Vehicular Platooning: Multi-Layer Consensus Seeking

Mauro Fusco, Elham Semsar-Kazerooni, Jeroen Ploeg and Nathan van de Wouw

***Abstract*— In this paper, a novel Multi-Layer Consensus Seeking (MLCS) framework is proposed, focusing on the vehicular platooning problem. The vehicles are described by linear heterogeneous dynamics. For example, we consider third-order systems, however the algorithms discussed are suitable for any higher-order. A velocity-dependent inter-vehicle spacing policy is rigorously addressed. The approach used is both *multi-layered* and *consensus-based*. The multi-layer approach allows to separate the problem of estimating the desired trajectories from the problem of controlling the vehicles towards those trajectories while keeping a safety distance. Consensus algorithms will be employed on each layer to solve these two problems.**

I. INTRODUCTION

In the past two decades, network science has experienced a spread in diverse fields [2]. A network is a description of a reality of interest as an interconnection between sub-systems, which are able to interact in order to achieve a certain task. These sub-systems are typically described by dynamical systems which exchange information that influence the global behavior. In [4], a platoon of vehicles is controlled using Vehicle to Vehicle (V2V) communication, which enables the entire platoon to exhibit a stable behavior. This is a motivation to look at Cooperative Adaptive Cruise Control [4] (CACC) as an example of a network control problem. In this paper, we take a more generic approach in the sense that we approach the vehicular platooning problem as a distributed coordination problem. In [7], a distributed coordination approach based on *consensus algorithms* is proposed to solve coordination problems of mobile robots, UAVs (Unmanned Aerial Vehicles) and, more in general, dynamical systems described by single or double integrator dynamics. Here, we present a solution to the platooning problem which takes into account higher-order linear dynamics and a velocity dependent spacing policy (such as in [4]). Furthermore, we will take into account heterogeneity in the dynamical models of the vehicles in the platoon. The challenges coming from these general assumptions are overcome using a *multi-layer* approach. The multi-layer approach is introduced in [3, 5–8]. In particular, in [3], the concept of the coordination variable is briefly explained. The approach presented in [8] is more exclusively focused on the concepts of coordination variable and coordination function. In [5], the implementation scheme of the strategy with the coordination variable is discussed, whereas in [6], the concept of distributed control through coordination variables on each agent is underlined. The work in [7] contains all the previous results and a more general description of the approach in four steps. All these publications deal with the single integrator dynamics, double integrator dynamics, or rigid body attitude dynamics whereas the agents' dynamical properties are identical, also referred to as homogeneous dynamics. In many applications, however it is important to consider heterogeneous dynamics. For example, in the platooning problem, vehicles are typically characterized by different models, e.g. a truck vs. a car, and their dynamics are not adequately described by double integrator dynamics.

M. Fusco, E. Semsar-Kazerooni and J. Ploeg are with the Integrated Vehicle Safety Department, **TNO**, 5700 AT Helmond, The Netherlands. N. v. d. Wouw is with the Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands. Contact author e-mail: `mauro.fusco@tno.nl`.

The main contribution of this paper is the presentation of a multi-layer approach to solve the vehicular platooning problem under higher-order linear heterogeneous dynamics and actual velocity-dependent spacing policy. Furthermore, this paper highlights that MLCS is a useful framework capable of considering more complex scenarios than that addressed with CACC in [4].

This paper is outlined as follows. In Section II, the vehicular platooning problem with velocity-dependent spacing policy, heterogeneous dynamics and safety requirements is stated. In Section III, the Multi-Layer Consensus Seeking (MLCS) framework is discussed. For each layer, we introduce a control objective and propose a consensus algorithm to achieve the objective. Finally, in Section IV, the proposed approach is validated through numerical simulation.

II. VEHICULAR PLATOONING: PROBLEM STATEMENT

We consider the platooning problem as presented, for example, in [4]. We adopt the following longitudinal heterogeneous vehicle dynamics:

$$\dot{x}_i = A_i x_i + B_i u_i, \qquad \text{with } i = 1, 2, \ldots, k, \tag{1}$$

where:

- $A_i \in \mathbb{R}^{n \times n}$ is the *system matrix* of the i-th vehicle, for example: $A_i = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_i} \end{pmatrix}$;
- $B_i \in \mathbb{R}^{n \times 1}$ is the *input matrix* of the i-th vehicle, for example: $B_i = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau_i} \end{pmatrix}$;
- $\tau_i \in \mathbb{R}$ is the time-constant which models the actuator dynamics of the i-th vehicle;
- $i \in \mathbb{N}$ is an index associated with each vehicle;
- $k \in \mathbb{N}$ is the number of the vehicles in the platoon;
- $x_i \in \mathbb{R}^n$ is the state vector of the i-th vehicle;
- $u_i \in \mathbb{R}$ is the control input of the i-th vehicle.

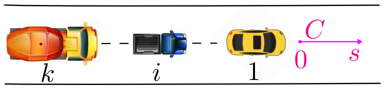

Fig. 1: *Single lane heterogeneous platoon.*

In general, dynamics in (1) can be described by completely different entries for the matrices A_i and B_i, we only require that the state dimension $n \in \mathbb{N}$ remains the same for each vehicle. For the platooning problem, with the particular 3×3 matrix A_i and 3×1 matrix B_i mentioned above, the state vector x_i can be specified as:

$$x_i = \begin{pmatrix} s_i \\ v_i \\ a_i \end{pmatrix}, \tag{2}$$

where:

- $s_i : T \to \mathbb{R}$ is the absolute position of the i-th vehicle in a global reference-frame C (depicted in Fig. 1);
- $v_i : T \to \mathbb{R}$ is the velocity of the i-th vehicle in the global reference-frame C;
- $a_i : T \to \mathbb{R}$ is the acceleration of the i-th vehicle in the global reference-frame C;
- $T \subseteq \mathbb{R}$: is the continuous time set.

A. Control Objective

The control objective for the ensemble of vehicles in (1) consists of driving at a desired speed and acceleration while keeping a velocity-dependent spacing policy between the vehicles. We define desired signals of position, velocity and acceleration for the leader vehicle. Without loss of generality, we assume that the first vehicle of the platoon (see Fig. 1) is the leader vehicle. The desired reference is:

$$x^d(t) = \begin{pmatrix} s^d(t) \\ v^d(t) \\ a^d(t) \end{pmatrix} \tag{3}$$

where:

- $s^d(t) = \int_{t_0}^{t} v^d(\tau)\, d\tau$, with t_0 the initial time instance;
- $v^d(t) = \int_{t_0}^{t} a^d(\tau)\, d\tau$;
- $a^d(t)$ is a reference acceleration signal for the leader vehicle.

Furthermore, each vehicle has to keep a desired distance from the preceding vehicle. Assuming the vehicles' length is equal to zero, the desired position of each vehicle can be written as:

$$s_1^d(t) = s^d(t), \quad s_i^d(t) = s^d(t) - \sum_{l=2}^{i} (r + h v_l(t)) \tag{4}$$

with $i = 2, \ldots, k$, where $r \in \mathbb{R}^+$ is a constant value, which corresponds to the standstill distance between the vehicles. Moreover, $h \in \mathbb{R}$ is the time-gap, which is a key parameter for the platooning problem (see for example [4]). If $h = 0\,\mathrm{s}$, then the spacing policy is *constant*, otherwise it is addressed as *velocity dependent*, reflecting the dependence of the i-th position from the actual local vehicular velocity. Concerning the velocity and acceleration of the i-th vehicle with the velocity-dependent spacing policy in (4), the desired velocities and accelerations are:

$$\begin{aligned} v_1^d(t) &= v^d(t), \quad v_i^d(t) = v^d(t) - h \sum_{l=2}^{i} a_l(t), \\ a_1^d(t) &= a^d(t), \quad a_i^d(t) = a^d(t) - h \sum_{l=2}^{i} \dot{a}_l(t) \end{aligned} \tag{5}$$

with $i = 2, \ldots, k$. The difficulty posed by the velocity-dependent spacing policy is, at least, twofold. First, in the expression of the i-th desired position, velocity and acceleration, see (4) and (5), the summation contains the influence of the actual velocities and accelerations of all preceding vehicles on the i-th vehicle. Second, considering time-varying signals of position, velocity and acceleration in (3), we are implicitly considering the problem of controlling the behavior of the platoon also during the transients and not only at a desired cruise speed.

B. Safety Requirement

Platooning technology intends, among other advantages, to ensure a high level of safety to the road users. Therefore, we have to include safety requirements in the design phase of the control strategy. The most obvious requirement, that we have to consider, is that the vehicles must be able to react to unexpected behavior of one or more neighboring vehicles. To ensure safety, we will require that vehicles interact based on measurements performed by sensing devices such as a radar and/or a camera. This will result in the presence of an *interaction topology* on the lower layer of MLCS. In other words, after estimating the desired trajectories, with MLCS the vehicles will be also capable of reacting to the tracking error of neighbor vehicles. This means that with MLCS, if a vehicle in the platoon is unable to track its desired reference signal, the neighboring vehicles will react and adapt to this unexpected behavior. This reaction is based on measurements performed by on-board sensors and the estimated trajectories of neighboring vehicles. Mathematically, an interaction network between vehicles can be modeled through a *graph*. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set of nodes $\mathcal{V} = \{1, 2, \ldots, k\}$ (i.e., the vehicles) and edges $\mathcal{E}$ (i.e., the sensing/communication of information). An edge can be indicated as a couple $(j, i) \in \mathcal{E}$ when the information flows from j to i. The network topology is described using an *adjacency matrix* $\mathcal{A} = \{a_{ij}\} \in \mathbb{R}^{k \times k}$ where a_{ij} are defined as:

$$a_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } (j, i) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

and a *Laplacian matrix* $\mathcal{L}$ defined as the difference between the diagonal matrix D (which entries are the number of the incoming edges for each node) and the adjacency matrix $\mathcal{A}$. Here, we only consider directed fixed topologies containing a directed spanning tree from the leader (see [7]).

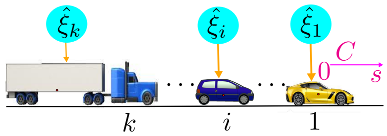

Fig. 2: *Coordination variable estimates $\hat{\xi}_i$ on the higher layer.*

III. Multi-Layer Consensus Seeking

We propose to solve the platooning problem in two steps, which constitute two layers. The first layer is intended to solve the problem of estimating the desired trajectory for each vehicle. This layer will be called the **higher layer**. The second layer is intended to control the vehicles towards the desired trajectories, while respecting the safety requirements. This layer will be called the **lower layer**.

A. Higher Layer

On the higher layer we are facing the problem of estimating the desired position (4), and velocity and acceleration (5) for each vehicle. The approach, proposed to solve this problem, consists of introducing a vector variable ξ, called *coordination variable*, which is defined as:

$$\xi(t) = \begin{pmatrix} s^d(t) \\ v^d(t) \\ a^d(t) \end{pmatrix}. \tag{7}$$

This coordination variable is supposed to be known, or communicated by a road-side unit (RSU), to the leader vehicle. In the remainder of this paper, we assume that $\xi(t)$ is communicated by a RSU to the first vehicle. The purpose of the coordination variable in the multi-layer approach is to enable the vehicles to reconstruct their desired trajectories (4) and (5). Therefore, the coordination variable should include only the information that cannot be *a priori* known by the vehicles. It is possible to extend the content of the coordination variable with extra information if there is a need of tuning other parameters in a coordinated manner. For example, we can imagine a scenario in which it is desirable to tune the standstill distance r or the time-gap h to change the platoon formation in order to realize complex maneuvers. The coordination variable cannot be broadcast to each vehicle, because we are supposing that no global information is available to the individual vehicles. Therefore, we introduce at each vehicle, an estimate of the i-th desired trajectory according to:

$$\hat{\xi}_i = \begin{pmatrix} \hat{s}_i^d \\ \hat{v}_i^d \\ \hat{a}_i^d \end{pmatrix}, \text{ with } i = 1, 2, \ldots, k. \tag{8}$$

In (8), the desired values of position, velocity and acceleration, which are locally computed estimates of (7) at the i-th vehicle (see Fig. 2), are denoted by a hat. In the literature (see [3], [8], [6] and [7]), this is called *decentralization* of the coordination variable ξ.

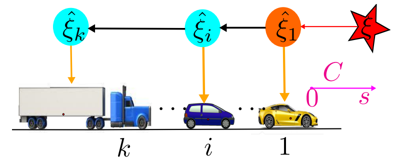

Fig. 3: *Coordination variable estimates $\hat{\xi}_i$ exchange on the higher layer. The red star represents the road side unit which communicates the coordination variable ξ to the leader vehicle (orange node).*

B. Consensus Algorithm on the Higher Layer

Upon introducing a local estimate of the coordination variable in each vehicle (8), we face the problem of ensuring convergence of the estimates towards the desired trajectories in (4) and (5). We can define the following deviations for the estimates (8) with respect to the trajectory communicated by the RSU to the first vehicle in (7):

$$\delta_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \delta_i = -\begin{pmatrix} \sum_{l=2}^{i}(r + hv_l) \\ h\sum_{l=2}^{i} a_l \\ h\sum_{l=2}^{i} \dot{a}_l \end{pmatrix}, \tag{9}$$

with $i = 2, 3, \ldots, k$. Assuming an initial condition for the estimates in (8) different from the values in (7) and/or time-varying references, we want to update the estimates (8) such that:

$$\lim_{t\to\infty} \hat{\xi}_i(t) = \xi(t) + \delta_i(t), \quad \text{with } i = 1, 2, \ldots, k. \tag{10}$$

For the sake of simplicity, each estimate of the coordination variable is updated according to the dynamics:

$$\dot{\hat{\xi}}_i = \nu_i, \quad \text{with } i = 1, 2, \ldots, k, \tag{11}$$

where ν_i is the estimator-law that we have to define. We assume that the platooning system is equipped with a communication mechanism that enables each vehicle to transmit the local estimate of the coordination variable to one or more neighboring vehicles. In addition, we also assume that the leader vehicle receives the true value of the coordination variable (7). In this way, it is possible to assume a *network topology on the higher layer* in which an edge from vehicle j to vehicle i indicates the communication of the coordination variable $\hat{\xi}_j$ to vehicle i (see Fig. 3 for an example of network topology on the higher layer). We introduce the desired separation between the coordination variable estimates as:

$$\Delta_{ij} \triangleq \delta_i - \delta_j, \quad \text{with } i, j = 1, 2, \ldots, k \text{ and } i \neq j. \tag{12}$$

Using the definition of the deviations (9), we obtain:

$$\Delta_{ij} = \delta_i - \delta_j = -\begin{pmatrix} \sum_{l=2}^{i}(r + hv_l) - \sum_{l=2}^{j}(r + hv_l) \\ h(\sum_{l=2}^{i} a_l - \sum_{l=2}^{j} a_l) \\ h(\sum_{l=2}^{i} \dot{a}_l - \sum_{l=2}^{j} \dot{a}_l) \end{pmatrix}, \tag{13}$$

with $i = 2, 3, \ldots, k$ and $j = 1, 2, \ldots, k$. Assuming $j < i$ (i.e. i is one of the followers of vehicle j) we have:

$$\Delta_{ij} = \delta_i - \delta_j = -\begin{pmatrix} \sum_{l=j+1}^{i}(r + hv_l) \\ h\sum_{l=j+1}^{i} a_l \\ h\sum_{l=j+1}^{i} \dot{a}_l \end{pmatrix}, \tag{14}$$

with $i = 2, 3, \dots, k$. Therefore, in (14) we have that Δ_{ij} depends only on the velocities, accelerations and jerks of the vehicles between vehicle i and vehicle $j+1$ (included). From (14), it follows that the desired separations between the coordination variable estimates of two consecutive vehicles are:

$$\Delta_{i(i-1)} = \delta_i - \delta_{i-1} = -\begin{pmatrix} r + hv_i \\ ha_i \\ h\dot{a}_i \end{pmatrix} \tag{15}$$

with $i = 2, 3, \dots, k$. Assuming a one-vehicle look-ahead interaction topology on the higher layer (see Fig. 3), we propose the following consensus algorithm with relative deviations:

$$\begin{aligned} \nu_1 &= \dot{\xi} + \gamma(\xi - \hat{\xi}_1) \\ \nu_i &= \nu_{i-1} + \dot{\Delta}_{i(i-1)} + \gamma(\hat{\xi}_{i-1} - \hat{\xi}_i + \Delta_{i(i-1)}), \end{aligned} \tag{16}$$

with $i = 2, 3, \dots, k$. In (16), $\gamma \in \mathbb{R}^+$ is a positive gain which determines the rate of convergence of the estimates $\hat{\xi}_i$ towards $\xi + \delta_i$, while $\nu_{i-1} + \dot{\Delta}_{i(i-1)}$ is a feed-forward estimation term which helps the tracking of time-varying trajectories.

C. Lower Layer

Assuming that each vehicle estimates its desired trajectory using (11) and (16), then a control strategy could consist of designing a local control action for each vehicle based on the coordination variable $\hat{\xi}_i$, as depicted in Fig. 3. In this case, the lower layer would be designed taking into account only the (estimated) reference signals, and no other exchange of information between the vehicles is considered. In other words, we could define the control objective on the lower layer as:

$$\lim_{t\to\infty} x_i(t) = \hat{\xi}_i(t), \quad \forall i \in \{1, 2, \dots, k\}. \tag{17}$$

However, this approach will not guarantee a safe platooning. In fact, in case of an unexpected behavior of one or more vehicles, the neighboring vehicles cannot react. This may happen because the higher layer is designed taking into account only the estimated reference signals without other exchange of information between the vehicles. For example, if the desired acceleration is positive but the first vehicle stops accelerating due to an unexpected failure, the following vehicles will continue to track their estimated reference trajectory. As a consequence, an accident will occur. Therefore, we assume that on the lower layer, another *interaction topology* exists (for example, see Fig. 4). Similar to the higher layer, this interaction can be mathematically described through a *network topology on the lower layer*, where an edge from the vehicle j to the vehicle i indicates the sensing/communication of information from vehicle j to vehicle i. We assume that each vehicle is equipped with a sensing/communication mechanism that enables each vehicle to measure and/or to transmit:

- the relative position from a neighbour vehicle $s_i - s_j$ (for example, measured with a radar and/or camera);
- the relative velocity from a neighbour vehicle $v_i - v_j$ (for example, measured with a radar and/or camera);

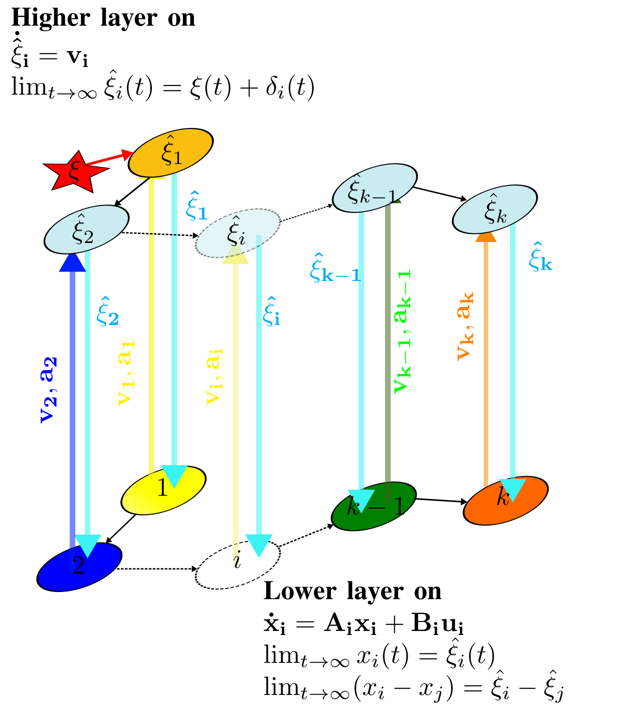

Fig. 4: *MLCS conceptual scheme. In this scheme, the lower layer represents heterogeneous interacting vehicles (nodes with different colors). Furthermore, it is worthwhile noticing that in general the interaction topologies can be different on the two layers.*

- the relative acceleration from a neighbour vehicle $a_i - a_j$ (for example, a_i is measured with a local accelerometer, while a_j is measured and transmitted through V2V communication).

On the basis of the observations above, we consider the following objective, instead of the control objective in (17):

$$\lim_{t\to\infty} (x_i - x_j) = \hat{\xi}_i - \hat{\xi}_j, \quad j \in \mathcal{N}_i, \ \forall i \in \{1, 2, \dots, k\}. \tag{18}$$

where $\mathcal{N}_i \subset \mathcal{V}$ is the set of neighbors of vehicle i. The control objective (17) is defined to ensure the tracking of the desired trajectory, therefore the control algorithm on the lower layer will still contain a feed-forward part to achieve (17). Instead, the control objective (18) is intended to ensure the correct relative distances, velocities and accelerations between the vehicles in the platoon independently from the control objective (17). We also specify that we aim to achieve (18) by introducing interaction between vehicles supported by sensing/communication devices. The control objective (18) can be also interpreted as follows:

$$\lim_{t\to\infty} (\hat{\xi}_i - x_i) = \lim_{t\to\infty} (\hat{\xi}_j - x_j), \quad j \in \mathcal{N}_i, \ \forall i \in \{1, 2, \dots, k\}. \tag{19}$$

i.e. we desire that the tracking error $e_i \triangleq \hat{\xi}_i - x_i$ converges to the same value as $e_j \triangleq \hat{\xi}_j - x_j$. This would ensure a safe behavior of the platoon at least asymptotically, because if e_j is not zero, we also desire that the vehicle i does not follow its desired trajectories (4) and (5), that are designed without taking into account failures on the vehicles.

D. Consensus Algorithm on the Lower Layer

Finally, it is possible to define a *consensus algorithm* that achieves the control objectives (17) and (18) as:

$$u_i = u_i^r + cL_i \sum_{j=1}^{k} a_{ij}(x_i - x_j + \hat{\xi}_j - \hat{\xi}_i), \qquad (20)$$

with $i = 1, 2, \ldots, k$, where $c \in \mathbb{R}^+$ denotes the **coupling strength** and $L_i \in \mathbb{R}^{1\times n}$ with $i = 1, 2, \ldots, k$ are **feedback gain vectors**. The protocol (20) contains:

1) a *feed-forward action* u_i^r computed in real time on the basis of the i-th estimated desired trajectory $\hat{\xi}_i$ and designed on the basis of the i-th linear model (1). This action is desired to fulfill the control objective (17);
2) a *feedback network-based action* $cL_i \sum_{j=1}^{k} a_{ij}(x_i - x_j + \hat{\xi}_j - \hat{\xi}_i)$ based on the interaction between the vehicles on the lower layer. This action is designed to fulfill the control objective (18).

There is no particular constraint for the choice of the technique used to compute u_i^r. However, assuming heterogeneous vehicles, u_i^r has to take into account the difference in vehicles' dynamics, and it is not possible to use the desired acceleration of the preceding vehicle as in [4]. Therefore, u_i^r must be computed with a control-tracking algorithm and based on the model (1). An example is given in Section IV. Regarding the choice of the parameters in (20), see [10] in which it is proposed to compute the i-th matrix L_i by solving the Linear Matrix Inequality:

$$A_i P_i + P_i A_i^T - 2B_i B_i^T < 0 \qquad (21)$$

to obtain a solution $P_i > 0$, and computing:

$$L_i = -B_i^T P_i^{-1} \qquad (22)$$

and also to choose c on the basis of the interaction topology on the lower layer as follows:

$$c \geq \frac{1}{\underset{i=2,3,\ldots,k}{min} \{Re(\lambda_i)\}} \qquad (23)$$

where $Re(\lambda_i)$, with $i = 2, 3, \ldots, k$ denote the real part of the non-zero eigenvalues of the Laplacian matrix, defined in relation to the topology on the lower layer.

IV. SIMULATION RESULTS

In this section, numerical simulations are shown to validate the stability of the whole closed-loop system, see Fig. 4. The simulations are performed with Matlab/Simulink. For the sake of simplicity, we consider a platoon of three vehicles of which the dynamics are characterized by the time constants in Table I, where the initial conditions are also listed. We choose one-vehicle look-ahead interaction topologies both on the higher and lower layer. The required standstill distance and the time-gap are set to $r = 5\,\mathrm{m}$ and $h = 0.6\,\mathrm{s}$, respectively. The control gain on the higher layer is set to $\gamma = 2$. On the lower layer, the gain vectors L_i with $i = 1, 2, 3$ are computed solving the LMIs in (21), using the SeDuMi toolbox [9]. The resulting gain vectors are given in Table II. The coupling strength on the lower layer is set to $c = 2$, given the fact that with a one-vehicle look-ahead interaction topology, all the non-zero eigenvalues of the Laplacian matrix are equal to 1.

TABLE I: Time constants and initial conditions.

Vehicle	τ_i $[s]$	$s_i(0)$ $[m]$	$v_i(0)$ $[m/s]$	$a_i(0)$ $[m/s^2]$
1	0.1	0	0	0
2	0.55	-10	0	0
3	0.08	-15	0	0

TABLE II: *Gain vectors on the lower layer.*

Vehicle	L_i
1	$(-0.0654\ -0.7236\ -1.7194)$
2	$(-0.4403\ -1.4519\ -1.8921)$
3	$(-0.0379\ -0.4561\ -1.2369)$

A. u_i^r design example

In this simulation, we propose to design the feed-forward action u_i^r by solving an optimal linear quadratic (LQ) tracking problem. Other choices are also possible. We define k reference models:

$$\dot{x}_i^r = A_i x_i^r + B_i u_i^r, \text{ with } i = 1, 2, \ldots, k, \qquad (24)$$

where the matrices in (24) are defined in (1). As a result, we design u_i^r such that $\hat{\xi}_i$ is a (globally asymptotically stable) solution of the i-th system in (24), $\forall i \in \{1, 2, \ldots, k\}$. Mathematically, we search for the solution of the following *linear quadratic optimization problem*:

$$\begin{aligned} &\min_{u_i^r(\cdot)} \int_0^{+\infty} \{(x_i^r - \hat{\xi}_i)^T Q_i (x_i^r - \hat{\xi}_i) + u_i^{rT} R_i u_i^r\}\, dt \\ &\text{subject to} \\ &\dot{x}_i^r = A_i x_i^r + B_i u_i^r, \end{aligned} \qquad (25)$$

where:

- $Q_i \geq 0$ is the state-weighting matrix;
- $R_i > 0$ is the input-weighting matrix.

The problem (25) can be solved by making the following variable substitution:

$$\zeta_i = x_i^r - \hat{\xi}_i \qquad (26)$$

As a consequence, (25) is transformed into the following *Optimal Linear Quadratic Regulation* problem (LQR-problem), with disturbance $w_i = A_i\hat{\xi}_i - \dot{\hat{\xi}}_i$:

$$\begin{aligned} &\min_{u_i^r(\cdot)} \int_0^{+\infty} \{\zeta_i^T Q_i \zeta_i + u_i^{rT} R_i u_i^r\}\, dt \\ &\text{subject to} \\ &\dot{\zeta}_i = A_i \zeta_i + B_i u_i^r + w_i \end{aligned} \qquad (27)$$

In [1], the solution of (27) under the "disturbance" w_i is given by:

$$u_i^r = K_{fb,i}(\hat{\xi}_i - x_i^r) - R_i^{-1} B_i K_{fw,i} w_i \qquad (28)$$

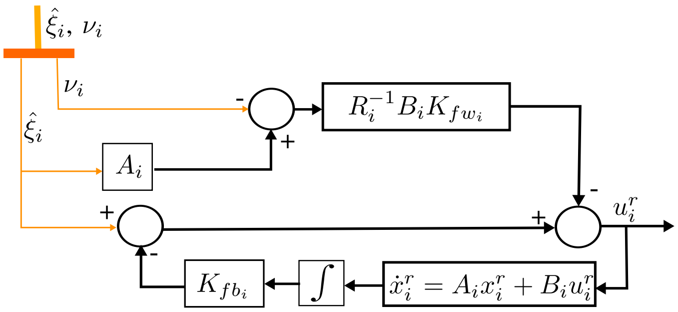

Fig. 5: *Feed-forward design with a LQ tracking approach.*

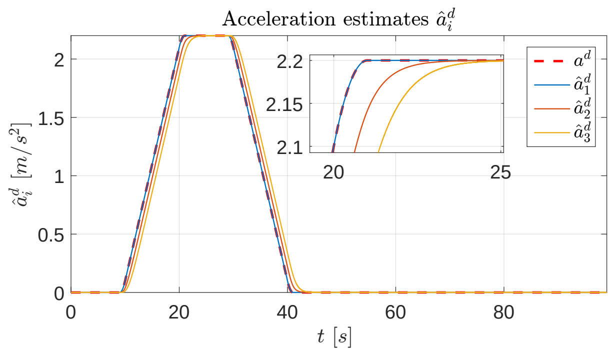

Fig. 6: *Acceleration estimates* $\hat{a}_i^d(t)$*, with* $i = 1, 2, 3$.

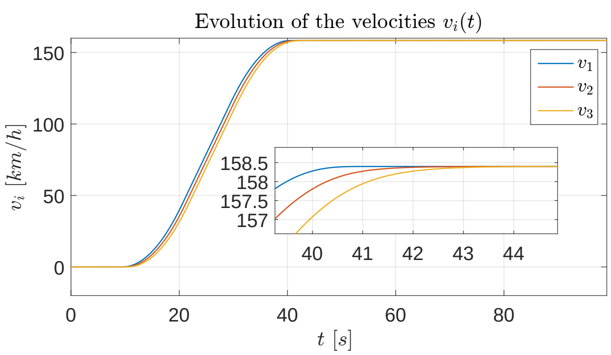

Fig. 7: *Velocities* v_i*, with* $i = 1, 2, 3$.

where the matrices $K_{fb,i}$ and $K_{fw,i}$ can be computed by solving some Algebraic Riccati Equations as discussed in [1]. The choice of the LQ weighting matrices are: $Q_i = diag\{1, 100, 100\}$ and $R_i = 120$, with $i = 1, 2, 3$. Also in this case, the LMIs for the computation of u_i^r, [1], are solved using the SeDuMi toolbox [9]. A trapezoidal desired acceleration is communicated by the RSU to the first vehicle in the platoon. With the consensus algorithm (16), the acceleration estimates $\hat{a}_i^d(t)$ follow the trajectories in Fig. 6. Absolute velocities are depicted in Fig. 7. The errors in the relative positions are shown in Fig. 8.

V. CONCLUSIONS

In this paper, we have proposed a Multi-Layer Consensus Seeking (MLCS) approach to solve the vehicular platooning problem. The dynamics considered are linear and heterogeneous, furthermore a velocity-dependent spacing policy

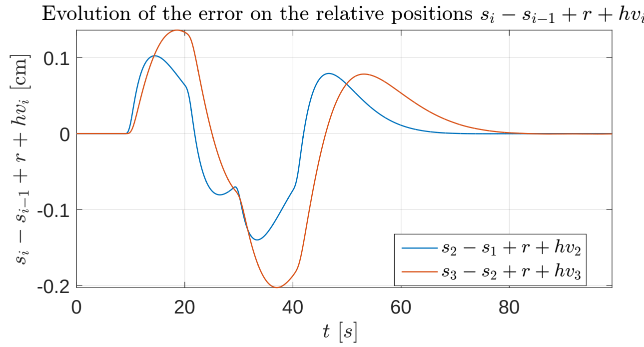

Fig. 8: *Errors in the relative positions.*

is adopted. MLCS allows to tackle the complexity coming from the above assumptions within a generic framework. We envision that this framework can be used to step in a higher level of automation when considering the realization of complex maneuvers. This is the subject of ongoing research.

REFERENCES

[1] Anderson B. D. O. and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Inc., 1979.

[2] Barabasi. *Taming complexity*. Vol. 1. Nature, 2005.

[3] T. W. McLain and R. W. Beard. *Coordination Variables, Coordination Functions, and Cooperative Timing Missions*. Vol. 28. Journal of Guidance, Control, and Dynamics. 2005, pp. 150–161.

[4] J. Ploeg, T. M. B. Scheepers, E. v. Nunen, N. v. d. Wouw, and H. Nijmeijer. *Design and Experimental Evaluation of Cooperative Adaptive Cruise Control*. 14th International IEEE Conference on Intelligent Transportation Systems. 2011, pp. 160–165.

[5] W. Ren. *Cooperative Control Design Strategies with Local Interactions*. IEEE International Conference on Networking, Sensing and Control. 2006, pp. 451–456.

[6] W. Ren. *Decentralization of Coordination Variables in Multi-vehicle Systems*. IEEE International Conference on Networking, Sensing and Control. 2006, pp. 550–555.

[7] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control - Theory and Applications*. Springer Verlag, 2007.

[8] W. Ren, T. W. McLain, and R. W. Beard. *Coordination Variables and Consensus Building in Multiple Vehicle Systems*. Vol. 309. Springer-Verlag Series: Lecture Notes in Control and Information Sciences, 2005, pp. 171–188.

[9] J.F. Sturm. *A MATLAB toolbox for optimization over symmetric cones*. Vol. 11. Optimization Methods and Software 1. 1999.

[10] L. Zhongkui, D. Zhisheng, and C. Guanrong. *Consensus of Multiagent Systems and Synchronization of Complex Networks: A Unified Viewpoint*. Vol. 57. IEEE Transactions on Circuits and Systems 1. 2010, pp. 213–224.