Cloud-Supported Formation Control of Second-Order Multi-Agent Systems

Antonio Adaldo, Davide Liuzza, Dimos V. Dimarogonas, Member, IEEE and Karl H. Johansson, Fellow, IEEE

Abstract—This paper addresses a formation problem for a network of autonomous agents with second-order dynamics and bounded disturbances. Coordination is achieved by having the agents asynchronously upload (download) data to (from) a shared repository, rather than directly exchanging data with other agents. Well-posedness of the closed-loop system is demonstrated by showing that there exists a lower bound for the time interval between two consecutive agent accesses to the repository. Numerical simulations corroborate the theoretical results.

I. INTRODUCTION

Coordination of networked multi-agent systems is the subject of a large body of research work, because such systems constitute a suitable model for a large number of phenomena in robotics, biology, physics, and social sciences [1]–[3].

In most realistic scenarios, the agents in a multi-agent system have limited communication capabilities. This happens, for example, when they communicate over a wireless medium, which is a shared resource with limited throughput capacity. In some cases, inter-agent communication is completely or almost completely interdicted. This challenge arises, for example, in the coordination of a fleet of autonomous underwater vehicles (AUVs) [4]. Because of their severely limited communication, sensing, and localization capabilities, underwater vehicles are virtually isolated systems. Underwater communication and positioning may be implemented by means of battery-powered acoustic modems, but such devices are expensive, limited in range, and power-hungry. Inertial sensors for underwater positioning are prohibitively expensive in most practical scenarios. Moreover, GPS is not available underwater, and a vehicle needs to surface whenever it needs to get a position fix [5].

When such limitations arise, coordination strategies that rely on continuous information exchanges among the agents cannot be implemented. To address this challenge, the idea of triggered control [6], [7] has been tailored to multi-agent systems. Triggered control was introduced to limit the amount of communication within the parts of a feedback control system (plant, sensors, actuators). In the context of multi-agent systems, triggered control is used to limit the communication among different agents. Various flavors of triggered control have been applied to multi-agent systems: namely, with event-triggered control, inter-agent communication is triggered when a given state condition is satisfied [8]; with self-triggered control, the agents schedule when to exchange data in a recursive fashion, so that there is no need to monitor a condition between consecutive communication instances [9]. However, even these triggered control schemes require that the agents exchange information, and, therefore, are not well-suited for those scenarios where direct inter-agent communication is interdicted. The use of a shared information repository in multi-agent control is subject to recent, but growing, research attention. In [10], the authors employ asynchronous communication with a base station to address a multi-agent coverage control problem. In [11], the authors present a cloud-supported approach to multi-agent optimization.

In this paper, we present a multi-agent control scheme where inter-agent communication is completely replaced by the use of a shared information repository hosted on a cloud. Differently than in traditional event-triggered coordination schemes, here each agent schedules its own cloud accesses independently, and does not need to be alert for information broadcast by other agents. When an agent accesses the repository, it uploads some data packets, and downloads other packets that were previously deposited by other agents. Therefore, each agent receives only outdated information about the state of the other agents. The control law and the rule for scheduling the cloud accesses are designed to guarantee that the closed-loop system is well-posed and achieves the control objective, in spite of only using outdated information. Our analysis extends the use of the edge Laplacian [12], [13] to second-order directed networks, which allows us to consider control tasks with asymmetric information flow among the agents, such as leader-following tasks. With respect to the related works [14]–[17], this paper introduces cloud support for multi-agent systems with second-order dynamics. Moreover, differently than [16], [17], here we consider additive disturbances (both persistent and vanishing) on the agent dynamics.

With respect to centralized solutions for multi-agent coordination, the proposed cloud-supported control scheme presents several important advantages: the computational burden can be distributed between the agents and the cloud according to the available resources; the architecture can be made resilient to failures of individual subsystems; fall-back local control laws can be used to put the agents in a failsafe state in case the communication with the cloud is temporarily lost; the framework can be also used for tasks that require the...
agents to perform local computations between two consecutive cloud accesses. We wish to emphasize that the proposed cloud-supported control scheme is, conversely to what happens to a centralized one, scalable with the number of agents. Indeed, each agent can carry its own computational resources, while performing only local computations. The amount of such computation does not scale with respect to the number of agents added to the overall system. Indeed, at any cloud access, only the data referred to a single agent is communicated and processed. The only centralized resource that grows with respect to the number of agents is the memory of the cloud, which scales linearly. Moreover, the proposed setup differs from existing control schemes for asynchronous consensus algorithms with communication delays, e.g. [18], in that the delay in the information acquisition is not an undesired exogenous phenomenon, but it is induced by the control policy itself. In particular, the proposed scheduling policy aims at prolonging as much as possible the interval between two consecutive cloud connections of the same agent, in order to reduce the total number of communication instances.

Our motivating application is a waypoint generation algorithm for formation control of AUVs, which, as described above, represents a challenging application, since underwater communication is interdicted: the traditional event-triggered communication schemes are not applicable, since the AUVs are isolated while navigating underwater, and even if one vehicle emerges to broadcast a message, the other vehicles would be unable to receive it. Instead, with the proposed cloud-supported scheme, each vehicle can access the cloud repository while on the water surface, thus being able to download data previously uploaded by different agents.

The rest of this paper is organized as follows. In Section II, we present some background notation and results. In Sections III and IV, we present the system model and outline the control strategy. In Section V, we state our main result, whose proof is given in Sections VI, VII and VIII. Section IX corroborates the theoretical results by presenting two numerical simulations of the proposed control strategy. Finally, in Section X, we present our conclusions and some directions for future research.

II. Preliminaries

The set of the positive real numbers is denoted as $\mathbb{R}_{++}$. The operator $\| \cdot \|$ denotes the Euclidean norm of a vector and the corresponding induced norm of a matrix. The operator $\otimes$ denotes the Kronecker product. For the properties of the Kronecker product, we refer the reader to [19]. The $n$-by-$n$ identity matrix is denoted as $I_n$, while the $n$-by-$m$ matrix whose entries are all zero is denoted as $0_{n \times m}$. Similarly, the column vector with $n$ zero entries is denoted as $0_n$. For a matrix $M \in \mathbb{R}^{m \times n}$, the entry in the $i$th row and $j$th column is denoted as $[M]_{ij}$, while $\text{eig}(M)$ is the set of the distinct eigenvalues of $M$.

In this paper, a graph is defined as a triple $G = (V, \mathcal{E}, w)$, where $V \{1, \ldots, N\}$ with $N \in \mathbb{N}$, $\mathcal{E} \subseteq V \times V$ with the constraint $(i, i) \notin \mathcal{E}$ for all $i \in V$, and $w : \mathcal{E} \rightarrow \mathbb{R}_{++}$.

Each element of $V$ is called a vertex, and each element of $\mathcal{E}$ is called an edge. For each edge $(j, i)$, the value $w(j, i)$ is called the weight of that edge. This type of graph is known in the literature as a simple weighted digraph [20]. The edges are denoted as $e_1, \ldots, e_M$, where $M$ is the number of edges in the graph. For each edge $e_i$, we denote as $\text{head}(e_i)$ and $\text{tail}(e_i)$ respectively the first and the second node of the edge. A graph is illustrated by representing each vertex as a circle and each edge $e$ as an arrow from $\text{tail}(e)$ to $\text{head}(e)$. For example, Figure 1 illustrates a graph with 4 vertices and 5 edges, each labeled with its index (the weights of the edges are not represented). For each vertex $i \in V$, the set $N_i = \{ j \in V : (i, j) \in \mathcal{E} \}$ is called the neighborhood of $i$, and the vertexes $j \in N_i$ are called the neighbors of $i$.

Moreover, the sets $\mathcal{E}_{i}^{\text{in}} = \{ e \in \mathcal{E} : \text{head}(e) = i \}$, $\mathcal{E}_{i}^{\text{out}} = \{ e \in \mathcal{E} : \text{tail}(e) = i \}$ are called respectively the edge in-neighborhood and edge out-neighborhood of vertex $i$. A path from a vertex $i_1$ to a vertex $i_p$ is a sequence of distinct vertexes $i_1, \ldots, i_p$ such that $(i_k, i_{k+1}) \in \mathcal{E}$ for each $k = 1, \ldots, P - 1$. The incidence matrix is defined as $B \in \mathbb{R}^{N \times M}$ such that $[B]_{i, e} = 1$ if $e_i \in \mathcal{E}_{i}^{\text{in}}$, $[B]_{i, e} = -1$ if $e_i \in \mathcal{E}_{i}^{\text{out}}$, and $[B]_{i, e} = 0$ otherwise. We also introduce the matrix $C \in \mathbb{R}^{N \times M}$ such that $[C]_{i, e} = w(e_i)$ if $e_i \in \mathcal{E}_{i}^{\text{in}}$ and $[C]_{i, e} = 0$ otherwise. The Laplacian matrix is defined as $L = CB^\top$. A spanning tree is a subset $T \subseteq \mathcal{E}$ of the edges with the following properties: (i) there exists a vertex $i_0$ such that there exists a path from $i_0$ to any other vertex in the graph made up of edges in $T$; (ii) the property (i) does not hold for any proper subset of $T$. The vertex $i_0$ is called the root of the spanning tree $T$. If a spanning tree exists, then it contains exactly $N - 1$ edges. For a graph containing a spanning tree, we take without loss of generality $T = \{ e_1, \ldots, e_{N-1} \}$, and, following [13], we define $B_T$ as the full column-rank minor of $B$ made up of the first $N - 1$ columns.

III. System Model

A. Agent Model

We consider a network of $N$ autonomous agents indexed as $1, \ldots, N$, and we let $V = \{1, \ldots, N\}$. Each agent $i$ has a position $p_i(t) \in \mathbb{R}^n$ and a velocity $v_i(t) \in \mathbb{R}^n$, which evolve according to

\begin{align}
\dot{p}_i(t) &= v_i(t), \\
\dot{v}_i(t) &= u_i(t) + d_i(t),
\end{align}

where $u_i(t) \in \mathbb{R}^n$ is a control input, and $d_i(t) \in \mathbb{R}^n$ is a disturbance input.

\[\text{Fig. 1. A graph with } 4 \text{ nodes and } 5 \text{ edges. The nodes and the edges are labeled with their indexes.}\]
Assumption III.1. The disturbance signals $d_i(t)$ in (1) satisfy $\|d_i(t)\| \leq \delta(t)$, where
\[
\delta(t) = (\delta_0 - \delta_\infty)e^{-\lambda_dt} + \delta_\infty,
\]
for some $0 \leq \delta_\infty \leq \delta_0$ and $\lambda_d > 0$.

Assumption III.1 allows to consider scenarios where only a constant upper bound is known ($\delta_0 = \delta_\infty$) as well as scenarios where the disturbances vanish exponentially ($\delta_\infty = 0$).

B. Control Objective

The control objective is to bring the agents to a formation defined by the bias vectors $b_1, \ldots, b_N \in \mathbb{R}^n$, in the sense that, for all $i \in \mathcal{V}$, we have $p_i(t) \rightarrow \bar{p}(t) + b_i$, where $\bar{p}$ is the average position, and $v_i(t) \rightarrow \bar{v}(t)$, where $\bar{v}(t)$ is the average velocity. This objective can be cast as the practical second-order consensus over a given graph of the unbiased positions $p_i(t) - b_i$ and of the velocities $v_i(t)$. To formalize this control objective mathematically, let $\mathcal{G}$ be a graph containing a spanning tree $T$, and let $B_T$ be the incidence matrix associated to the edge in the tree. Define the edge states of the network as $x(t) = (B_T \otimes I_n)(p(t) - b)$ and $y(t) = (B_T \otimes I_n)v(t)$, where we have denoted $p(t) = [p_1(t)^\top, \ldots, p_N(t)^\top]^\top$, and similarly for $b$ and $v(t)$. Finally, let $\xi(t) = [x(t)^\top, y(t)^\top]^\top$. We say that the multi-agent system (1) achieves practical consensus over $\mathcal{G}$ if there exists $\chi \geq 0$ such that
\[
\limsup_{t \to \infty} \|\xi(t)\| \leq \chi.
\]
In particular, if the system achieves practical consensus with $\chi = 0$, we say that the system achieves asymptotic consensus. In the rest of the paper, we take $b_i = 0$, for all $i \in \mathcal{V}$ to avoid clutter in the notation. The results extend trivially to the case of nonzero bias vectors.

C. Cloud Repository

The agents cannot exchange any information directly, but can only upload and download information on a shared repository hosted on a cloud, which is accessed intermittently by each agent and asynchronously by different agents. The topology of the information exchanges happening through the cloud is described by a network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$: each vertex represents one of the agents, and each agent $i$ downloads the information uploaded by its neighbors $j \in \mathcal{N}_i$ in the graph.

Assumption III.2. The network graph $\mathcal{G}$ is time-invariant and contains a spanning tree.

When an agent accesses the cloud, it also has access to a sampled measurement of its own state. The time instants when agent $i$ accesses the cloud are denoted as $t_{i,k}$, with $k \in \mathbb{N}$, and by convention $t_{i,0} = 0$ for all the agents. For convenience, we denote as $l_i(t)$ the index of the most recent access time of agent $i$ before time $t$, i.e.,
\[
l_i(t) = \max\{k \in \mathbb{N} : t_{i,k} \leq t\}.
\]
The measurement obtained by agent $i$ upon the time instant $t_{i,k}$ is denoted as $x_{i,k}$. The control signals $u_i(t)$ are held constant between two consecutive cloud accesses:
\[
u_i(t) = u_{i,k} \quad \forall t \in [t_{i,k}, t_{i,k+1}).
\]
The data contained in the cloud at a generic time instant $t \geq 0$. The $i$-th column corresponds to the latest packet uploaded by agent $i$. The time dependence of the functions $l_i$ is omitted to keep the notation agile.

Table III-C. Data contained in the cloud at a generic time instant $t \geq 0$. The $i$-th column corresponds to the latest packet uploaded by agent $i$. The time dependence of the functions $l_i$ is omitted to keep the notation agile.

<table>
<thead>
<tr>
<th>agent</th>
<th>1</th>
<th>2</th>
<th>\ldots</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>last access</td>
<td>$t_{1,k}$</td>
<td>$t_{2,k}$</td>
<td>\ldots</td>
<td>$t_{N,k}$</td>
</tr>
<tr>
<td>position</td>
<td>$p_{1,k}$</td>
<td>$p_{2,k}$</td>
<td>\ldots</td>
<td>$p_{N,k}$</td>
</tr>
<tr>
<td>velocity</td>
<td>$v_{1,k}$</td>
<td>$v_{2,k}$</td>
<td>\ldots</td>
<td>$v_{N,k}$</td>
</tr>
<tr>
<td>control</td>
<td>$u_{1,k}$</td>
<td>$u_{2,k}$</td>
<td>\ldots</td>
<td>$u_{N,k}$</td>
</tr>
<tr>
<td>next access</td>
<td>$t_{1,k+1}$</td>
<td>$t_{2,k+1}$</td>
<td>\ldots</td>
<td>$t_{N,k+1}$</td>
</tr>
</tbody>
</table>

Remark III.1. In most existing self-triggered control protocols for multi-agent coordination, when one agent updates its control input, such information is broadcast immediately to that agent’s neighbors, which requires the neighbors to always be alert for possibly incoming information. Conversely, in the
Fig. 2. Excerpt of a possible sequence of cloud accesses on the time line. Recall that \( t_j, j(t_i) \) denotes the most recent cloud access of agent \( j \) with respect to the time \( t \). Note that there can be more than one access of agent \( j \) between two consecutive accesses of agent \( i \).

**Algorithm 1** Operations executed by agent \( i \) at time \( t_{i,k} \).

- download measurements \( p_{i,k} \) and \( v_{i,k} \)
  - for \( j \in N_i \) do
    - download packet \( \{ t_{j,l_j(i_{j,k})}, p_{i,l_j(i_{j,k})}, v_{i,l_j(i_{j,k})}, u_{j,l_j(i_{j,k}), t_{j,l_j(i_{j,k})+1}} \} \)
  - end for
- compute control input \( u_{i,k} \)
- schedule next access \( t_{i,k+1} \)
- upload packet \( \{ t_{i,k}, p_{i,k}, v_{i,k}, u_{i,k}, t_{i,k+1} \} \)

The proposed cloud-based framework the agents do not need to be alert for incoming information, because they only acquire new information on the scheduled cloud accesses.

**D. Controller**

The control inputs \( u_{i,k} \) are computed through a second-order Laplacian flow of predicted states, where the predictions are based on the sample measurements acquired from the cloud, and on assuming that no disturbances are acting on the agents. Namely, we let \( \hat{v}_{i,k}(t) \) (respectively, \( \hat{p}_{i,k}(t) \)) be the velocity (respectively, position) of agent \( j \) at time \( t \) as predicted by agent \( i \) upon its \( k \)-th access to the cloud. These predictions are computed by agent \( i \) using the data downloaded from the cloud about agent \( j \) as follows:

\[
\begin{align*}
\hat{v}_{i,j}(t) &= v_{j,l_j(i_{j,k})} + (t - t_{j,l_j(i_{j,k})})u_{j,l_j(i_{j,k})}, \\
\hat{p}_{i,j}(t) &= p_{j,l_j(i_{j,k})} + \int_{t_{j,l_j(i_{j,k})}}^{t} \hat{v}_{i,j}(\tau) d\tau.
\end{align*}
\]

Note that the predictions \( \hat{v}_{i,j}(t) \) are obtained by integrating the agent dynamics \( (1) \) in the time interval \( [t_{j,l_j(i_{j,k})}, t] \), while neglecting the effect of the disturbances. Finally, the control input \( u_{i,k} \) is computed as

\[
\begin{align*}
u_{i,k} = \sum_{j \in N_i} w_{ij}(k_p(p_{j,k}(t_{i,k}) - p_{i,k}) + k_v(\hat{v}_{i,j}(t_{i,k}) - v_{i,k})),
\end{align*}
\]

where \( k_p \) and \( k_v \) are positive gains, \( N_i \) is the set of the neighbors of agent \( i \) in the network graph, and \( w_{ij} = w(j, i) > 0 \) is the weight of edge \((j, i)\).

**E. Dynamics of the Edge States**

Since the controller is based on a Laplacian flow, it is convenient to rewrite the system dynamics in terms of edge states \( p_{j}(t) - p_{i}(t) \) and \( v_{j}(t) - v_{i}(t) \), where \((j, i)\) is an edge of the spanning tree in the network graph. Namely, define the edge states \( z(t) = (x(t), y(t)) \) as in Section III-B, with \( B_T \) referred to the network graph. Using (1), the dynamics of the edge states are described by

\[
\begin{align*}
\dot{x}(t) &= y(t), \\
\dot{y}(t) &= (B_T^2 \otimes I_n)(u(t) + d(t)),
\end{align*}
\]

where we have denoted \( u(t) = [u_1(t)^T, \ldots, u_N(t)^T]^T \) and similarly for \( d(t) \).

**IV. SELF-TRIGGERED CLOUD ACCESS SCHEDULING**

Each agent schedules its own access to the cloud recursively; that is, agent \( i \) schedules the access \( t_{i,k+1} \) when it accesses the cloud at time \( t_{i,k} \). The scheduling is based on computing an upper bound \( \sigma_i(t) \) on the difference \( \hat{u}_i(t) \) between the actual control signal \( u_i \) and an ideal diffusive coupling control, defined as

\[
\sigma_i(t) = \sum_{j \in N_i} w_{ij}(k_p(p_{j}(t) - p_{i}(t)) + k_v(v_{j}(t) - v_{i}(t))).
\]

When the error bound becomes larger than a given threshold function (given later in (10)), a cloud access is triggered, so that the control error is reset to a lower value thanks to the new data acquired from the cloud. Following [21], the threshold function is chosen as

\[
\varsigma(t) = \varsigma_0 + (\varsigma_0 - \varsigma_\infty)e^{-\lambda_c t},
\]

with \( \lambda_c > 0 \) and \( 0 \leq \varsigma_0 < \varsigma_\infty \).

**Remark IV.1.** Note that the parameters \( \varsigma_0, \varsigma_\infty \) and \( \lambda_c \) represent a tradeoff between convergence performance and the number of control updates: smaller values of \( \varsigma_\infty \) lead to a smaller convergence radius, but possibly induce a larger number of control updates; smaller values of \( \varsigma_0 \) and larger values of \( \lambda_c \) lead to faster convergence rate, but possibly induce a larger number of control updates.

Next, we complete the definition of our control algorithm by giving the expression for \( \sigma_i(t) \) and the scheduling law. Note that \( \sigma_i(t) \) needs to account for the effect of the unknown disturbances, for the control error induced by the sampling, and for the fact that the control input applied by a neighbor becomes unknown after \( t_{j,l_j(i_{j,k})+1} \). Therefore, \( \sigma_i(t) \) is defined by aggregating three functions that capture these effects. To capture the effect of the disturbances, let

\[
\begin{align*}
\Omega_i(t) &= k_v \int_{t_{i,k}}^{t} \delta(\tau) d\tau + k_p \int_{t_{i,k}}^{t} \int_{t_{i,k}}^{\tau} \delta(\theta) d\theta d\tau, \\
\Psi_i(t) &= \sum_{j \in N_i} w_{ij}(\Omega_{j,i}(t) + \Omega_{j,l_j(i_{j,k})}(t)).
\end{align*}
\]

To capture the effect of the sampling, let

\[
\Theta_i(t) = \left\| \sum_{j \in N_i} w_{ij}(k_p(p_{j}(t) - p_{i}(t)) + k_v(v_{j}(t) - v_{i}(t))) + k_v(\hat{v}_{i,j}(t) - \hat{v}_{i}(t)) - u_{i,k} \right\|.
\]
unknown control inputs. To this aim, let \( R = B_T^T C T^T \), where \( L \) is the Laplacian matrix of the network graph, and \( T \) is such that \( B = B_T T \). (Note that, under Assumption III.2 such a matrix \( T \) exists, since \( B_T \) is full rank.) Then, let

\[
H = \begin{bmatrix}
0_{(N-1) \times (N-1)} & I_{N-1} \\
-k_p R & -k_c R
\end{bmatrix},
\]

where \( k_p \) and \( k_c \) are the control gains used in (5). Under Assumption III.2, it is always possible to choose these gains in such a way that \( H \) is Hurwitz (the interested reader is referred to the Appendix for the proof), and throughout the rest of the paper we shall assume that they are indeed chosen to make \( H \) Hurwitz. Let \( \lambda = -\max \{|\text{Re}(\lambda_H)| : \lambda_H \in \sigma(H)\} \) and \( \eta(\eta_0, t_0, t) \)

\[
e^{-\lambda(t-t_0)} \eta_0 + \sqrt{N} \|B_T\| \int_{t_0}^t e^{-\lambda(t-\tau)}(\varsigma(\tau) + \delta(\tau))d\tau.
\]

(15)

As we shall see, \( \eta(\eta_0, t_0, t) \) constitutes an upper bound for \( \|\xi(t)\| \) whenever \( \eta_0 \geq \|\xi(t_0)\| \), and it is mapped to an upper-bound on the control signals:

\[
\mu_i(t) = \beta_i \eta(\eta_0, 0, t) + \varsigma(t),
\]

(16)

where \( \eta_0 \geq \|\xi(0)\| \) and \( \beta_i = \|C_i T^T K\| \), with \( C_i \) being the \( i \)th row of \( C \), and \( K = I_N \otimes [k_p, k_c] \). Note that a suitable \( \eta_0 \) can be computed by knowing only some bounds on the possible initial conditions. Finally, let \( N_i(t) \) be the subset of \( N_i \) containing the neighbors of \( i \) with unknown control input at time \( t \); namely,

\[
N_{i,k}(t) = \{ j \in N_i : t_{j,i,(t_i,k)+1} + 1 < t \}.
\]

The effect of the unknown control inputs of some neighbors is captured by the function

\[
\Phi_{i,k}(t) = \sum_{j \in N_{i,k}(t)} u_{ij} \left( \int_{t_{j,i,(t_i,k)+1}}^t \mu_j(\tau)d\tau \right)
\]

\[
+ \int_{t_{j,i,(t_i,k)+1}}^t \int_{t_{j,i,(t_i,k)+1}}^\tau \mu_j(\theta)d\theta d\tau,
\]

(17)

We can now define \( \sigma_i(t) \) as

\[
\sigma_i(t) = \Omega_i(t) + \Theta_i(t) + \Phi_i(t),
\]

(18)

and the scheduling rule is given by

\[
t_{i,k+1} = \inf\left\{ t > t_{i,k} : \sigma_i(t) \geq \varsigma(t) \text{ or } \Omega_i(t) \geq \frac{\alpha}{\nu_i} \varsigma(t) \right\},
\]

(19)

Remark IV.2. The parameter \( \alpha \) represents the fraction of the tolerance \( \varsigma(t) \) reserved to the control error caused by the disturbances that have acted on the neighbors of agent \( i \) in the interval \([t_{j,h},t_{i,k}]\). While the choice of \( \alpha \) may influence the number of control updates triggered by the algorithm, the convergence properties hold for any \( \alpha \in (0,1) \).

Remark IV.3. Note that (19) can be evaluated by agent \( i \) when it accesses the cloud (i.e., at time \( t_{i,k} \)) and does not require communication with the other agents. Note also that \( \sigma_i(t) \) is a sum of only linear, quadratic or exponential functions of \( (t-t_{i,k}), (t-t_{j,h}), \) and \( (t-t_{j,h+1}) \), which can be evaluated numerically with the information downloaded from the cloud.

Improved scheduling for a cloud with computational capabilities. If the cloud has some computational capabilities (although they are not needed for the convergence of the proposed control scheme), then it may provide the agents with a tighter upper-bound on \( \|\xi(t)\| \) than \( \eta(\eta_0, 0, t) \). Namely, consider the estimated states \( \hat{\eta}(t) = [\hat{x}_{1N}^T(t) \tau, \ldots, \hat{x}_{N}^N(t) \tau] \) and \( \hat{v}(t) \) (defined similarly), and let \( \tilde{t}(t) = (B_T \otimes I_n)\hat{v}(t) \), \( \check{t}(t) = (B_T \otimes I_n)\hat{v}(t) \), and \( \xi(t) = [\tilde{x}(t), \hat{v}(t)]^T \). Moreover, let

\[
\Delta_i(t) = \int_{t_i,k}^t \delta(\tau)d\tau + \int_{t_i,k}^t \int_{t_i,k}^\tau \delta(\theta)d\theta d\tau,
\]

(20)

\[
\Delta(t) = \left[ \Delta_{1,i}(t), \ldots, \Delta_{N_i,N}(t) \right]^T.
\]

(21)

Note that \( \xi(t) \) and \( \Delta(t) \) (21) can always be computed in the cloud, and, by the triangular inequality, \( \|\xi(t)\| \leq \|\xi(t)\| + \|B_T\| \|\Delta(t)\| \). Hence, if the cloud provides \( \hat{\eta}, \xi(t) \), \( \|\xi(t)\| \), \( \|B_T\| \|\Delta(t)\| \), then agent \( i \) can use

\[
\mu_i(t) = \beta_i \eta(\hat{\eta}_{i,k}, t_{i,k}, t) + \varsigma(t)
\]

(22)

in the scheduling law, instead of (16). However, such information is used only for improving the performances, in the sense of reducing the cloud accesses. The convergence properties of the algorithm still hold if such information is not available, because they only rely on \( \mu_i(t) \) being a valid upper bound for \( u_i(t) \), which is true for both (16) and (22). Hence, in the rest of the paper, all the proofs refer to the case that no global information is computed by the cloud (i.e., (16) is used in the scheduling). The case of \( \hat{n}, \xi(t) \) being computed by the cloud and shared with agent \( i \) upon access \( k \) is easily captured by preliminarily observing that \( \eta(\hat{n}, t_{i,k}, t) \leq \eta(\eta_0, 0, t) \).

V. MAIN RESULT

Our main result is formalized as the following theorem.

Theorem V.1. Consider the multi-agent system (1), with control law (5)–(7) and cloud accesses scheduled by (19). Let Assumptions III.1 and III.2 hold, and let \( k_p \) and \( k_c \) be such that \( H(14) \) is Hurwitz. If \( \varsigma(0) > 0 \), the closed-loop system does not exhibit Zeno behavior and achieves practical consensus over the network graph with

\[
\chi = \frac{\sqrt{N} \|B_T\| (\varsigma(0) + \delta(0))}{\lambda},
\]

(23)

where \( \varsigma(0) \) is the asymptotic value of the threshold function (10), \( \delta(0) \) is the asymptotic value of the disturbance bound (2), \( \lambda \) is defined in Section IV, and \( B_T \) is the incidence matrix of the network graph. If \( \delta(0) = 0, \varsigma(0) = 0 \) and \( \lambda \leq \min\{\lambda, \lambda_b\} \), then the closed-loop system does not exhibit Zeno behavior and achieves asymptotic consensus over the network graph.

Remark V.1. Note that our convergence result (23) is similar to that obtained in related works on event-triggered coordination of multi-agent system, see for example [21]. Here, however, convergence is obtained by using an asynchronously accessed repository, rather than by direct inter-agent communication. Note also that the convergence error represented by
\(\chi\) is distributed across the whole network, which is reflected in that \(\chi\) grows with \(\sqrt{N}\). Such dependence vanishes if we focus on the mean square error attained by a single agent, which is bounded by \(\sqrt{\frac{\chi^2}{N}}\). Finally, note that the edge weights \(w_{ij}\) influence the convergence radius \(\chi\) through the parameter \(\lambda\).

The proof of Theorem V.1 is given in the following three sections of the paper. Namely, in Section VI, we study the convergence properties of the closed-loop system, while in Section VII, we show that the closed-loop system does not exhibit Zeno behavior [22]. Finally, in Section VIII we put the results of Sections VI and VII together to state a formal proof of Theorem V.1.

**VI. CONVERGENCE PROOF**

Our first step in the analysis of the closed-loop system is to rewrite the closed-loop dynamics of the edge-state vector \(\xi(t)\). First, we compare the control signals \(u_{i,k}\) defined by (5) with the ideal diffusive coupling \(z_i(t)\). We can write \(z_i(t)\) in terms of the Laplacian matrix of the network graph as

\[
z_i(t) = -(L_i^T \otimes I_n)(k_{p}\theta(t) + k_{v}v(t)),
\]

where \(L_i^T\) denotes the \(i\)th row of \(L\). Letting \(z(t) = [z_1(t)^T, \ldots, z_N(t)^T]^T\), we can rewrite (24) in the compact form

\[
z(t) = -(L \otimes I_n)(k_{p}\theta(t) + k_{v}v(t)).
\]

Now recall that \(L = CBT\), and that, since \(BT\) is full column rank, there exists a matrix \(T\) such that \(B = BT T\) (namely, \(T = B_T(B_T B_T)^{-1} B\)). Therefore,

\[
z(t) = -(CT^T B_T \otimes I_n)(k_{p}\theta(t) + k_{v}v(t)).
\]

By the mixed-product property of the Kronecker product, (26) can be rewritten in terms of the edge states as

\[
z(t) = -(CT^T \otimes I_n)(k_{p}x(t) + k_{v}y(t)).
\]

Let \(\tilde{u}_{i}(t)\) be the mismatch between the control input of agent \(i\) and \(z_i(t)\), namely,

\[
\tilde{u}_{i}(t) = u_i(t) - z_i(t).
\]

We denote \(\tilde{u}(t) = [\tilde{u}_1(t)^T, \ldots, \tilde{u}_N(t)^T]^T\), so that we can rewrite (28) as

\[
\tilde{u}(t) = u(t) - z(t).
\]

From (29) and (26), we have

\[
u(t) = (C BT \otimes I_n)(k_{p}x(t) + k_{v}y(t)) + \tilde{u}(t),\]

which substituted in (8) yields

\[
\dot{x}(t) = y(t),
\]

\[
\dot{y}(t) = - (B_{T}^T \otimes I_n)(C BT \otimes I_n)(k_{p}x(t) + k_{v}y(t)) + (B_{T}^T \otimes I_n)(\tilde{u}(t) + d(t)).
\]

Having introduced \(R = B_{T}^T C BT\) in Section IV, we can use the mixed-product property of the Kronecker product to rewrite (31) as

\[
\dot{x}(t) = y(t),
\]

\[
\dot{y}(t) = - (R \otimes I_n)(k_{p}x(t) + k_{v}y(t)) + (B_{T}^T \otimes I_n)(\tilde{u}(t) + d(t)).
\]

Recalling that \(\xi(t) = [x(t)^T, y(t)^T]^T\), 26 and (32) can be rewritten compactly as

\[
z(t) = -(\underbrace{(CT^T K) \otimes I_n}_G)\xi(t),
\]

\[
\tilde{\xi}(t) = (H \otimes I_n)\xi(t) + (G \otimes I_n)(\tilde{u}(t) + d(t)),
\]

where \(H\) is the Hurwitz matrix defined in (14), \(G = [0^{T(N-1)}_{1} N, B_T]^T\), and \(K = I_N \otimes [k_{p}, k_{v}]\).

The following Lemma relates a bound on the control errors \(\tilde{u}(t)\) to a bound on the state error vector \(\xi(t)\) and on the control signals \(u_i(t)\).

**Lemma VI.1.** Consider the multi-agent system (1), and let Assumptions III.1 and III.2 hold. Suppose that

\[
\|\tilde{u}_i(\tau)\| \leq \varsigma(\tau)
\]

for all \(\tau \in [0, t]\) and all \(i \in \mathcal{V}\), where \(\varsigma(\tau)\) is defined by (28) and \(\varsigma(\cdot)\) is the threshold function (10). Then, for all \(\tau \in [0, t]\), we have \(\|\tilde{\xi}(t)\| \leq \eta(\tau_0, 0, \mu_0)\eta(\varsigma(\tau), 1, \varsigma(\tau))\) where \(\varsigma(\cdot, \cdot, \cdot)\) is defined by (15), and \(\|u_j(\tau)\| \leq \mu_j(\tau)\) for all \(j \in \mathcal{V}\).

**Proof.** The Laplace solution of (34) reads

\[
\xi(\tau) = e^{\int_{t_0}^{\tau} G \otimes I_n}(\tilde{u}(\theta) + d(\theta))d\theta.
\]

Taking norms of both sides in (36), and using (35), Assumption (III.1), the properties of the Kronecker product, and the triangular inequality, and observing that \(\|e^{F_{0} \otimes (\tau - t_0)}\| \leq e^{-\lambda(\tau - t_0)}\), and that \(\|G\| = \|B_T\|\), we have \(\|\xi(t)\| \leq \eta(\tau_0, 0, \tau)\). Moreover, from (28), we have \(u_i(\tau) = z_i(\tau) + \tilde{u}_i(\tau).\) Taking norms of both sides, and using the triangular inequality, we have \(\|u_i(\tau)\| \leq \|z_i(\tau)\| + \|\tilde{u}_i(\tau)\|\). Selecting the rows corresponding to the \(j\)th agent in (33), we have \(z_j(\tau) = (C_T^T K) \otimes I_n)\tilde{x}(\tau)\), where \(C_T\) denotes the \(j\)th row of \(C\). Taking norms of both sides, and substituting the result in the previous inequality, we have \(\|u_j(\tau)\| \leq \beta_j(\|\tilde{x}(\tau)\| + \|\tilde{u}_i(\tau)\|)\). The proof is concluded by using \(\|\tilde{\xi}(\tau)\| \leq \eta(\tau_0, 0, \tau)\) and (35), to obtain \(\|u_j(\tau)\| \leq \mu_j(\tau)\).

The following Lemma VI.2 shows that, under the scheduling rule (19), we can guarantee that \(\|\tilde{u}_i(\tau)\| \leq \|\xi(\tau)\|\) for all agents, thus satisfying the hypotheses of Lemma VI.1.

**Lemma VI.2.** Consider the multi-agent system (1) under the control law (5)-(7) and the scheduling rule (19). Under Assumption III.1, \(\|\tilde{u}_i(\tau)\| \leq \varsigma(\tau)\) for all \(t \geq 0\) and all \(i \in \mathcal{V}\).

**Proof.** Since (19) guarantees \(\sigma_{\tau,i}(t) \leq \varsigma(t)\) for all \(t \geq 0\) and all \(i \in \mathcal{V}\), we only need to show that \(\|\tilde{u}_i(t)\| \leq \sigma_{\tau,i}(t)\). For each \(t \geq 0\) and all \(i \in \mathcal{V}\). Without loss of generality, let \(l_i(t) = k,\) and consider \(t \in [t_{i,k}, t_{i,k+1})\). Using (5) and (9) in (28), we have

\[
\tilde{u}_i(t) = u_{i,k} - \sum_{j \in \mathcal{N}_i} w_{ij}(k_p(p_j(t) - p_i(t)) + k_v(v_j(t) - v_i(t))).
\]

To show that \(\|\tilde{u}_i(t)\| \leq \sigma_{i,k}(t)\), we shall break down the terms \(v_i(t), p_i(t), v_j(t)\) and \(p_j(t)\) in (37). First, consider the term \(v_i(t)\). Integrating (1b) in \([t_{i,k}, t]\), and using (6a), we have

\[
v_i(t) = \tilde{v}_{i,k}(t) + \int_{t_{i,k}}^{t} d_i(t)\tau.
\]
Second, consider the term \( p_i(t) \). Integrating (1a) in \([t_{i,k}, t]\), we have
\[
p_i(t) - p_{i,k} = \int_{t_{i,k}}^{t} v_i(\tau) d\tau,
\]
which using (38) and (6c) can be rewritten as
\[
p_i(t) = \hat{p}_{i,k}^{j}(t) + \int_{t_{i,k}}^{t} d_i(\theta) d\theta d\tau.
\]
(40)

Third, consider the term \( v_j(t) \). Without loss of generality, let \( t_j(t_{i,k}) = h \). Integrating (1b) for agent \( j \) in \([t_{j,h}, t]\), we have
\[
v_j(t) = v_{j,h} + \int_{t_{j,h}}^{t} u_j(\tau) d\tau + \int_{t_{j,h}}^{t} d_j(\tau) d\tau.
\]
(41)

Here we need to distinguish two cases: namely \( t \leq t_{j,h+1} \) or \( t > t_{j,h+1} \). In the first case, we have \( u_j(\tau) = u_{j,t_{j,h}}(\tau) \) for all \( \tau \in [t_{i,k}, t] \), and therefore (41) becomes, also using (6a),
\[
v_j(t) = \hat{v}_{j,k}^{i}(t) + \int_{t_{j,h}}^{t} d_j(\tau) d\tau.
\]
(42)

In the second case, we have \( u_j(\tau) = u_{j,t_{j,h}}(\tau) \) for \( \tau \in [t_{i,k}, t_{j,h,t_{i,k}}(t_{i,k})] \), so we can rewrite (41) as
\[
v_j(t) = \hat{v}_{j,k}^{i}(t) + \int_{t_{j,h}}^{t} u_j(\tau) d\tau + \int_{t_{j,h}}^{t} d_j(\tau) d\tau,
\]
(43)

where again we have also used (6a). Last, consider the term \( p_j(t) \). Integrating (1a) for agent \( j \) in \([t_{j,h}, t]\), and using (41), we have
\[
p_j(t) = p_{j,h} + (t - t_{j,h}) u_{j,h} + \int_{t_{j,h}}^{t} (u_j(\theta) + d_j(\theta)) d\theta d\tau.
\]
(44)

Again, we need to distinguish the two cases \( t \leq t_{j,h+1} \) and \( t > t_{j,h+1} \). In the first case, we have simply \( u_j(\theta) = u_{j,h} \) for all \( \theta \in [t_{j,h}, t] \) and all \( \tau \in [t_{i,k}, t] \); therefore, (44) becomes, using (6c),
\[
p_j(t) = \hat{p}_{j,k}^{i}(t) + \int_{t_{j,h}}^{t} d_j(\theta) d\theta d\tau.
\]
(45)

In the second case, the control input of agent \( j \) is not known for \( t > t_{j,h+1} \), and therefore, it does not contribute to the estimate \( \hat{p}_{j,k}^{i}(t) \); namely, a few passages show that in this case, using (6c), (44) becomes
\[
p_j(t) = \hat{p}_{j,k}^{i}(t) + \int_{t_{j,h}}^{t} u_j(\tau) d\tau + \int_{t_{j,h}}^{t} d_j(\tau) d\theta d\tau.
\]
(46)

Using (38), (40), (42), (43), (45) and (46) in (37), we have
\[
\hat{u}_i(t) = u_{i,k} - \sum_{j \in \mathcal{N}_i(t_{i,k})} u_{j,i,k} - \sum_{j \in \mathcal{N}_j(t_{i,k})} u_{j,i,k} - \int_{t_{i,k}}^{t} d_{i}(\theta) d\tau + \int_{t_{i,k}}^{t} d_{j}(\theta) d\theta d\tau
\]
\[
- \hat{p}_{i,k}^{j}(t) - \int_{t_{i,k}}^{t} d_{i}(\tau) d\tau + k_v \int_{t_{i,k}}^{t} \hat{v}_{j,k}^{i}(t) d\theta d\tau
\]
\[
+ \int_{t_{i,k}}^{t} d_{j}(\tau) d\tau - \hat{v}_{j,k}^{i}(t) - \int_{t_{i,k}}^{t} d_{j}(\tau) d\tau
\]
\[
- \sum_{j \in \mathcal{N}_j(t_{i,k})} u_{j,i,k} - \int_{t_{j,h}}^{t} u_{j,i,k} d\tau + k_p \int_{t_{j,h}}^{t} d_{j}(\tau) d\theta d\tau.
\]
(47)

Now we can take norms of both sides in (47), use the triangular inequality, use Assumption III.1 to bound the disturbance terms, and use \( \|u_j(\tau)\| \leq \mu_j(\tau) \) to bound the unknown control terms. Altogether, we obtain
\[
\|\hat{u}_i(t)\| \leq \Theta_{i,k}(t) + \Psi_{i,k}(t) + \Phi_{i,k}(t) = \sigma_{i,k}(t),
\]
(48)

where \( \Theta_{i,k}(\cdot), \Psi_{i,k}(\cdot), \Phi_{i,k}(\cdot), \) and \( \sigma_{i,k}(\cdot) \) have been defined in Section IV. Observing that the scheduling rule (19) imposes \( \sigma_{i,k}(t) \leq \varsigma(t) \) concludes the proof.

**VII. WELL-POSEDNESS PROOF**

The second step in our analysis is to prove that the closed-loop system is well posed, in the sense that the sequence of the updates \( t_{i,k} \) for \( k \in \mathbb{N}_0 \) does not present Zeno behavior for any of the agents. We are going to distinguish two cases, namely \( \varsigma_{\infty} > 0 \) and \( \varsigma_{\infty} = 0 \), where \( \varsigma_{\infty} \) is the asymptotic value of the threshold function (10).

**Lemma VII.1.** Consider the multi-agent system (1), with control law (5)–(7) and cloud accesses scheduled by (19). Let \( k_p \) and \( k_v \) be chosen in such a way that \( H \) is Hurwitz and choose \( \varsigma_{\infty} > 0 \) in (10). Then, under Assumptions III.1 and III.2, the closed-loop system does not exhibit Zeno behavior.

**Proof.** Without loss of generality, let \( t \in [t_{i,k}, t_{i,k+1}] \) and \( h = l_j(t_{i,k}) \). We are going to show that there exists a lower bound for the interval \( t_{i,k+1} - t_{i,k} \). First consider the triggering condition \( \Omega_{i,k}(t) \geq (\alpha/\nu_1)\varsigma(t) \). We can use (2) and (11) to compute \( \Omega_{i,k}(t) \) explicitly as
\[
\Omega_{i,k}(t) = \frac{\delta_0 - \delta_{\infty}}{\lambda_{\delta}} \left( \frac{v_k(1 - e^{-\lambda_{\delta} \varsigma(t - t_{i,k})})}{\lambda_{\delta}} + k_p \left( (t - t_{i,k}) + \frac{1 - e^{-\lambda_{\delta}(t - t_{i,k})}}{\lambda_{\delta}} \right) \right)
\]
(49)
\[
+ \delta_{\infty}(t - t_{i,k}) (v_k + 0.5 k_p (t - t_{i,k})).
\]

Since \( \varsigma(t) \) can be lower-bounded as \( \varsigma(t) \geq \varsigma_{\infty} \), a necessary condition to have \( \Omega_{i,k}(t) \geq (\alpha/\nu_1)\varsigma(t) \) is that the right-hand side of (49) is larger than \( \varsigma_{\infty} \). This condition implies that \( t \geq t_{i,k} + \tau^*_i \), where \( \tau^*_i \) is the smallest (strictly) positive solution of
\[
\frac{\delta_0 - \delta_{\infty}}{\lambda_{\delta}} \left( \frac{v_k(1 - e^{-\lambda_{\delta} \varsigma})}{\lambda_{\delta}} + k_p \left( \tau + \frac{1 - e^{-\lambda_{\delta}\tau}}{\lambda_{\delta}} \right) \right)
\]
\[
+ \delta_{\infty} \tau (v_k + 0.5 k_p \tau) = (\alpha/\nu_1)\varsigma_{\infty}.
\]
(50)

Now we produce a similar argument for the triggering condition \( \sigma_{i,k}(t) \geq \varsigma(t) \). First consider the term \( \Psi_{i,k}(t) \). Note that, evaluating (11) for agent \( j \), and splitting the integration interval \([t_j,h, t]\) into \([t_j,h, t_{j,k}]\) and \([t_{j,k}, t]\), we have
\[
\Omega_{j,h}(t) = \frac{\Omega_{j,h}(t) + \Omega_{j,k}(t) + k_p \int_{t_{j,k}}^{t} d_j(\tau) d\theta d\tau}{t_{j,h}}.
\]
(51)

Splitting the integration interval \([t_j,h, t]\) further in \([t_j,h, t_{j,k}]\) and \([t_{j,k}, t]\), we have
\[
\Omega_{j,h}(t) = \frac{\Omega_{j,h}(t) + \Omega_{j,k}(t) + k_p \int_{t_{j,k}}^{t} d_j(\tau) d\theta d\tau}{t_{j,h}}.
\]
(52)

Observing that \( \delta(\theta) \leq \delta_{\infty} \), and that \( t_{i,k} - t_{j,h} \leq t_{j,h+1} - t_{j,h} \),
we have \( \int_{t_{i,h}}^{t_{i+k}} \delta(t) d\theta \leq \delta_0 \tau_{j}^* \), where \( \tau_{j}^* \) denotes the smallest (strictly) positive solution of (50), evaluated for agent \( j \). Hence, we can upper-bound (52) as
\[
\Omega_{j,h}(t) \leq \Omega_{j,h}(t_{i,k}) + \Omega_{i,k}(t) + k_p \delta_0 \tau_{j}^*(t - t_{i,k}). \tag{53}
\]
Thanks to the scheduling rule (19), we have \( \Omega_{j,h}(t_{i,k}) \leq (\alpha/\nu_j) \varsigma(t_{i,k}) \), which substituted in (53) yields
\[
\Omega_{j,h}(t) \leq (\alpha/\nu_j) \varsigma(t_{i,k}) + \Omega_{i,k}(t) + k_p \delta_0 \tau_{j}^*(t - t_{i,k}). \tag{54}
\]
Using (54) in (20), we can upper-bound \( \Psi_{i,k}(t) \) as
\[
\Psi_{i,k}(t) \leq \sum_{j \in \mathcal{N}_i} \left( \frac{\nu_j}{\nu_j} \varsigma(t_{i,k}) + \Omega_{i,k}(t) + k_p \delta_0 \tau_{j}^*(t - t_{i,k}) \right). \tag{55}
\]
Since, by the definition of \( \nu_j \), \( \sum_{j \in \mathcal{N}_i} (w_{ij}/\nu_j) \leq 1 \), we can further bound (55) as
\[
\Psi_{i,k}(t) \leq \alpha \varsigma(t_{i,k}) + \sum_{j \in \mathcal{N}_i} \left( \sum_{j \in \mathcal{N}_i} w_{ij} \right) (t - t_{j,h+1} + 0.5(t - t_{j,h+1})^2). \tag{56}
\]
Next, consider the term \( \Phi_{i,k}(t) \). Since \( \eta(t_0, t_0, t) \) is an upper-bounded function of \( t \), we can denote its largest value of \( \eta(t_0, t_0, t) \) for \( t \geq 0 \), which, by observing also that \( \varsigma(t) \leq \varsigma_0 \), allows us to bound \( \mu_j(t) \) as \( \mu_j(t) \leq \beta_j \varsigma_0 + \varsigma_0 \). Consequently, from (17), we have
\[
\Phi_{i,k}(t) \leq \sum_{j \in \mathcal{N}_i^c(t)} \left( \sum_{j \in \mathcal{N}_i} w_{ij} \right) (t - t_{j,h+1} + 0.5(t - t_{j,h+1})^2). \tag{57}
\]
Last, consider the term \( \Theta_{i,k}(t) \), and note that, using (7), it can be written as
\[
\Theta_{i,k}(t) = \left\| \sum_{j \in \mathcal{N}_i} w_{ij} \left( k_v h_j \left( \int_{t_{i,k}}^{t_{j,h}} u_{j,k} d\tau - \int_{t_{i,k}}^{t_j} u_{i,k} d\tau \right) \right) + \delta_0 e^{-\lambda_j t_{i,k}} \left( k_v (1 - e^{-\lambda_j (t - t_{i,k}))} \right) \right\|. \tag{58}
\]
Using (6), the right-hand side of (58) can be rewritten as
\[
\Theta_{i,k}(t) = \left\| \sum_{j \in \mathcal{N}_i} w_{ij} \left( k_v \left( \int_{t_{i,k}}^{t_{j,h}} u_{j,k} d\tau - \int_{t_{i,k}}^{t_j} u_{i,k} d\tau \right) + k_p \left( \int_{t_{i,k}}^{t_{j,h}} u_{j,k} d\tau - \int_{t_{i,k}}^{t_j} u_{i,k} d\tau \right) \right) \right\|. \tag{59}
\]
where we have denoted \( t'_{j,h} = \min\{t, t_{j,h+1}\} \) for brevity. Since each control input is upper-bounded as \( \|u_i(t)\| \leq \mu_i(t) \leq \beta_i \varsigma_0 + \varsigma_0 \), using the triangular inequality on the right-hand side of (59), summing by side by side with (57), yields
\[
\Theta_{i,k}(t) + \Phi_{i,k}(t) \leq \sum_{j \in \mathcal{N}_i} \left( w_{ij} \left( (\beta_j + \beta_i) \varsigma_0 + 2 \varsigma_0 \right) \right) k_v (t - t_{i,k}) + 0.5 k_p (t - t_{i,k})^2. \tag{60}
\]
Now we can sum the right-hand sides of (60) and (55) to obtain an upper bound for \( \sigma_{i,k}(t) \). To keep the notation compact, note—considering also (49)—that this upper bound only contains terms of the types \( t - t_{i,k} \), \( (t - t_{i,k})^2 \) and \( 1 - e^{\lambda_j (t - t_{i,k})} \), plus the term \( \alpha \varsigma(t_{i,k}) \), so that we can write
\[
\sigma_{i,k}(t) \leq \sigma_1 (t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 \left( 1 - e^{-\lambda_j (t - t_{i,k})} \right) + \alpha \varsigma(t_{i,k}), \tag{61}
\]
with \( \sigma_1, \sigma_2, \sigma_3 > 0 \). From (61), it is clear that a necessary condition to have \( \sigma_{i,k}(t) \geq \varsigma(t) \) is
\[
\sigma_1 (t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 \left( 1 - e^{-\lambda_j (t - t_{i,k})} \right) + \alpha \varsigma(t_{i,k}) \geq \varsigma(t). \tag{62}
\]
Since \( \varsigma(t) \geq \varsigma(t_{i,k}) - \lambda_j (\varsigma_0 - \varsigma_\infty) (t - t_{i,k}) \), (62) implies
\[
(\sigma_1 + \lambda_j (\varsigma_0 - \varsigma_\infty))(t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 \left( 1 - e^{-\lambda_j (t - t_{i,k})} \right) \geq (1 - \alpha) \varsigma(t_{i,k}). \tag{63}
\]
Finally, observing that \( \varsigma(t_{i,k}) \geq \varsigma_\infty \), (63) implies \( t_{i,k+1} \geq t_{i,k} + \tau_{i,k}^* \), where \( \tau_{i,k}^* \) is the smallest (strictly) positive solution of \( \sigma_1 + \lambda_j (\varsigma_0 - \varsigma_\infty) \tau + \sigma_2 \tau^2 + \sigma_3 (1 - e^{-\lambda_j \tau}) \geq (1 - \alpha) \varsigma_{\infty} \). Since \( t_{i,k+1} \) is defined as the smallest time when either \( \Omega_{j,h}(t) \geq \varsigma(t) \) or \( \sigma_{i,k}(t) \geq \varsigma(t) \), and both these conditions require a finite value of \( t - t_{i,k} \), we can conclude that the scheduling law (19) does not induce Zeno behavior, and guarantees a positive lower bound between two consecutive updates.
\[
\square
\]

**Lemma VII.2.** Consider the multi-agent system (1), with control law (5)–(7), and cloud access schedules rendered by (19).

Let Assumptions III.2 and III.1 hold, with \( \delta_0 = 0 \) in Assumption III.1. Choose \( k_p \) and \( k_v \) such that \( H \) is Hurwitz, and choose \( \varsigma_\infty = 0 \) and \( \lambda_j < \min\{\lambda, \lambda_\delta\} \). Then, the closed-loop system does not exhibit Zeno behavior.

**Proof.** Using \( \delta_\infty = 0 \) and \( \lambda_\delta \geq \lambda_j \) in (11), we obtain
\[
\Omega_{i,k}(t) \leq \delta_0 e^{-\lambda_j t_{i,k}} \left( k_v (1 - e^{-\lambda_j (t - t_{i,k})}) + k_p \left( t - t_{i,k} + \frac{1 - e^{-\lambda_j (t - t_{i,k})}}{\lambda_j} \right) \right) \tag{64}
\]
Moreover, observe that, with \( \varsigma_\infty = 0 \), the threshold function can be written as
\[
\varsigma(t) = \varsigma_0 e^{-\lambda_j t} = \varsigma_0 e^{-\lambda_j t_{i,k}} e^{-\lambda_j (t - t_{i,k})}. \tag{65}
\]
Inequalities (64) and (65) show that the triggering condition \( \Omega_{i,k}(t) \geq (\alpha / \nu_j) \varsigma(t) \) implies that \( t \geq t_{i,k} + \tau_{i,k}^* \), where \( \tau_{i,k}^* \) is the smallest (strictly) positive solution of
\[
\delta_0 \left( k_v + \frac{k_p}{\lambda_j} \right) \left( 1 - e^{-\lambda_j \tau} \right) + k_p \tau \geq \varsigma_0 \nu_j e^{-\lambda_j \tau}. \tag{66}
\]
Reasoning as in Lemma VII.1, we find again that (52) holds, but since by hypothesis \( \delta(\theta) = \delta_0 e^{-\lambda_j \theta} \leq \delta_0 e^{-\lambda_j \theta} \), we compute therein
\[
\int_{t_{i,k}}^{t} \int_{t_{j,h}}^{t_{j,h+1}} \delta(t) d\theta d\tau \leq \delta_0 \left( e^{\lambda_j (t_{i,k} - t_{j,h+1}) - 1} - e^{\lambda_j (t_{i,k} - t_{j,h})} \right). \tag{67}
\]
Recalling that \( t_{i,k} - t_{j,h} \leq t_{j,h+1} - t_{j,h} \leq \tau_{j}^* \) allows us to
upper-bound (67) as $(\delta_0/\lambda_\varsigma)(e^{\lambda_\varsigma \tau^*_j} - 1)e^{-\lambda_i t_i,k}(t-t_i,k)$, which
substituted in (52) yields
\[ \Omega_{j,h}(t) \leq \Omega_{j,h}(t_{i,k}) + \Omega_{i,k}(t) + k_p(\delta_0/\lambda_\varsigma)(e^{\lambda_\varsigma \tau^*_j} - 1)e^{-\lambda_i t_i,k}(t-t_i,k). \] (68)

Similarly as done in Lemma VII.1, we can now use (68) in (20), obtaining
\[ \Psi_{i,k}(t) \leq \alpha\varsigma(t_{i,k}) + \left(\sum_{j \in N_i} w_{ij}\right)\Omega_{i,k}(t) + k_p(\delta_0/\lambda_\varsigma)(\sum_{j \in N_i} w_{ij})^2(t-t_{i,k}). \] (69)

Now note that, using (15) with $\delta_\infty = 0$ and $\varsigma_\infty = 0$, $\eta(t)$ can be upper-bounded by the slowest exponential among $e^{-\lambda_i t}$, $e^{-\lambda_\varsigma \tau^*_j}$ and $e^{-\lambda_i t}$. Since by hypothesis $\lambda_\varsigma \leq \min\{\lambda, \lambda_3\}$, we can write $\eta(t) \leq \eta e^{-\lambda_\varsigma \tau^*_j}$, where $\eta > 0$ depends on the initial conditions. Consequently, we can upper-bound the control inputs as
\[ \|u_i(t)\| \leq \beta_i \eta(t) + \varsigma(t) \leq (\beta_i \eta + \varsigma_0)e^{-\lambda_i t_{i,k}}. \] (70)

Using (70) in (17), we have
\[ \Phi_{i,k}(t) \leq \sum_{j \in N_i} w_{ij}(\beta_j + \beta_i)\eta + \varsigma_0. \] (71)

Similarly as done in Lemma VII.1, we can now sum (59) and (71) side by side, then use the triangular inequality and (71) to obtain
\[ \Theta_{i,k}(t) + \Phi_{i,k}(t) \leq \sum_{j \in N_i} w_{ij}(\beta_j + \beta_i)\eta + \varsigma_0 + \left(k_\nu + \frac{k_p}{\lambda_\varsigma}\right)(t-t_{j,h+1} + 0.5(t-t_{j,h+1})^2). \] (72)

By summing the right-hand sides of (69) and (72), considering also (49), and observing that $\varsigma(t_{i,k}) = \varsigma_0 e^{-\lambda_i t_{i,k}}$, we have an upper bound for $\sigma_{i,k}(t)$ in the form
\[ \sigma_{i,k}(t) \leq \sigma_1(t-t_{i,k}) + \sigma_2(t-t_{i,k})^2 + \sigma_3(1-e^{-\lambda_i(t-t_{i,k})})e^{-\lambda_i t_{i,k}} + \alpha\varsigma(t_{i,k}). \] (73)

From (73), we reason as in Lemma VII.1 to show that $\sigma_{i,k}(t) \geq \varsigma(t)$ implies $t \geq t_{i,k} + \tau^*_i$, where $\tau^*_i$ is the smallest (strictly) positive solution of $\sigma_1 + \sigma_2 t + \sigma_3(1-e^{-\lambda_i t}) + \alpha \varsigma = 0$. Since $t_{i,k+1}$ is defined as the smallest time when either $\Omega_{i,k}(t) \geq \varsigma(t)$ or $\sigma_{i,k}(t) \geq \varsigma(t)$, and both these conditions require a finite value of $t-t_{i,k}$, we can conclude that the scheduling law (19) does not exhibit Zeno behavior, and guarantees a positive lower bound between two consecutive updates.

VIII. PROOF OF THEOREM V.1

We are now ready to prove Theorem V.1 by using the results developed in the previous two sections. From Lemma VI.2, we know that, under the control law (5)–(7) and the scheduling rule (19) the hypotheses of Lemma VI.1 are satisfied.

If $\delta_\infty > 0$, we know from Lemma VII.1 that the closed-loop system does not exhibit Zeno behavior. Therefore, we can take $t \to \infty$ in (15) in Lemma VI.1, obtaining
\[ \limsup_{t \to \infty} \|\xi(t)\| \leq \chi \] with $\chi$ given by (23). If $\delta_\infty = 0$, $\varsigma_\infty = 0$, we know from Lemma (VII.2) that the closed-loop system does not exhibit Zeno behavior. Therefore, we can take again $t \to \infty$ in (15), obtaining
\[ \limsup_{t \to \infty} \|\xi(t)\| \leq \chi. \] But since $\delta_\infty = \varsigma_\infty = 0$, (23) evaluates to zero, and therefore $\lim_{t \to \infty} \xi(t) = 0$.

IX. NUMERICAL SIMULATIONS

In this section, two numerical simulations of the proposed control algorithm are presented, one for a scenario where practical convergence is reached, and one for a scenario where asymptotic convergence is reached. For both simulations, we consider a multi-agent system made up of $N = 4$ agents with state in $\mathbb{R}^2$, which exchange information through a cloud repository according to the graph $G$ illustrated in Figure 1, where all the edges are assigned unitary weights. The assigned graph contains a spanning tree $T$ made up of the first three edges. The corresponding matrix $R$ is
\[ R = \begin{bmatrix} 2 & 0 & -1 \\ -1 & 2 & 1 \\ 0 & -1 & 1 \end{bmatrix}. \] (74)

The control gains are chosen as $k_p = 0.5$ and $k_d = 1.0$, which leads to $\lambda = -\max\{\Re(s) : s \in \text{eig}(H)\} = 0.5$ and $\|B_T\| \approx 2.45$. The disturbances are chosen as
\[ d_i(t) = \delta(t) \begin{bmatrix} \cos(2\pi(i/N)t + 2\pi((N-i)/N)) \\ \sin(2\pi(i/N)t + 2\pi((N-i)/N)) \end{bmatrix}, \] (75)

where $\delta(t)$ is defined by Assumption III.1 with $\theta_0 = 0.2$, $\lambda_\delta = 0.45$, $\varsigma_\infty = 0.02$ in the first simulation, and $\delta_\infty = 0$ in the second simulation. It is easy to see that, with these parameters, Assumption III.1 is satisfied. The threshold function is chosen as (10), with $\varsigma_0 = 5.0$, $\lambda_\varsigma = 0.4$, $\varsigma_\infty = 0.5$ for the first simulation, and $\varsigma_\infty = 0$ for the second simulation. Note that, with these choices, the first simulation scenario satisfies the hypotheses of Theorem V.1 for practical consensus, and the second simulation scenario satisfies the hypotheses of Theorem V.1 for asymptotic consensus. For the coefficient $\alpha$ that appears in (19), we choose $\alpha = 0.05$. The upper bounds on the control signals are computed as (22).

The results of the first simulation are illustrated in Figure 3. From Figure 3 it looks clear that the multi-agent system only achieves practical convergence, but the norm of the disagreement vector is significantly reduced. From Figure 3, we can also see that the cloud accesses do not accumulate; on the contrary, they seem to become less frequent over time, which corroborates the result that the closed-loop system does not exhibit Zeno behavior. The results of the second simulation are illustrated in Figure 4. From Figure 4 it looks clear that $\xi(t) \to 0$, which means that asymptotic convergence is reached. From Figure 4, we can also see that the cloud accesses do not accumulate even if the threshold function is converging to zero, which again corroborates the result that the closed-loop system does not exhibit Zeno behavior.
beeen proved by showing that there is a lower bound for the time interval between two consecutive accesses to the cloud. The proposed scheme can be adopted in all cases when direct communication among agents is interdicted, as illustrated in our motivating example of controlling a fleet of AUVs.

Future work will address possible imperfections in the communication with the repository, such as time delays and packet losses, as well as more complex control objectives.

**APPENDIX - PROOF THAT \( H \) IS HURWITZ**

Consider the matrix \( F = \begin{bmatrix} 0_{N \times N} & I_N \\ -k_p L & -k_v L \end{bmatrix} \), where \( L \) is the Laplacian matrix of the graph \( \mathcal{G} \). A well-known result in multi-agent coordination is that, under Assumption III.2, \( k_p \) and \( k_v \) can always be chosen in such a way that \( F \) has exactly \( 2(N-1) \) eigenvalues with negative real parts (counted with their multiplicities) and a double eigenvalue in zero [23]. But \( H \in \mathbb{R}^{2(N-1) \times 2(N-1)} \), therefore, it has exactly \( 2(N-1) \) eigenvalues (counted with their multiplicity). Therefore, if we show that \( F \) and \( H \) have the same nonzero eigenvalues with the same multiplicities, then we can conclude that \( H \) is Hurwitz. Using the rule for the determinant of block-diagonal matrices\(^1\), we can compute the characteristic polynomial of \( F \) is

\[
\mathcal{P}(\lambda) = \det(\lambda I_{2N} - F) = \det\left(\lambda^2 I_N + (\lambda k_v + k_p)L\right),
\]

which for \( \lambda \neq 0 \) can be written as

\[
\mathcal{P}(\lambda) = \lambda^{2N} \det(I_N + (\lambda k_v + k_p)/\lambda^2 L).
\]

Now consider the matrix \( F_e = \begin{bmatrix} 0_{M \times N} & I_M \\ -k_p L_e & -k_v L_e \end{bmatrix} \), where \( L_e = B^T C \) is the edge Laplacian of \( \mathcal{G} \). Similarly as done for \( F \), we can compute the characteristic polynomial of \( F_e \) as

\[
\mathcal{P}_e(\lambda) = \det(\lambda^2 I_M + (\lambda k_v + k_p)L_e),
\]

which for \( \lambda \neq 0 \) can be rewritten as

\[
\mathcal{P}_e(\lambda) = \lambda^{2M} \det(I_M + (\lambda k_v + k_p)/\lambda^2 L_e).
\]

Since \( L = CB^T \) and \( L_e = B^T C \), by (77), (79) and Sylvester’s determinant identity\(^2\), we have \( \mathcal{P}(\lambda)/\lambda^{2N} = \mathcal{P}_e(\lambda)/\lambda^{2M} \) for any \( \lambda \neq 0 \), which implies that \( F \) and \( F_e \) have the same nonzero eigenvalues with the same multiplicity. Therefore, we only need to prove that \( F_e \) and \( H \) have the same nonzero eigenvalues with the same multiplicity. To this aim, consider the matrix

\[
S = \begin{bmatrix} I_{N-1} & 0_{(N-1) \times (M-N+1)} \\ -T^T & I_{M-N+1} \end{bmatrix},
\]

and note that

\[
SL_e S^{-1} = \begin{bmatrix} R & 0_{(M-N+1) \times (N-1)} \\ 0_{(M-N+1) \times (M-N+1)} & * \end{bmatrix}.
\]

\(^1\) \( \det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(AD - BC) \) whenever \( C \) and \( D \) commute [24].

\(^2\) \( \det(I_n + AB) = \det(I_m + BA) \) for appropriate \( n, m \in \mathbb{N} \) [25].
Multiplying the right-hand side of (78) by $\det(S) \det(S^{-1}) = 1$, and using (81), we have

$$\mathcal{P}_c(\lambda) = \det(S(\lambda^2 I_M + (\lambda k_v + k_p) L_c) S^{-1})$$

$$= \lambda^{2(M-(N-1))} \det(\lambda^2 I_{N-1} + (\lambda k_v + k_p) R)$$

$$= \lambda^{2(M-(N-1))} \mathcal{P}_{c,r}(\lambda),$$

where $\mathcal{P}_{c,r}(\lambda)$ is the characteristic polynomial of $H$. Therefore, $P_c$ and $H$ have the same nonzero eigenvalues with the same multiplicity, which concludes the proof.

REFERENCES

ation with event-based communication,” in American Control Conference, 2011.
chronous cloud access,” in American Control Conference, 2016.
agent systems via asynchronous cloud communication,” in IEEE Confer-
ence on Decision and Control, 2016.

Karl H. Johansson is Director of the Stockholm Strategic Research Area ICT The Next Generation and Professor at the School of Electrical Engineering, KTH Royal Institute of Technology. He received MSc and PhD degrees in Electrical Engineering from Lund University. His research interests are in networked control systems, cyber-physical systems, and applications in transportation, energy, and automation. He is a member of the IEEE Control Systems Society Board of Governors and the European Control Association Council. He has received several distinctions, including a ten-year Wallenberg Scholar Grant, the Future Research Leader Award from the Swedish Foundation for Strategic Research, and the triennial Young Author Prize from IFAC. He is member of the Royal Swedish Academy of Engineering Sciences, Fellow of the IEEE, and IEEE Distinguished Lecturer.

ADALDO et al.: CLOUD-SUPPORTED COORDINATION OF SECOND-ORDER MULTI-AGENT SYSTEMS 11

Antonio Adaldo received his M.Sc. degree in Au-
tomation Engineering in Dec. 2013 from the Uni-
versity of Naples Federico II, Italy. Since Feb. 2014 he is a PhD candidate in the ACCESS Linnaeus Centre at KTH Royal Institute of Technology. His research interests include hybrid systems, nonlinear control, and coordination in multi-agent systems.

Davide Liuzza received his Ph.D. degree in Au-
tomation Engineering from the University of Naples Federico II, in 2013. He has been visiting Ph.D. student at the Department of Applied Mathematics, University of Bristol (UK) and at the ACCESS Linnaeus Centre, KTH Royal Institute of Technol-
ogy. From 2013 to 2015 he had a Post-doctoral scholarship at the Automatic Control Laboratory in KTH. Currently, he is a postdoc at the department of Engineering, University of Sannio, Benevento, Italy.

Dimos V. Dimarogonas graduated from the School of Electrical and Computer Engineering of the Na-
tional Technical University of Athens (NTUA) in 2001 and was awarded a PhD in Mechanical En-
gineering at NTUA in 2007. He held postdoctoral positions at KTH and MIT between 2007 and 2010. He has joined the Department of Automatic Control at KTH Royal Institute of Technology as an As-
sistant Professor in 2010, and he is currently serving as Professor of Automatic Control in the same department. Currently serving as Associate Editor for Automatica, IET Control Theory & Applications and IEEE Transactions on Automation Science and Engineering.