$$\begin{aligned} \xi_e(k) &= \begin{bmatrix} \xi(k-d+1) & \xi(k-d+2) & \dots & \xi(k-1) \end{bmatrix}^\top \\ \nu_e(k) &= \begin{bmatrix} \nu(k-d+1) & \nu(k-d+2) & \dots & \nu(k-1) \end{bmatrix}^\top \\ x_e(k) &= \begin{bmatrix} x(k)^\top & \xi(k-d) & \xi_e(k)^\top & \nu(k-d) & \nu_e(k)^\top \end{bmatrix}^\top, \end{aligned}$$

we can write (30) as

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \xi(k-d+1) \\ \xi_e(k+1) \\ \nu(k-d+1) \\ \nu_e(k+1) \end{bmatrix} &= \begin{bmatrix} A & B & 0 & E & 0 \\ 0 & 0 & e_1^\top & 0 & 0 \\ 0 & 0 & \Omega & 0 & 0 \\ 0 & 0 & 0 & 0 & e_1^\top \\ 0 & 0 & 0 & 0 & \Omega \end{bmatrix} \begin{bmatrix} x(k) \\ \xi(k-d) \\ \xi_e(k) \\ \nu(k-d) \\ \nu_e(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0_{n,n} \\ 0 \\ e_d \\ 0 \\ 0_d \end{bmatrix} \xi(k) + \begin{bmatrix} 0_{n,n} \\ 0 \\ 0_d \\ 0 \\ e_d \end{bmatrix} \nu(k) \\ z(k) &= \begin{bmatrix} C_z & 0_{2,d}^\top & 0_{2,d}^\top \end{bmatrix} x_e(k) + R\xi(k), \end{aligned}$$

where $e_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^\top$, $e_d = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^\top$ and

$$\Omega = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^d \times \mathbb{R}^d.$$

This dynamics can be summarized in the form

$$\begin{cases} x_e(k+1) &= A_dx_e(k) + B_d\xi(k) + E_d\nu(k) \\ z(k) &= C_dx(k) + R\xi(k), \end{cases} \tag{31}$$

which directly parallels the form (12). The design described in the previous sections on system (12) shall actually be applied to this lifted system to compute a controller that takes into account the delays.

V. Numerical results

We have implemented our dynamical model and our controller using the software[1] Matlab/Simulink. We simulated platoons of up to 30 vehicles (as described below), showing consistent results, as expected from the inherent scalability of the proposed distributed controller. The implementation of each vehicle and its controller includes multiple features that have been disregarded during the design, such as transmission delays, input delays and measurement noise. The vehicle dynamics are defined according to (1) and simulated in continuous time, whereas the controller is digital as described in Section IV. The solver used is `ode23s`, as the large number of delays in the system leads to chattering if a non-stiff solver is used. The dynamical parameters are set as $\tau = 0.1s$, input delay $\phi = 0.2s$, transmission delay $\theta = 0.02s$, and measurement delay $\psi = 0.05s$. The sampling rate used by the digital controller and the communication system is $T_s = 0.01s$.

[1] The relevant Simulink code is available at http://www.gipsa-lab.fr/~paolo.frasca/docs/updated_simulink_code_Acciani2020.zip.

In our default set of parameters, we set the time-headway $h = 0.25$, though lower and higher values have also been tested. The deterministic $\mathscr{H}_\infty$ problem is defined with $\varepsilon = 10^{-1}$ and $r = 1$ and is solved by exploiting the continuous-time equivalence presented in [41] and using the continuous Riccati equation solver in Matlab. In the covariance minimization, the approximation (27) is used with $g = 0.9734$. After lifting, the system has order 43: this figure is relatively high but still compatible with the efficient solvers that we use for design. Each packet transmitted by one vehicle to the following one is received with probability $1 - p$.

In order to test the ability of our controller to ensure string stability, we simulate the following dynamic scenario. At time zero, all vehicles start at rest. The first vehicle of the platoon is connected to a *virtual* leader vehicle: the latter is a Simulink block that creates the trajectory to be followed by the platoon, by linearly ramping up in speed from 0 to $17m/s$, then keeping constant speed. In this dynamic scenario, we first test our controller in the case when there are no losses ($p = 0$). Fig. 4 illustrates the string stable behavior that is obtained in this case: observe how the input to each vehicle monotonically becomes smaller in the downstream direction. Our tests show that the controller can handle headways at least as small as 0.2, which matches state-of-the-art performance [11].

After verifying the good performance when there are no losses, we simulate the controller when each packet has a positive probability p of not being received. We observe that the switching controller designed in the previous section is very good at stabilizing the platoon in presence of losses. Its performance is robust to small measurement noise and uncertainties in the system parameters, such as τ and p, which are assumed to be known for design. The control inputs for a simulation with $p = 0.8$ are displayed in Fig. 5a, showing good string stability despite the high loss probability (input curves are smoother for smaller p). When the loss probability becomes even higher, performance begins to deteriorate and for a probability like $p = 0.9$ string stability is degraded (Fig. 6a). Note however that even when performance deteriorates at the level of the control inputs, our controller is nevertheless able to guarantee a smooth and coherent platoon motion without velocity oscillations or vehicle collisions: this positive feature remains true irrespectively of the number of vehicles (Fig. 7).

Finally, simulation results demonstrate that our switching architecture with covariance minimization is essential to achieve these good results. To make this fact apparent, we have performed some simulations without covariance minimization, that is, using a non-switching controller (10) with nominal gain matrices $\overline{F}$ and $\overline{L}$ and replacing any lost value of v by the most recent value received in the past. By design, this non-switching controller is such that the resulting closed-loop systems is string stable in expectation (where the expectation is taken over the random delays that are induced by the packet losses). This property might let us hope for a good performance, but in fact the realized trajectories can be far from string stable. The impressive improvements brought by the covariance minimizing controller are evident by comparing Fig. 5b against Fig. 5a and Fig. 6b against Fig. 6a, respectively.

VI. Conclusion

In this paper, we have addressed the question of designing a distributed cooperative controller that can robustly stabilize a platoon of vehicles when the communication is affected by losses. This has been achieved by a switching controller that has been designed with a twofold objective: promoting plant stability and string stability of the average dynamics, and promoting trajectories to be close to the average. The former objective has been sought by applying $\mathscr{H}_\infty$ control tools to the average dynamics, while the latter has been sought by minimizing the variance of the trajectories.

Even though the $\mathscr{H}_\infty$ approach of minimizing the ratio (9) does not (strictly speaking) guarantee that the string stability condition (8) is met for a given value of h, minimizing the ratio makes it less than one when there exists a controller of the form $\xi(k) = \overline{F}x(k) + \overline{L}v(k)$ that is capable of achieving string stability. In other words, we can say that if the system without losses can be made string stable by a controller, then our controller will make it string stable.

A key point in our work has been the stochasticity of the communication and the consequent need to cope with it. In our design solution, we have chosen to focus on the first two moments: the first moment to ensure a good average behavior and the second moment to make such good behavior likely. In designing the control for the average, the choice of combining state feedback with an unknown input observer has two advantages. First, it keeps the $\mathscr{H}_\infty$ problem tractable; second, it provides an avenue to refine our control design. Indeed, the unknown input observer can be used to produce an estimate of the input v, which could then replace the communicated value when the latter is unavailable. This design option has been tested in a frequency domain approach in [30] with positive results.

Thanks to minimizing the variance among the trajectories, our controller shows very good ability to cope with high levels of losses without degrading the platooning performance in terms of the headway h. This feature constitutes an improvement upon previous works that simply aimed at a graceful degradation of performance in presence of losses [11]. Future work could be devoted to quantify the maximum level of affordable losses, to more thoroughly investigate the robustness of our design to uncertainty and heterogeneity of the parameters, such as τ and p, and possibly to propose robust or adaptive approaches [36]. Future work should also be devoted to refine our way of dealing with randomness, in order to provide probabilistic guarantees on the behaviors of the possible trajectories. In the context of analysis, relevant definitions and results have been recently given by [8], with the notion of $n\sigma$ string stability (that is, that all trajectories within a neighborhood the average of radius n times the standard deviation are string stable).

Acknowledgement

The authors would like to thank Elham Semsar-Kazerooni for insightful discussions on the topics of this paper.

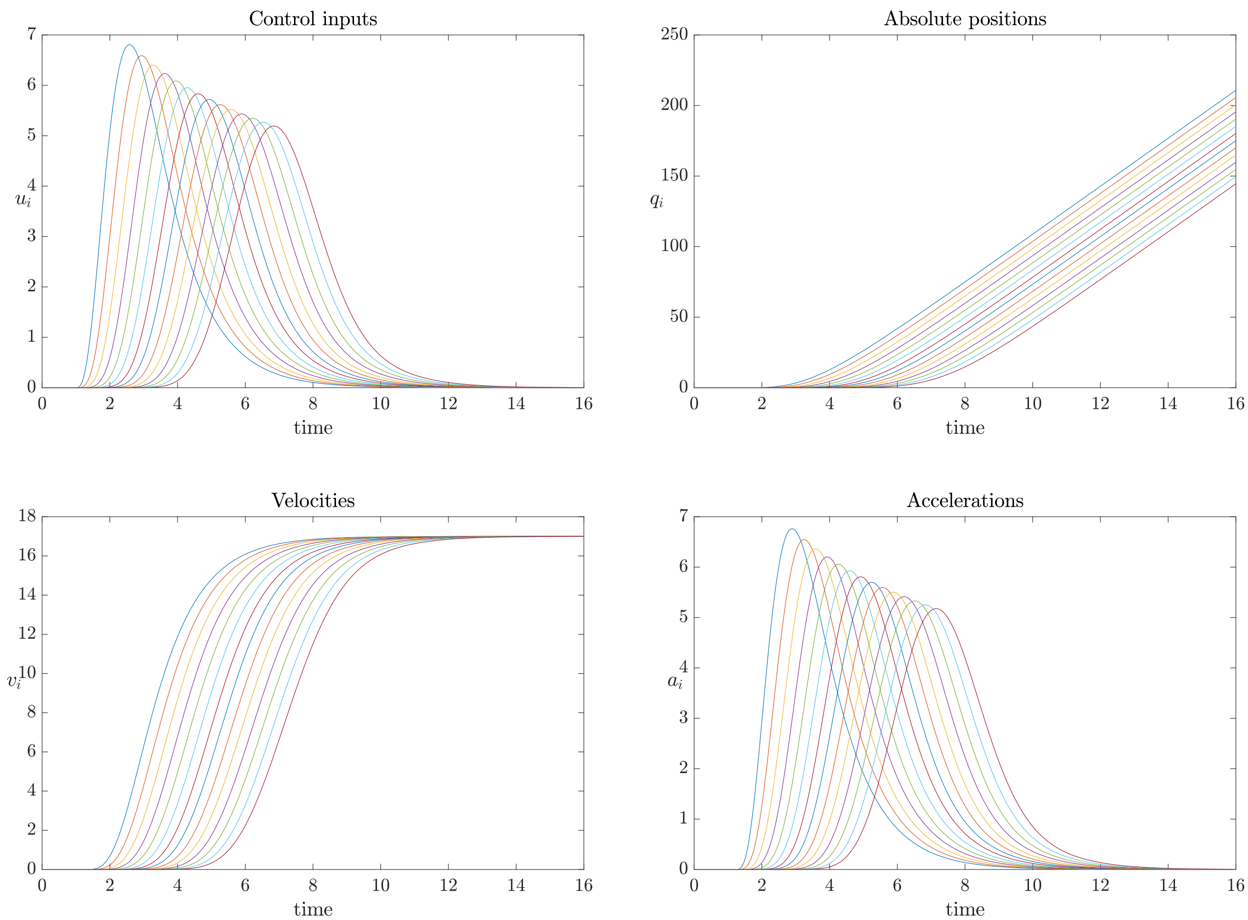
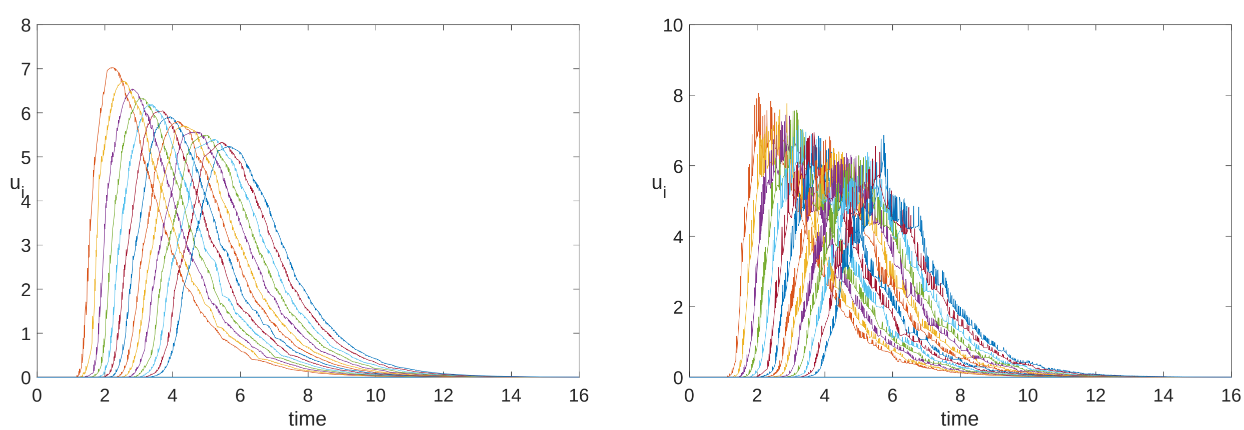


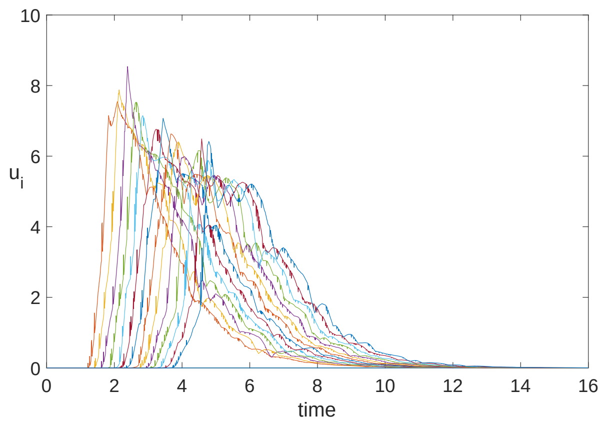
Fig. 4: Time evolutions of inputs, positions, velocities, and accelerations for a controlled platoon of 14 vehicles when no losses are present.



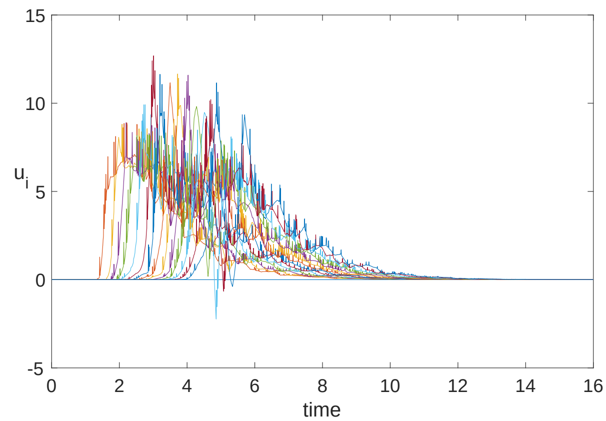
(a) Minimizing covariance (switching) controller

(b) Non-switching controller, without minimizing covariance

Fig. 5: Control inputs for a platoon of 14 vehicles with $p = 0.8$. Observe how the switching minimizing-covariance controller outperforms the constant controller without covariance minimization.



(a) Minimizing covariance (switching) controller



(b) Non-switching controller, without minimizing covariance

Fig. 6: Control inputs for a platoon of 14 vehicles with $p = 0.9$. Observe how the switching minimizing-covariance controller outperforms the constant controller without covariance minimization.

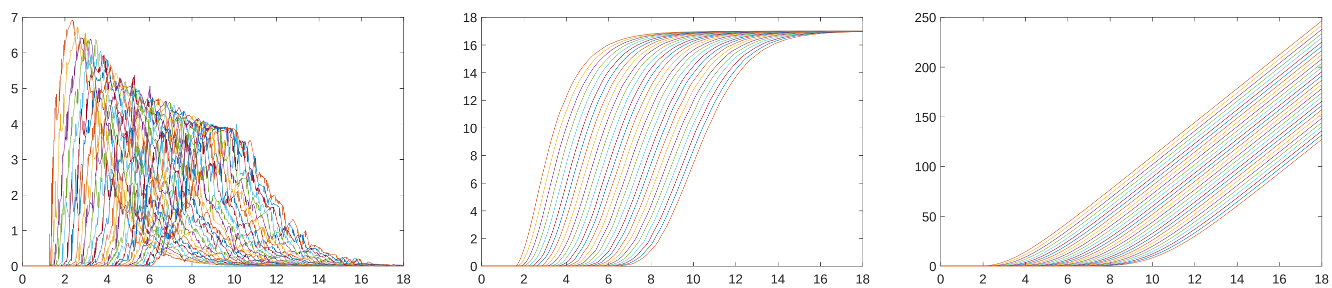


Fig. 7: Inputs, velocities, and positions (from left to right) in a sample simulation with $p = 0.9$, $h = 0.25$ and $n = 30$.

References

[1] G. Gunter, D. Gloudemans, R. E. Stern, S. McQuade, R. Bhadani, M. Bunting, M. L. D. Monache, R. Lysecky, B. Seibold, J. Sprinkle, B. Piccoli, and D. B. Work, "Are commercially implemented adaptive cruise control systems string stable?" *IEEE Trans. on Intelligent Transportation Systems*, pp. 1–12, 2020.

[2] F. J. Vargas, A. I. Maass, and A. A. Peters, "String stability for predecessor following platooning over lossy communication channels," in *International Symposium on Mathematical Theory of Networks and Systems*, 2018.

[3] X. Liu, S. Sonia Mahal, A. Goldsmith, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," *IEEE Transaction on Intelligent Transportation Systems*, 2001.

[4] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, 2006.

[5] R. Pates, C. Lidström, and A. Rantzer, "Control using local distance measurements cannot prevent incoherence in platoons," in *56th IEEE Annual Conference on Decision and Control (CDC 2017)*, Dec. 2017, pp. 3461–3466.

[6] C. Lei, M. Eenennaam, W. Klein Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg, "Impact of packet loss on CACC string stability performance," in *11th International Conference on ITS Telecommunications (ITST)*, 2011, pp. 381–386.

[7] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, 2014.

[8] W. B. Qin, M. M. Gomez, and G. Orosz, "Stability and frequency response under stochastic communication delays with applications to connected cruise control design," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 2, pp. 388–403, 2017.

[9] S. Santini, A. Salvi, A. Valente, A. Pescapé, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Trans. on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, 2017.

[10] C. Zhao, X. Duan, L. Cai, and P. Cheng, "Vehicle platooning with non-ideal communication networks," *IEEE Trans. on Vehicular Technology*, vol. 70, no. 1, pp. 18–32, 2021.

[11] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cooperative adaptive cruise control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 488–497, Feb 2015.

[12] G. Giordano, M. Segata, F. Blanchini, and R. Lo Cigno, "The joint network/control design of platooning algorithms can enforce guaranteed safety constraints," *Ad Hoc Networks*, vol. 94, p. 101962, 2019.

[13] R. Merco, F. Ferrante, and P. Pisu, "A hybrid controller for dos-resilient string-stable vehicle platoons," *IEEE Trans. on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1697–1707, 2021.

[14] L. Peppard, "String stability of relative-motion PID vehicle control systems," *IEEE Trans. on Automatic Control*, vol. 19, no. 5, pp. 579–581, October 1974.

[15] A. Pant, P. Seiler, and J. K. Hedrick, "Mesh stability of look-ahead interconnected systems," *IEEE Trans. on Automatic Control*, vol. 47, no. 2, pp. 403–407, 2002.

[16] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE Trans. on Automatic Control*, vol. 41, no. 3, pp. 349–357, 1996.

[17] Y. A. Harfouch, S. Yuan, and S. Baldi, "An adaptive switched control

approach to heterogeneous platooning with intervehicle communication losses," *IEEE Trans. on Control of Network Systems*, vol. 5, no. 3, pp. 1434–1444, 2018.

[18] T. G. Molnár, W. B. Qin, T. Insperger, and G. Orosz, "Application of predictor feedback to compensate time delays in connected cruise control," *IEEE Trans. on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 545–559, Feb 2018.

[19] R. Merco, F. Ferrante, and P. Pisu, "Network Aware Control Design for String Stabilization in Vehicle Platoons: An LMI Approach," in *American Control Conference 2019*, Philadelphia, USA, Jul. 2019.

[20] V. K. Vegamoor, D. Kalathil, S. Rathinam, and S. Darbha, "Reducing time headway in homogeneous CACC vehicle platoons in the presence of packet drops," in *European Control Conf.*, June 2019, pp. 3159–3164.

[21] W. B. Qin, M. M. Gomez, and G. Orosz, "Moment-based plant and string stability analysis of connected cruise control with stochastic delays," in *2015 American Control Conference (ACC)*, July 2015, pp. 2747–2752.

[22] V. S. Dolk, J. Ploeg, and W. P. M. H. Heemels, "Event-triggered control for string-stable vehicle platooning," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3486–3500, Dec 2017.

[23] A. A. Stoorvogel, *The H_∞ control problem: a state space approach*. New York: Prentice-Hall, 1992.

[24] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer, "Controller synthesis for string stability of vehicle platoons," *IEEE Trans. on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 854–865, April 2014.

[25] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed H-infinity controller synthesis," *IEEE Trans. on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1353–1364, 2018.

[26] J. G. Li, J. Q. Yuan, and J. G. Lu, "Observer-based H_∞ control for networked nonlinear systems with random packet losses," *ISA Trans.*, vol. 49, no. 1, pp. 39–46, 2010.

[27] Z.-M. Li and X.-H. Chang, "Robust H_∞ control for networked control systems with randomly occurring uncertainties: Observer-based case," *ISA Trans.*, vol. 83, pp. 13–24, 2018.

[28] P. Seiler and R. Sengupta, "An H_∞ approach to networked control," *IEEE Trans. on Automatic Control*, vol. 50, no. 3, pp. 356–364, March 2005.

[29] ——, "Analysis of communication losses in vehicle control problems," in *Proceedings of the 2001 American Control Conference*, vol. 2, June 2001, pp. 1491–1496.

[30] F. Acciani, P. Frasca, A. Stoorvogel, E. Semsar-Kazerooni, and G. Heijenk, "Cooperative adaptive cruise control over unreliable networks: An observer-based approach to increase robustness to packet loss," in *European Control Conf.)*, Limassol, Cyprus, Jun. 2018.

[31] S. K. Yadlapalli, S. Darbha, and K. R. Rajagopal, "Information flow and its relation to stability of the motion of vehicles in a rigid formation," *IEEE Trans. on Automatic Control*, vol. 51, no. 8, pp. 1315–1319, 2006.

[32] G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-stable CACC design and experimental validation: A frequency-domain approach," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 9, pp. 4268–4279, Nov 2010.

[33] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *American Control Conference (ACC)*, 1990, pp. 291–296.

[34] S. Baldi, M. R. Rosa, P. Frasca, and E. B. Kosmatopoulos, "Platooning merging maneuvers in the presence of parametric uncertainty," *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 148–153, 2018.

[35] M. D. Ankem and S. Darbha, "Effect of heterogeneity in time headway on error propagation in vehicular strings," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2612–2619.

[36] S. Baldi, S. Yuan, and P. Frasca, "Output synchronization of unknown heterogeneous agents via distributed model reference adaptation," *IEEE Trans. on Control of Network Systems*, vol. 6, no. 2, pp. 515–525, June 2019.

[37] D. Liu, S. Baldi, V. Jain, W. Yu, and P. Frasca, "Cyclic communication in adaptive strategies to platooning: the case of synchronized merging," *IEEE Trans. on Intelligent Vehicles*, pp. 1–1, 2020.

[38] D. Swaroop, J. Hedrick, C. C. Chien, and P. Ioannou, "A comparision of spacing and headway control laws for automatically controlled vehicles," *Vehicle System Dynamics*, vol. 23, pp. 597–625, 1994.

[39] J. Ploeg, B. T. M. Scheepers, E. Nunen, N. Van De Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2011.

[40] S. Öncü, "String stability of interconnected vehicles: network-aware modelling, analysis and experiments," Ph.D. dissertation, Technische Universiteit Eindhoven, 2014.

[41] B. Chen, A. Saberi, and Y. Shamash, "A nonrecursive method for solving the general discretetime Riccati equations related to the H_∞ control problem," *International Journal of Robust and Nonlinear Control*, vol. 4, no. 4, pp. 503–519, 1994.

[42] P. Frank, "Robust model-based fault detection in dynamic systems," *IFAC Proceedings Volumes*, vol. 25, no. 4, pp. 1–13, 1992, IFAC Symposium on On-line Fault Detection and Supervision in the Chemical Process Industries, Newark, Delaware, 22-24 April.

[43] A. Popescu, G. Besancon, and A. Voda, "A new robust observer approach for unknown input and state estimation," in *2018 European Control Conf.*, Limassol, Cyprus, Jun. 2018.

Francesco Acciani obtained his Master and Bachelor degrees in Automation Engineering from Politecnico di Bari, Bari, Italy, in 2014 and 2011. He is currently a PhD candidate at the university of Twente, Enschede, the Netherlands. His research interests are in cooperative control over unreliable networks, consensus, and applications of graph theory.

Paolo Frasca received the Ph.D. degree in Mathematics for Engineering Sciences from Politecnico di Torino, Turin, Italy, in 2009. From 2013 to 2016, he has been an Assistant Professor at the University of Twente in Enschede, the Netherlands. Since October 2016 he is CNRS Researcher with GIPSA-lab, Grenoble, France. His research interests are in the theory of networks, learning, and control systems, with applications to robotic, infrastructural and social networks.

Geert Heijenk is a full professor in Wireless Networks and Mobility at the University of Twente, the Netherlands. He has held a part-time position at KPN Research from 1989 until 1991. From 1995 until 2003, he was with Ericsson EuroLab Netherlands, leading a networking research group on Wireless Internet Technologies. From 2015 until 2018 he has been program director of the Computer Science and Internet Science & Technology bachelor and master programs of University of Twente. The main research area of Geert Heijenk is network architectures and resource management for wireless communication networks, such as (beyond) 5G networks. Next to his scientific publication record, he holds several patents in the area of wireless networks. Geert Heijenk is steering committee member of IFIP WWIC (International Conference on Wired/Wireless Internet Communications) and IEEE VNC (Vehicular Networking Conference).

Anton A. Stoorvogel received the M.Sc. degree in Mathematics from Leiden University in 1987 and the Ph.D. degree in Mathematics from Eindhoven University of Technology, the Netherlands in 1990. Currently, he is a professor in systems and control theory at the University of Twente, the Netherlands. Anton Stoorvogel is the author of five books and numerous articles.