Cooperative adaptive cruise control over unreliable networks: an observer-based approach to increase robustness to packet loss

Francesco Acciani, Paolo Frasca, Anton Stoorvogel, Elham Semsar-Kazerooni and Geert Heijenk

Abstract— Cooperative Adaptive Cruise Control (CACC) is nowadays a promising technique to increase highway through-put, safety and comfort for vehicles. Enabled by wireless communication, CACC allows a platoon of vehicles to achieve better performance than Adaptive Cruise Control; however, since wireless is employed, problems related to unreliability arise. In this paper, we design a digital controller to achieve platoon stability, enhanced by an observer to increase robustness against packet losses. A preliminary set of simulation results is presented, which confirms the interest of using an observer in combination with a local and cooperative digital controller.

I. INTRODUCTION

Vehicle platooning is a promising way to optimise the use of highways by reducing the distance between vehicles and thus increasing the throughput of the road. One effective way to implement such a strategy is the use of Adaptive Cruise Control (ACC), where the acceleration of a vehicle is controlled automatically to achieve a desired speed while maintaining a prescribed distance from the previous vehicle, using radar or lidar measurements [1], [2]. Moreover, such approach is known to reduce fuel consumption and provide more comfort to the user, compared to the human control of vehicles [3]. By allowing the vehicles to share their motion and control information through wireless communication, the minimum allowable inter-vehicular time headway can be decreased [4], [5]. However, the wireless medium that is used to share this information is unreliable. Therefore, in this cooperative approach, known as Cooperative Adaptive Cruise Control (CACC), the influence of the packet loss should be taken into account for a safe and comfortable behaviour.

Previous works focussed on the effect of delays on the performance of CACC, providing analysis of the effect of constant delays on the string stability properties of a vehicle platoon [6], [7] [8], while little work has been conducted on the effect of packet loss over the stability performance of a platoon [9]. Some authors suggest to deal with losses at network-level, by modifying the communication protocol between vehicles [10] or switch to an ACC scheme when the network performance become worse [11].

Towards this goal, the aim of this work is to design a controller that: 1) stabilises the single vehicle, allowing for a zero-error speed regulation, 2) stabilises the vehicle platoon, reducing the amplification of disturbances over the string [12] and 3) is robust to communication packet loss. To achieve these objectives, two digital controllers will be designed: a local controller that stabilises a single vehicle and a cooperative controller that improves the performance of the platoon, reducing the minimum allowable inter-vehicle time headway. The cooperative controller uses as its input the output provided by an observer, which estimates the current input (transmitted by the other vehicles in the platoon). The observer is always in-use, in contrast with other works that resorted to use an observer only in presence of heavy losses [11]. We use digital controllers because the communication between vehicles inherently introduces sampling phenomena; moreover a suitable choice of the sampling time in the digital framework allows us for a simpler model of time delays, as detailed in Section 2.

The contribution of this paper is twofold: 1) we design a local digital controller and a cooperative one to achieve stable behaviour for a platoon of vehicles, and 2) we explore the use of an observer to reduce the impact of losses on the disturbance attenuation over the string.

The paper is organised as follows: in Section 2 the control problem and the vehicle model are presented. In Section 3 the local controller, the cooperative controller and the observer are designed, and in Section 4 our simulation results are commented on. Finally, in Section 5 our conclusions are presented.

II. PROBLEM FORMULATION

A way to increase road throughput is to allow vehicles to drive as close as possible to each other, while keeping the platoon dynamics stable. In order to do so, a speed-dependent time spacing policy is adopted [13], i.e. the required distance between vehicle i and its preceding vehicle $i-1$ is determined as:

$$d_{r,i}(t) = r_i + hv_i(t) \qquad (1)$$

where h is the time headway, $v_i(t)$ is the speed of the ith vehicle, and r_i is the standstill distance. As this work focusses mostly on the effect of disturbances in a platoon of vehicles, r_i is assumed to be zero for simplicity: by a change of variables, it is possible to prescribe any standstill distance, without affecting the stability properties of the platoon. The

F. Acciani is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands f.acciani@utwente.nl

P. Frasca is with Univ. Grenoble Alpes, CNRS, Inria, GIPSA-lab, F-38000 Grenoble, France paolo.frasca@gipsa-lab.grenoble-inp.fr

A. Stoorvogel is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands a.a.stoorvogel@utwente.nl

E. Semsar-Kazerooni is with The Netherlands organisation for Applied research (TNO), Automotive campus 30, Helmond, The Netherlands. elham.semsarkazerooni@tno.nl

G. Heijenk is with Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, 7500 AE Enschede, The Netherlands geert.heijenk@utwente.nl

spacing policy in (1) is known to improve string stability and contribute to safety [1]. The objective of the controller is to bring the distance between vehicles to the reference distance in (1), while guaranteeing a stable behaviour for the platoon and achieve speed regulation, i.e. $v_i(t) \rightarrow v_r$. The value v_r is known by the platoon leader, but is not transmitted to the other vehicles. We assume that each vehicle is equipped with a radar sensor, to measure the distance from the previous vehicle, as well as a speed sensor, to measure its own absolute velocity. The distance and absolute velocity are referred to as the local measurements available to each car in the remainder of this paper, these are the signals used by the local controller to drive the vehicle. The cooperative controller receives the control input of the previous vehicle, transmitted over the wireless medium.

The local measurements are noisy and suffer from delays. On the other hand, the control input of the previous vehicle is a noise-free quantity, which suffers – when the communication channel is reliable – only from transmission delay; the latter is smaller than the measurement and the internal delays of the vehicles. This allows to obtain an estimate of the preceding vehicles state variables faster compared to local measurements. However, when the communication happens over the wireless channel, every packet has a probability of not being received. To improve robustness, the input of the previous car is estimated by an observer, which takes as input both the unreliably transmitted input and the local measurements.

As general model for the longitudinal vehicle dynamics we use the following from [14]:

$$\dot{a}_i(t) = -\frac{1}{\tau} a_i(t) + \frac{1}{\tau} u_i(t - \phi), \qquad (2)$$

where $\dot{a}_i(t)$ is the derivative of the acceleration of the ith vehicle. In (2), ϕ represents an internal delay of the system.

It is possible to study the single vehicle in the frequency domain, by computing the system transfer functions for position and speed and then convert the continuous-time dynamics to a discrete-time one. To prevent aliasing errors, the sampling time has to be sufficiently smaller than τ. We choose the sampling time as $T_s = \frac{\tau}{10}$. The transfer function approach can easily take delays into account, when the delay is a multiple integer of the sampling time. However, when the sampling time becomes too small, the order of the model with delays grows, and this motivates the reasonable choice for T_s, which shall not be too small to prevent high order model. The model of the single vehicle is then used to find a controller, to both stabilise the vehicle dynamics and achieve disturbance attenuation over the platoon.

III. Controller design

In this section a local and a cooperative controller are designed, and after this, an observer is designed to increase robustness to packet loss. The objective of the controller is threefold: 1) stabilise the dynamics of each single vehicle, 2) bring the distance between vehicles to the desired distance given in (1), and 3) avoid disturbance amplification along the platoon.

A. Local controller

The first objective of the controller is to stabilise the system and bring the error e_i to zero, where:

$$e_i(k) = d_i(k) - d_{r,i}, \qquad (3)$$

while $d_i(k)$ is the distance from vehicle i to vehicle $i - 1$, and $d_{r,i}$ is the reference distance given in (1). We use k to denote the discrete time framework. We first find the transfer function between the input of the vehicle u_i and the output y_i defined as:

$$y_i(k) = q_i(k) + h v_i(k), \qquad (4)$$

By defining the transfer function between the input u_i and the position q_i as $G_p(z)$ and between u_i and v_i as $G_v(z)$, the transfer function between the input u_i and the output y_i is:

$$P(z) = G_p(z) + h G_v(z). \qquad (5)$$

The control objective is to design a closed-loop transfer function $W_0(z) = \frac{y_i(z)}{q_{i-1}(z)}$ to achieve asymptotic stability and input tracking.

Firstly, we want asymptotic trajectory tracking. We assume that the $(i-1)^{th}$ vehicle proceeds at constant speed, i.e its absolute position $q_{i-1}(z)$ is a ramp:

$$q_{i-1}(z) = \frac{T_s z}{(z-1)^2}. \qquad (6)$$

This assumption is done because the objective of the controller is to implement a *speed* control: the platoon receives as input a target speed, and all the vehicles must reach the reference speed, keeping a distance between them defined by the error (3). Asymptotically, the trajectories are then ramp signals.

The error transfer function is then:

$$E_i(z) = \frac{e_i(z)}{q_{i-1}(z)} = \frac{q_{i-1}(z) - y_i(z)}{q_{i-1}(z)} = 1 - W_0(z).$$

By using the final value theorem, recalling that the input is a ramp, we find conditions to achieve asymptotic tracking, i.e $\lim_{k \to \infty} e_i(k) = 0$:

$$\lim_{z \to 1} \frac{T_s}{(z-1)} (1 - W_0(z)) = 0 \qquad (7)$$

To avoid critical zero-pole cancellation, we need to impose that the unstable zeros of $P(z)$ are roots of $W_0(z)$ and that the unstable poles of $P(z)$ are roots of $1 - W_0(z)$.

Since $P(z)$ has a double pole in 1 for every value of h, we need to impose the following conditions:

1) $W_0(1) = 1$ to avoid zero-pole cancellation and impose condition (7) and
2) $\frac{d}{dz}(1 - W_0(z))\big|_{z=1} = 0$ to avoid zero-pole cancellation

Then, $W_0(z)$ is designed as:

$$W_0(z) = \frac{w_1 z + w_0}{z(z^2 + p_1 z + p_2)} \qquad (8)$$

where the two parameters w_1 and w_0 are chosen to impose the conditions 1) and 2) in the list, the relative degree
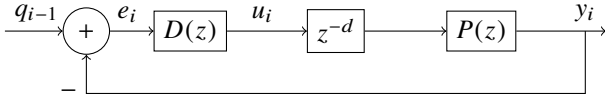
Fig. 1: Delayed plant block diagram

of $W_0(z)$ is chosen to be the same as the relative degree of $P(z)$, and the parameters p_1 and p_2 are chosen such that $W_0(z)$ has the same nonzero poles as the discrete time equivalent of:

$$W(s) = \frac{1}{s^2 + 2\delta\omega_n + \omega_n^2} \tag{9}$$

where $\delta = \frac{\sqrt{2}}{2}$ and $\omega_n = \frac{\sqrt{2}}{2\tau}$. This choice for δ yields the fastest response time, subject to the condition of presenting no peaks in the frequency domain. The choice of ω_n influences the speed of the system: it must be high enough to provide satisfactory performance, but can not be too large, in order to achieve stable behaviour. Converting (9) in the digital time domain allows us to find the values for p_1 and p_2, and then we can impose conditions 1) and 2) to obtained $W_0(z)$ and the controller transfer function $D(z)$ as:

$$D(z) = \frac{1}{P(z)} \frac{W_0(z)}{1 - W_0(z)}.$$

The local controller design process has been conducted without taking the input delay of the original system into account. It is possible, after the controller design, to modify the controller to minimise the effects of delay on the stability, by using a Smith predictor. The input delay is typically $\phi = 0.2s$, which reflects in the block z^{-d} in Figure 1 where $d = \frac{\phi}{T_s}$ is the number of samples of delay introduced in the system. Since $d = 20$, it would have been difficult to design a controller while taking the delay effects into account.

The controller that is actually implemented is then the following:

$$\hat{D}(z) = \frac{D(z)}{1 + D(z)P(z)(1 - z^{-d})}. \tag{10}$$

As there are no critical pole-zero cancellation between $D(z)$ and $P(z)$ from the design of the original $D(z)$, the use of $\hat{D}(z)$ does not cause any stability issue, therefore the local controller design is complete.

B. Cooperative controller

The controller in Section 3.A is purely local, relying only on measurement of the error (3), which is available to each vehicle. It is possible to enhance the controller performance by using a feedforward compensation term, as in Figure 2. The diagram in Figure 2 is equivalent to the interconnection of a Smith predictor and the system with delays.

Imposing a perfect reference tracking, we obtain as feedforward filter:

$$\frac{y_i(z)}{q_{i-1}(z)} = 1 = \frac{F_q(z)P(z) + D(z)P(z)}{1 + P(z)D(z)} \implies F_q(z) = \frac{1}{P(z)}.$$

To use this filter it is necessary to know $q_{i-1}(k)$, which is not available to the ith vehicle. However, because $q_{i-1}(z) =$

$G_p(z)u_{i-1}(z)$, we can substitute it in the filter and then the feedforward term becomes:

$$F(z) = \frac{G_p(z)}{P(z)}$$

so that the feedforward quantity is now the input of the previous vehicle $u_{i-1}(k)$. Moreover $F(z)$ is minimum phase, in contrast with $P(z)$, whose inverse is not causal.
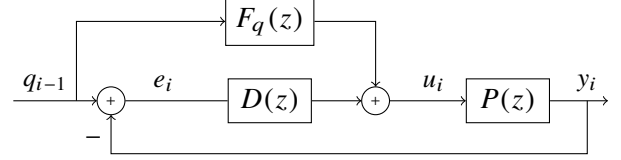

Fig. 2: (Position) feedforward block diagram

C. Packet loss and observer

The control scheme developed so far is made by two components: a locally stabilising term $\hat{D}(z)$ and a feedforward one $F(z)$. The latter controller relies on perfect knowledge of $u_{i-1}(k)$, which can not be achieved because of the communication. The communication induces a transmission delay, denoted in the following by θ, losses, and might happen at a different transmission rate than the controller. To cope with imperfections in the communication channel, each vehicle estimates the input of its preceding vehicle from the local measurements. To devise an observer, we consider the extended block diagram in Figure 3, where the relation between $q_{i-1}(z)$ and $u_{i-1}(z)$ and the transmission delay θ is made explicit. The delay θ is smaller than the input delay d, and this motivates the use of the observer: it is possible to have an estimate of the error e_i before a local measurement is available.

The observer uses the local measurements (distance and speed) to produce an estimate $\hat{u}_{i-1}$ of the input of the previous vehicle. However, when the actual input is successfully received, the state of the observer is reset to the just received value.

The complete block diagram when the observer is employed is depicted in Figure 4 (the blocks $H(z)$ and $A(z)$ will be explained later). We first compute the error transfer function $E_i(z) = \frac{e_i(z)}{u_{i-1}(z)}$. We then use $E_i(z)$ to compute an estimate of the error, $\hat{e}_i(z) = E_i(z)\hat{u}_{i-1}$, where $\hat{u}_{i-1}$ is the output of the observer. By using a controller $L(z)$, we drive to zero the estimation error $\varepsilon_i = e_i(z) - \hat{e}_i(z)$. Asymptotically, $\hat{e}_i \to e_i$ and $\hat{u}_{i-1} \to u_{i-1}$, and the observer transfer function is:

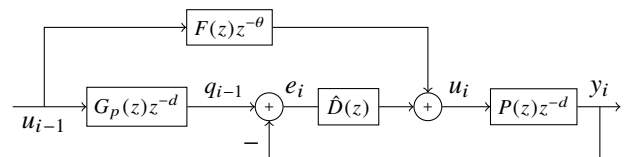$$O(z) = \frac{\hat{u}_{i-1}}{e_i} = \frac{L(z)}{1 + L(z)E_i(z)}. \tag{11}$$

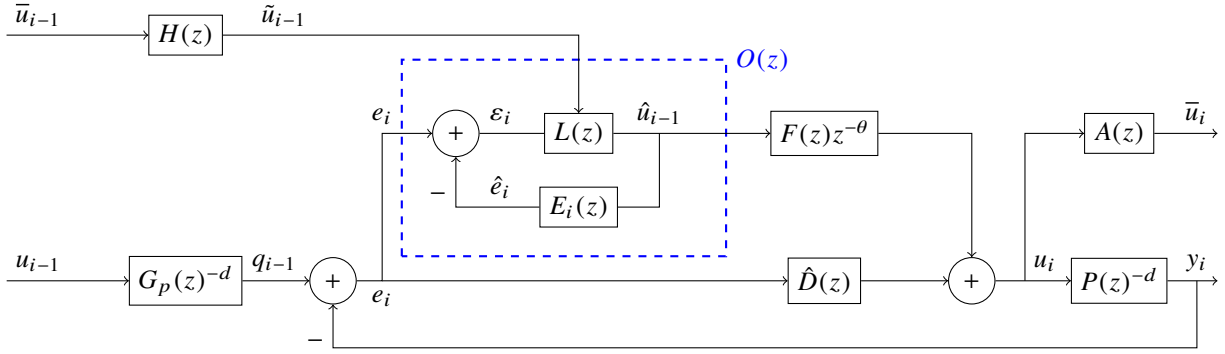Fig. 3: Delayed feedforward loop diagram

Fig. 4: Observer loop diagram

This transfer function, however, describes the observer dynamics only partially: when a signal is successfully received, the observer state jumps to the received value, inducing discontinuities that can not be captured by a frequency domain model, but which are useful to better reconstruct the signal u_{i-1}. The problem of finding the observer for u_{i-1} reduces to finding the error transfer function $E_i(z)$ and the controller $L(z)$. Clearly, performance depends on $E_i(z)$ being an accurate description of the error, as we rely on a good modelling of the error dynamics to obtain the input estimate. The rest of the section is devoted to find $E_i(z)$.

As first approximation we can find $E_i(z)$ from the block diagram in Figure 3:

$$E_i(z) = \frac{e_i(z)}{u_{i-1}(z)} = \frac{G_p(z)(1 - z^{-\theta})z^{-d}}{1 + P(z)\hat{D}(z)z^{-d}}. \tag{12}$$

The error transfer function in (12) allows for a good input reconstruction when the transmission is at the same rate as the controller ($0.01s$). However, when a more realistic case is simulated, the estimate of the input becomes worse. We want to study the system at the fastest rate between transmission and digital controller: this requires an approximation of the downsampling phenomenon.

To take the downsampling phenomenon into account we approximate its effect by using a transfer function and we modify the sending strategy of control input over the wireless medium. Referring to the block diagram in Figure 4, the block $H(z)$ represents the downsampling block and $A(z)$ is an average filter to improve the error reconstruction.

The previous vehicle's input u_{i-1}, which drives the plant $G_p(z)$, is not the same signal filtered by $F(z)$, because it is downsampled by a factor $m = 4$: the radio transmits a packet every $0.04s$. It is not possible to find a transfer function to model the downsampling phenomenon, i.e. the output signal kept constant for $m-1$ sampling instant: we propose as useful approximation an averaging filter, like the following:

$$\tilde{u}(k + 1) = \tilde{u}(k) + \frac{u(k) - u(k - m)}{m}$$

where $\tilde{u}$ is the approximation of the downsampled signal, transmitted over the wireless medium. This leads to the

following transfer function:

$$H(z) = \frac{\tilde{u}(z)}{u(z)} = \frac{1}{m} \frac{z^{-1}(1 - z^{-m})}{1 - z^{-1}} = \frac{1}{m} \sum_{j=1}^{m} z^{-j}$$

that will be used to model the downsampling in the rest of the paper.

The same filter $H(z)$ is also used to average the transmitted message, by sending an average of the previous m samples instead of only one sample. With reference to Figure 4, the signal $\bar{u}(k)$ is an average of $u(k)$, the signal $\tilde{u}(k)$ is a downsampled version of $\bar{u}(k)$, modelled as an average of $\bar{u}(k)$. The average filter transfer function $A(z)$ is the same as the downsampling one:

$$\bar{u}_i(k) = \frac{1}{m} \sum_{j=1}^{m} u_i(k - j) \Rightarrow A(z) = \frac{1}{m} \sum_{j=1}^{m} z^{-j} = H(z)$$

but while $H(z)$ is a model for the downsampling phenomenon, used only to find the error transfer function, $A(z)$ is actually implemented in the control loop. Using these approximations, the error transfer function becomes:

$$E_i(z) = \frac{G_p(z)(1 - A(z)H(z)z^{-\theta})z^{-d}}{1 + P(z)\hat{D}(z)z^{-d}}.$$

This error transfer function is of high order. It allows to estimate the error in simulation, but it can not be effectively used to design the controller $L(z)$. To run the simulations and obtain the first results, a second order low-pass filter is used as controller for $E_i(z)$:

$$L(z) = \mathcal{Z}\left\{ \frac{\omega_0^2}{s^2 + 2\delta_0\omega_0 s + \omega_0^2} \right\}. \tag{13}$$

After finding $E_i(z)$ and $L(z)$, the observer in (11) allows the system is able to achieve a satisfactory behaviour, even if the input reconstruction is not perfect and suffers from delays (which can not be avoided). The observer state is reset when a new input is received: this allows the observer to be constantly used. When the communication is perfect, without losses or downsampling, the input will be available every sampling time, and the observer output will be the transmitted one. On the other hand, when the communication starts to be unreliable and at a different rate, the observer output will be the transmitted input when it is available, or its estimate when the real value has not been received.
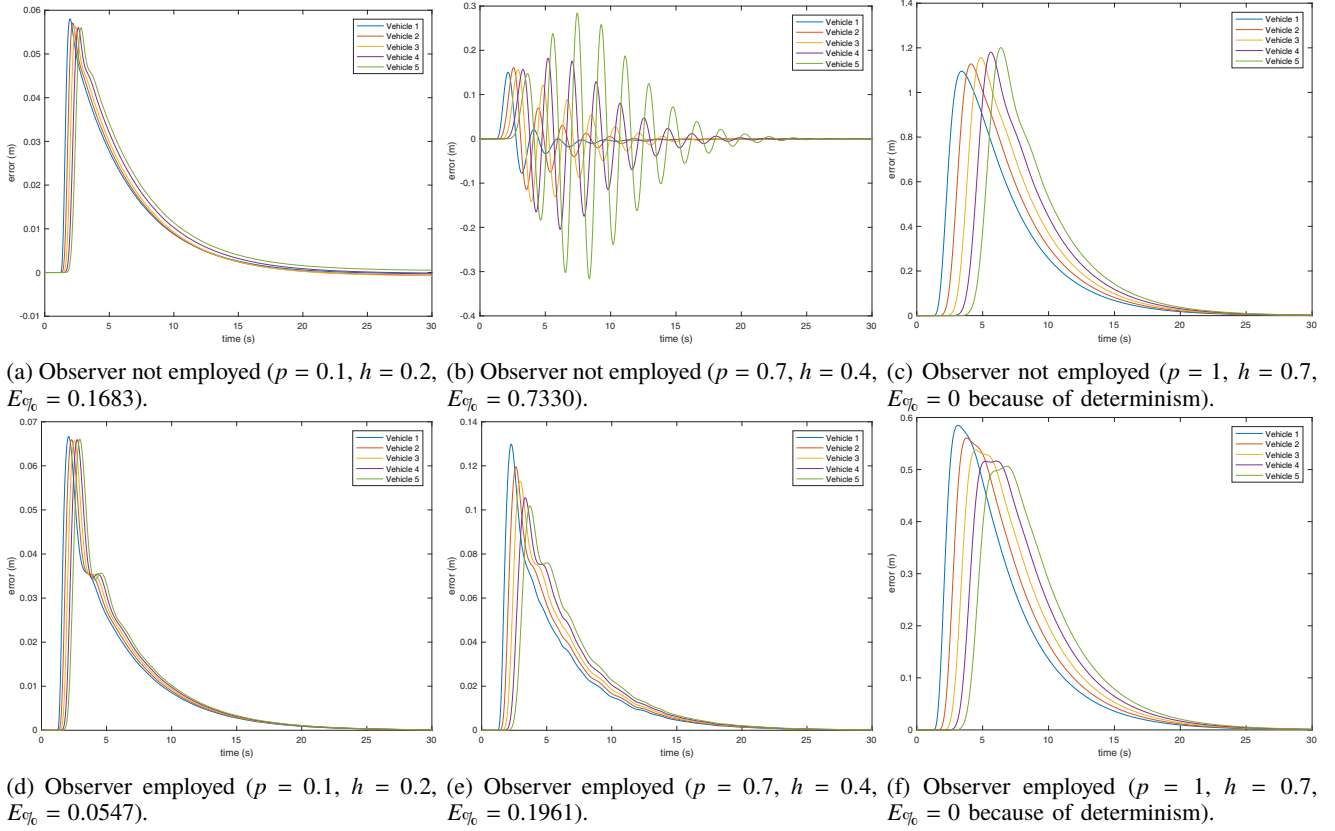
(a) Observer not employed ($p = 0.1$, $h = 0.2$, $E\%_0 = 0.1683$).

(b) Observer not employed ($p = 0.7$, $h = 0.4$, $E\%_0 = 0.7330$).

(c) Observer not employed ($p = 1$, $h = 0.7$, $E\%_0 = 0$ because of determinism).

(d) Observer employed ($p = 0.1$, $h = 0.2$, $E\%_0 = 0.0547$).

(e) Observer employed ($p = 0.7$, $h = 0.4$, $E\%_0 = 0.1961$).

(f) Observer employed ($p = 1$, $h = 0.7$, $E\%_0 = 0$ because of determinism).

Fig. 5: Error behaviour when the observer is not employed (top) and when it is employed (bottom)

IV. SIMULATION RESULTS

The objective of CACC is to achieve a stable behaviour for the vehicles, bringing the error (3) to zero, while attenuating the effects of disturbances over the platoon. When a transfer function is available to model the vehicles interconnected in the platoon, the so called *string stability* condition can be checked, by evaluating the norm of the transfer function: a peak value smaller than one in the Bode diagram assures disturbance damping. However, when packet losses are taken into account, a transfer function model does not describe the dynamics of the system satisfactorily anymore: for this reason we resort to simulations. The platoon dynamics is stochastic. We assume that every communication between vehicles can fail with a probability p, and then when this happens, the observer provides the estimate of the input of the preceding vehicle. This randomness requires a suitable definition of stable behaviour for the vehicle string. For instance, a deterministic worse-case approach is too conservative: for every probability of packet loss, the worse case is always the one without communication at all. Instead, it seems natural to require stability in the expected value case. On the other hand, we are aware that considering the average case might not be informative enough, and that further reflection is needed.

We define a behaviour to be *satisfactory* when the peaks in the average behaviour of the subsequent vehicles become smaller in the downstream direction, after a change in the reference speed for the leader vehicle, i.e.:

$$\max_k y_i(k) \geq \max_k y_{i-1}(k) \quad \forall i. \qquad (14)$$

We run our simulations using the Simulink FixedStep(ode3) solver, a sampling time of $T_s = 0.01s$, and the simulations end at $T_e = 30s$. The vehicles' time constant is $\tau = 0.1s$. We simulate a platoon of 5 vehicles and every vehicle is connected to the preceding one, which transmits its input data every $4T_s = 0.04s$: the downsampling phenomenon that has been modelled using the $H(z)$ transfer function, is fully implemented in the simulation by discarding $(m - 1)$ packets every m sampling times. The numerical value of the previous car input is then hold by each vehicle until a new packet is received, and every packet is received with probability $1 - p$ (there are no retransmissions). We simulate the behaviour of the platoon for a probability of packet loss $p \in [0.1, 0.2, \ldots, 1]$ and for a headway time $h \in [0.2, 0.3, \ldots, 1]$. The displayed results are the average of 30 simulations. The first vehicle is connected to a virtual vehicle, which has a velocity step as input: at time $t = 1s$ the reference speed is set to $v_r = 12m/s$. For the local controller, we set $\omega_n = \sqrt{2}/(2\tau)$ and the observer is the second order filter in (13). In order to assess the statistical accuracy of our results, we computed a quadratic relative error measurement for the satisfactory behaviours, defined as:

$$E\%_0 = \max_{i,j} \sqrt{\left(\frac{\sum_0^{T_e} |e_{i,j}(k) - \bar{e}_i(k)|^2}{\sum_0^{T_e} |\bar{e}_i(k)|^2} \right)} \qquad (15)$$

where $e_{i,j}$ is the error trajectory for car i during the jth simulation and $\bar{e}_i$ is the average error trajectory of car i. This quantity, which represents how much the trajectories are
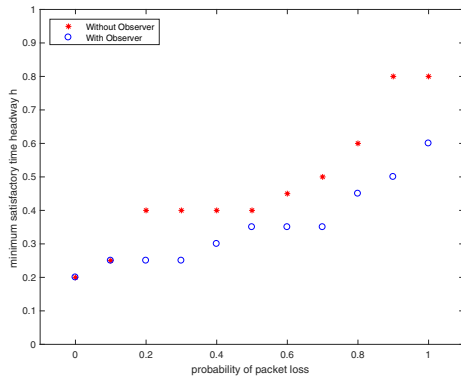
Fig. 6: Simulated minimum stabilising time headway for different values of probability of packet loss.

different from the displayed ones, is reported for each set of 30 simulations and is on average quite small ($E_\% = 0.097$), while it becomes larger for behaviours that are (closer to being) unsatisfactory.

For low probability of packet loss, e.g. $p = 0.1$, the performance of the system is satisfactory, and there is no need to use any observer. As presented in Figure 5a and Figure 5d the platoon has a satisfactory behaviour, for $h = 0.2$. On the contrary, when the loss probability becomes very high, e.g. $p = 0.7$, the performance becomes worse. The behaviour is not satisfactory anymore, and a higher value of h is required to stabilise the system for higher probability of packet loss. For example, when $h = 0.4$, the results when the observer is employed are displayed in Figure 5b. Here the advantage brought by the observer becomes evident: when the observer is not used, the error behaviour is the one displayed in Figure 5e. Lastly, we can check what happens when the communication is never available, i.e. $p = 1$. This is an Adaptive Cruise Control scenario, as it is not cooperative. The vehicles have a stable behaviour for $h = 0.7$, but when the observer is not used, the platoon behaviour is not satisfactory. The displayed results are for a platoon of 5 vehicles, but the error grows in the downstream direction (Figure 5c): for a bigger platoon, the error would grow unbounded. When the observer is employed, the opposite happens: the effect of disturbances decreases along the platoon, as in Figure 5f. A general performance comparison is summarised in Figure 6 where the minimum stabilising value of h is presented when the observer is employed (blue circles) or not (red stars), for the simulated probability range. The minimum stabilising value of h is defined as the value of h such that the condition in (14) is satisfied. From Figure 6 the advantages of employing an observer are clear: it is possible to achieve smaller values for h, which enables vehicles to drive closer, increasing the road throughput while maintaining satisfactory stable behaviour.

V. Conclusion and future work

We showed that a local discrete-time controller, enhanced with a feedforward compensator and an observer, can improve performance for a platoon of vehicles over a lossy wireless channel. Nevertheless, some questions are still open.

For the observer loop in Figure 4 we want to estimate the error transfer function (12) in the best possible way, as the observer design relies on a good estimate of the error $\hat{e}_i(z)$. The stochastic nature of packet loss can not be captured accurately by our frequency domain approach, but only approximated, and this could undermine the accuracy of the input reconstruction. Despite this potential pitfall, our approach of an always-on observer leads to good performance.

Moreover, the design of the observer $O(z)$ could be refined. In this paper, a simple second order system has been employed to test the validity of our approach. The obtained results make apparent the potential improvements allowed by the use of our observer-based architecture.

Finally, this work encourages further reflection over a suitable definition for string stable behaviour which has to be adapted to the stochastic nature of the communication model.

References

[1] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," in *IEEE Vehicular Technology Conference (VTC)*, vol. 42, Dec. 1993, pp. 657 – 672.

[2] A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, vol. 4, Oct. 2003, pp. 143 – 153.

[3] J. Ma, F. Zhou, and M. J. Demetsky, "Evaluating mobility and sustainability benefits of cooperative adaptive cruise control using agent-based modeling approach," in *IEEE Systems and Information Engineering Design Symposium (SIEDS)*, Apr. 2012.

[4] 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.

[5] E. Semsar-Kazerooni, J. Verhaegh, J. Ploeg, and M. Alirezaei, "Cooperative adaptive cruise control: An artificial potential field approach," in *IEEE Intelligent Vehicles Symposium*, Jun. 2016, pp. 361–367.

[6] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, vol. 15, Aug. 2014, pp. 1527–1537.

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

[8] U. Montanaro, G. Fiengo, A. Tufano, and S. Santini, "On the effectiveness of the extended cooperative adaptive control for vehicles platooning," in *European Control Conference (ECC)*, Jun. 2016, pp. 2453–2458.

[9] C. Lei, M. Eenennaam, W. Klein Wolterink, G. Karagiannis, G. Heijenk, and J. Ploeg, "Impact of packet loss on CACC string stability performance," *International Journal of Mechanics and Materials in Design*, pp. 381 – 386, Sep. 2011.

[10] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Towards inter-vehicle communication strategies for platooning support," *International Workshop on Communication Technologies for Vehicles*, pp. 1–6, Oct. 2014.

[11] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of CACC performance subject to unreliable wireless communication," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2013, pp. 1210–1216.

[12] P. Barooah and J. Hespanha, "Error amplification and disturbance propagation in vehicle strings with decentralized linear control," in *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, vol. 2005, Jan. 2006, pp. 4964 – 4969.

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

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