Cloud-Supported Formation Control of Second-Order Multiagent 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.**

***Index Terms*—Cloud-supported control, second-order consensus, self-triggered control.**

I. Introduction

COORDINATION of networked multiagent 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 of the realistic scenarios, the agents in a multiagent 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, interagent 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 of the practical scenarios. Moreover, GPS is not available underwater, and a vehicle needs to surface whenever it needs to get a position fix [5].

Manuscript received April 6, 2017; revised July 5, 2017; accepted July 12, 2017. Date of publication July 27, 2017; date of current version December 14, 2018. This work was supported by the European Union Horizon 2020 Research and Innovation Program under Grant 644128, AEROWORKS, by the Swedish Foundation for Strategic Research, by the Swedish Research Council, and by the Knut and Alice Wallenberg Foundation. Recommended by Associate Editor D. Nesic. *(Corresponding author: Antonio Adaldo.)*

A. Adaldo, D. V. Dimarogonas, and K. H. Johansson are with the Department of Automatic Control and ACCESS Linnaeus Center, School of Electrical Engineering, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: adaldo@kth.se; dimos@kth.se; kallej@kth.se).

D. Liuzza is with the Department of Engineering, University of Sannio, 82100 Benevento, Italy (e-mail: davide.liuzza@unisannio.it).

Digital Object Identifier 10.1109/TCNS.2017.2732819

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 multiagent systems. Triggered control was introduced to limit the amount of communication within the parts of a feedback control system (plant, sensors, and actuators). In the context of multiagent systems, triggered control is used to limit the communication among different agents. Various flavors of triggered control have been applied to multiagent systems: namely, with *event-triggered control*, interagent 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 interagent communication is interdicted. The use of a shared information repository in multiagent control is subject to recent, but growing, research attention. In [10], the authors employ asynchronous communication with a base station to address a multiagent coverage control problem. In [11], the authors present a cloud-supported approach to multiagent optimization.

In this paper, we present a multiagent control scheme, where interagent communication is completely replaced by the use of a shared information repository hosted on a cloud. Different than 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, despite 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 multiagent systems with second-order dynamics. Moreover, different than [16] and [17], here, we consider additive disturbances (both persistent and vanishing) on the agent dynamics.

With respect to centralized solutions for multiagent 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 fail-safe state in case the communication with the cloud is temporarily lost; the framework can also be 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 are 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 the existing control schemes for asynchronous consensus algorithms with communication delays, for example, [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–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 ith row and the jth column is denoted as $\{M\}_{ij}$, while $\operatorname{eig}(M)$ is the set of the distinct eigenvalues of M.

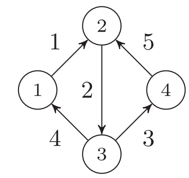


Fig. 1. Graph with four nodes and five edges. The nodes and the edges are labeled with their indexes.

In this paper, a *graph* is defined as a triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where $\mathcal{V} = \{1, \dots, N\}$ with $N \in \mathbb{N}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ with the constraint $(i, i) \notin \mathcal{E}$ for all $i \in \mathcal{V}$, and $w : \mathcal{E} \to \mathbb{R}_{++}$. Each element of $\mathcal{V}$ 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 $e_1, \dots, e_M$, where M is the number of edges in the graph. For each edge e_ι, we denote as $\operatorname{head}(e_\iota)$ and $\operatorname{tail}(e_\iota)$, 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 $\operatorname{tail}(e)$ to $\operatorname{head}(e)$. For example, Fig. 1 illustrates a graph with four vertexes and five edges, each labeled with its index (the weights of the edges are not represented).

For each vertex $i \in \mathcal{V}$, the set $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{V}\}$ is called the *neighborhood* of i, and the vertexes $j \in \mathcal{N}_i$ are called the *neighbors* of i. Moreover, the sets $\mathcal{E}_i^{\text{in}} = \{e \in \mathcal{E} : \operatorname{head}(e) = i\}$ and $\mathcal{E}_i^{\text{out}} = \{e \in \mathcal{E} : \operatorname{tail}(e) = i\}$ 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 $i_1, \dots, i_P$ such that $(i_k, i_{k+1}) \in \mathcal{E}$ for each $k = 1, \dots, P-1$. The *incidence matrix* is defined as $B \in \mathbb{R}^{N\times M}$ such that $\{B\}_{i\iota} = 1$ if $e_\iota \in \mathcal{E}_i^{(\text{in})}$, $\{B\}_{i\iota} = -1$ if $e_\iota \in \mathcal{E}_i^{(\text{out})}$, and $\{B\}_{i\iota} = 0$ otherwise. We also introduce the matrix $C \in \mathbb{R}^{N\times M}$ such that $\{C\}_{i\iota} = w(e_\iota)$ if $e_\iota \in \mathcal{E}_i^{(\text{in})}$ and $\{C\}_{i\iota} = 0$ otherwise. The *Laplacian matrix* is defined as $L = CB^\intercal$. A *spanning tree* is a subset $\mathcal{T} \subseteq \mathcal{E}$ 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 $\mathcal{T}$ and (ii) property (i) does not hold for any proper subset of $\mathcal{T}$. The vertex i_0 is called the *root* of the spanning tree $\mathcal{T}$. 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 $\mathcal{T} = \{e_1, \dots, e_{N-1}\}$, and, following [13], we define $B_\mathcal{T}$ 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 $1, \dots, N$, and we let $\mathcal{V} = \{1, \dots, N\}$. 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

$$\dot{p}_i(t) = v_i(t) \tag{1a}$$

$$\dot{v}_i(t) = u_i(t) + d_i(t) \tag{1b}$$

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

Assumption III.1: The disturbance signals $d_i(t)$ in (1) satisfy $\|d_i(t)\| \leq \delta(t)$, where

$$\delta(t) = (\delta_0 - \delta_\infty)e^{-\lambda_\delta t} + \delta_\infty \tag{2}$$

for some $0 \leq \delta_\infty \leq \delta_0$ and $\lambda_\delta > 0$.

Assumption III.1 allows us 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 $p_i(t) \to \bar{p}(t) + b_i$, where $\bar{p}$ is the average position, and $v_i(t) \to \bar{v}(t)$, 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 $\mathcal{T}$, and let $B_\mathcal{T}$ be the incidence matrix associated with the edge in the tree. Define the *edge states* of the network as $x(t) = (B_\mathcal{T} \otimes I_n)(p(t) - b)$ and $y(t) = (B_\mathcal{T} \otimes I_n)v(t)$, where we have denoted $p(t) = [p_1(t)^\intercal, \ldots, p_N(t)^\intercal]^\intercal$, and similarly for b and $v(t)$. Finally, let $\xi(t) = [x(t)^\intercal, y(t)^\intercal]^\intercal$. We say that the multiagent system (1) achieves *practical consensus* over $\mathcal{G}$ if there exists $\chi \geq 0$ such that

$$\limsup_{t \to \infty} \|\xi(t)\| \leq \chi. \tag{3}$$

In particular, if the system achieves practical consensus with $\chi = 0$, we say that the system achieves *asymptotic consensus*. In the rest of this paper, we take $b_i = 0_n$ 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 occurring 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 of 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.,

$$l_i(t) = \max\{k \in \mathbb{N} : \ t_{i,k} \leq t\}. \tag{4}$$

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

$$u_i(t) = u_{i,k} \quad \forall t \in [t_{i,k}, t_{i,k+1}). \tag{5}$$

The data contained in the cloud at a generic time instant are represented in Table I.

TABLE I
DATA CONTAINED IN THE CLOUD AT A GENERIC TIME INSTANT $t \geq 0$

agent	1	2	...	N
last access	t_{1,l_1}	t_{2,l_2}	...	t_{N,l_N}
position	p_{1,l_1}	p_{2,l_2}	...	p_{N,l_N}
velocity	v_{1,l_1}	v_{2,l_2}	...	v_{N,l_N}
control	u_{1,l_1}	u_{2,l_2}	...	u_{N,l_N}
next access	t_{1,l_1+1}	t_{2,l_2+1}	...	t_{N,l_N+1}

The ith 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.

When an agent accesses the cloud, it uploads data that other agents may download later, when they, in turn, access the cloud. Namely, when agent i accesses the cloud at time $t_{i,k}$, it uploads a packet containing the following information: the current time $t_{i,k}$; the measurements $p_{i,k}$ and $v_{i,k}$; the value $u_{i,k}$ of the control input that is going to be applied in the time interval $[t_{i,k}, t_{i,k+1})$; and the time $t_{i,k+1}$ of the next access. This packet overwrites the packet that was uploaded on the previous access, thus avoiding that the amount of data contained in the cloud grew over time. When agent i accesses the cloud at time $t_{i,k}$, it downloads and stores the latest packet uploaded by each agent $j \in \mathcal{N}_i$. This information, together with the measurements $p_{i,k}$ and $v_{i,k}$, is used by agent i to compute its control input $u_{i,k}$ for the upcoming time interval $[t_{i,k}, t_{i,k+1})$, and to schedule the next cloud access $t_{i,k+1}$. If the cloud is endowed with some computational capabilities, it may also compute some global information about the state of the system for the agents to download. In this case, the cloud provides a positive scalar $\hat{\eta}_{i,k}$, which represents an upper bound on $\|\xi(t)\|$. This estimate is formally defined in Section IV. However, as we shall see, this information is not necessary to the convergence properties of the closed-loop system, but can be introduced purely for performance improvement.

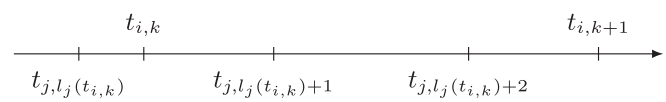


Fig. 2. Excerpt of a possible sequence of cloud accesses on the time line. Recall that $t_{j,l_j(t)}$ denotes the most recent cloud access of agent j with respect to the time t. Note that there can be more than one access of agent j between two consecutive accesses of agent i.

To better illustrate the access sequence and the corresponding notation, Fig. 2 shows a possible sequence of cloud accesses on the time line. Note that in the scenario depicted in Fig. 2, agent j surfaces and changes its control input more than one time within the interval $[t_{i,k}, t_{i,k+1})$. Agent i does not know the control

Algorithm 1: Operations executed by agent i at time $t_{i,k}$.

```
download measurements p_{i,k} and v_{i,k}
for j ∈ N_i do
    download packet {t_{j,l_j}, p_{i,l_j}, v_{i,l_j}, u_{j,l_j}, t_{j,l_j+1}}
end for
download η̂_{i,k}
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}}
```

input that agent j will apply after t_{j,h_j+1}, nor it knows whether agent j will surface more times after t_{j,h_j+1}. Our scheduling algorithm is able to guarantee the overall system's convergence despite these limitations.

The operations that each agent i performs upon each cloud access $t_{i,k}$ are summarized as the following Algorithm 1.

Remark III.1: In most of the existing self-triggered control protocols for multiagent 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 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 $\hat{v}_j^{i,k}(t)$ (respectively, $\hat{p}_j^{i,k}(t)$) be the velocity (respectively, position) of agent j at time t as predicted by agent i upon its kth access to the cloud. These predictions are computed by agent i using the data downloaded from the cloud about agent j as follows:

$$\hat{v}_j^{i,k}(t) = v_{j,l_j(t_{i,k})} + (t - t_{j,l_j(t_{i,k})})u_{j,l_j(t_{i,k})} \quad \text{for } t \in [t_{i,k}, t_{j,l_j(t_{i,k})+1}) \tag{6a}$$

$$\hat{v}_j^{i,k}(t) = \hat{v}_j^{i,k}(t_{j,h_j(t_{i,k})+1}) \text{ for } t > t_{j,l_j(t_{i,k})+1} \tag{6b}$$

$$\hat{p}_j^{i,k}(t) = p_{j,l_j(t_{i,k})} + \int_{t_{j,l_j(t_{i,k})}}^{t} \hat{v}_j^{i,k}(\tau)d\tau. \tag{6c}$$

Note that the predictions (6) are obtained by integrating the agent dynamics (1) in the time interval $[t_{j,l_j(t_{i,k})}, t]$, while neglecting the effect of the disturbances. Finally, the control input $u_{i,k}$ is computed as

$$u_{i,k} = \sum_{j\in\mathcal{N}_i} w_{ij}(k_p(\hat{p}_j^{i,k}(t_{i,k}) - p_{i,k}) + k_v(\hat{v}_j^{i,k}(t_{i,k}) - v_{i,k})) \tag{7}$$

where k_p and k_v are positive gains, $\mathcal{N}_i$ 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 $x(t)$ and $y(t)$ as in Section III-B, with $B_\mathcal{T}$ referred to as the network graph. Using (1), the dynamics of the edge states are described by

$$\dot{x}(t) = y(t) \tag{8a}$$

$$\dot{y}(t) = (B_\mathcal{T}^\intercal \otimes I_n)(u(t) + d(t)) \tag{8b}$$

where we have denoted $u(t) = [u_1(t)^\intercal, \ldots, u_N(t)^\intercal]^\intercal$ 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 $\sigma_{i,k}(t)$ on the difference $\tilde{u}_i(t)$ between the actual control signal $u_{i,k}$ and an ideal diffusive coupling control, defined as

$$z_i(t) = \sum_{j\in\mathcal{N}_i} w_{ij}(k_p(p_j(t) - p_i(t)) + k_v(v_j(t) - v_i(t))). \tag{9}$$

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

$$\varsigma(t) = \varsigma_\infty + (\varsigma_0 - \varsigma_\infty)e^{-\lambda_\varsigma t} \tag{10}$$

with $\lambda_\varsigma > 0$ and $0 \le \varsigma_\infty < \varsigma_0$.

Remark IV.1: Note that the parameters ς_0, ς_∞, and λ_ς represent a tradeoff between convergence performance and the number of control updates: smaller values of ς_∞ lead to a smaller convergence radius, but possibly induce a larger number of control updates; smaller values of ς_0 and larger values of λ_ς lead to a 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 $\sigma_{i,k}(t)$ and the scheduling law. Note that $\sigma_{i,k}(t)$ 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 $t_{j,l_j(t_{i,k})+1}$. Therefore, $\sigma_{i,k}(t)$ is defined by aggregating three functions that capture these effects. To capture the effect of the disturbances, let

$$\Omega_{i,k}(t) = k_v \int_{t_{i,k}}^{t} \delta(\tau)d\tau + k_p \int_{t_{i,k}}^{t}\int_{t_{i,k}}^{\tau} \delta(\theta)d\theta d\tau \tag{11}$$

$$\Psi_{i,k}(t) = \sum_{j\in\mathcal{N}_i} w_{ij}(\Omega_{i,k}(t) + \Omega_{j,l_j(t_{i,k})}(t)). \tag{12}$$

To capture the effect of the sampling, let

$$\Theta_{i,k}(t) = || \sum_{j\in\mathcal{N}_i} w_{ij} \left(k_p \left(\hat{p}_j^{i,k}(t) - \hat{p}_i^{i,k}(t)\right) + k_v \left(\hat{v}_j^{i,k}(t) - \hat{v}_i^{i,k}(t)\right)\right) - u_{i,k}||. \quad (13)$$

To capture the effect of some neighbors' inputs being unknown, we shall construct an upper bound to apply to the unknown control inputs. To this aim, let $R = B_{\mathcal{T}}^{\intercal} C T^{\intercal}$, where L is the Laplacian matrix of the network graph, and T is such that $B = B_{\mathcal{T}} T$. (Note that, under Assumption III.2, such a matrix T exists, since $B_{\mathcal{T}}$ is full rank.) Then, let

$$H = \begin{bmatrix} 0_{(N-1)\times(N-1)} & I_{N-1} \\ -k_p R & -k_v R \end{bmatrix} \quad (14)$$

where k_p and k_v 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 this paper, we shall assume that they are indeed chosen to make H Hurwitz. Let $\lambda = -\max\{\Re(\lambda_H) : \lambda_H \in \operatorname{eig}(H)\}$ and

$$\eta(\eta_0, t_0, t) = e^{-\lambda(t-t_0)}\eta_0 + \sqrt{N}\|B_{\mathcal{T}}\| \int_{t_0}^{t} e^{-\lambda(t-\tau)}(\varsigma(\tau) + \delta(\tau))d\tau. \quad (15)$$

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: let

$$\mu_i(t) = \beta_i \eta(\eta_0, 0, t) + \varsigma(t) \quad (16)$$

where $\eta_0 \geq \|\xi(0)\|$ and $\beta_i = \|C_i T^{\intercal} K\|$, with C_i being the ith row of C, and $K = I_N \otimes [k_p, k_v]$. Note that a suitable η_0 can be computed by knowing only some bounds on the possible initial conditions. Finally, let $\mathcal{N}'_{i,k}(t)$ be the subset of $\mathcal{N}_i$ containing the neighbors of i with unknown control input at time t, namely, $\mathcal{N}'_{i,k}(t) = \{j \in \mathcal{N}_i : t_{j,l_j(t_{i,k})+1} < t\}$. The effect of the unknown control inputs of some neighbors is captured by the function

$$\Phi_{i,k}(t) = \sum_{j\in\mathcal{N}'_{i,k}(t)} w_{ij} \left(\int_{t_{j,l_j(t_{i,k})+1}}^{t} \mu_j(t)d\tau + \int_{t_{j,l_j(t_{i,k})+1}}^{t} \int_{t_{j,l_j(t_{i,k})+1}}^{\tau} \mu_j(\theta)d\theta d\tau\right). \quad (17)$$

We can now define $\sigma_{i,k}(t)$ as

$$\sigma_{i,k}(t) = \Omega_{i,k}(t) + \Theta_{i,k}(t) + \Phi_{i,k}(t) \quad (18)$$

and the scheduling rule is given by

$$t_{i,k+1} = \inf\left\{t > t_{i,k} : \sigma_{i,k}(t) \geq \varsigma(t) \text{ or } \Omega_{i,k}(t) \geq \frac{\alpha}{\nu_i}\varsigma(t)\right\} \quad (19)$$

where $\nu_i = \max_{q:\, i\in\mathcal{N}_q}\{\sum_{s\in\mathcal{N}_q} w_{sq}\}$ and $\alpha \in (0, 1)$.

Remark IV.2: The parameter α 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 α 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 $\sigma_{i,k}(t)$ 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 $\hat{p}(t) = [\hat{p}_1^{1,l_1(t)}(t)^{\intercal}, \ldots, \hat{p}_N^{N,l_N(t)}(t)^{\intercal}]^{\intercal}$ and $\hat{v}(t)$ (defined similarly), and let $\hat{x}(t) = (B_{\mathcal{T}} \otimes I_n)\hat{p}(t)$, $\hat{y}(t) = (B_{\mathcal{T}} \otimes I_n)\hat{v}(t)$, and $\hat{\xi}(t) = [\hat{x}(t)^{\intercal}, \hat{y}(t)^{\intercal}]^{\intercal}$. Moreover, let

$$\Delta_{i,k}(t) = \int_{t_{i,k}}^{t} \delta(\tau)d\tau + \int_{t_{i,k}}^{t}\int_{t_{i,k}}^{\tau} \delta(\theta)d\theta d\tau \quad (20)$$

$$\Delta(t) = [\Delta_{1,l_1(t)}, \ldots, \Delta_{N,l_N(t)}]^{\intercal}. \quad (21)$$

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

$$\mu_i(t) = \beta_i \eta(\hat{\eta}_{i,k}, t_{i,k}, t) + \varsigma(t) \quad (22)$$

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 this 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 $\hat{\eta}_{i,k}$ being computed by the cloud and shared with agent i upon access k is easily captured by preliminarily observing that $\eta(\hat{\eta}_{i,k}, t_{i,k}, t) \leq \eta(\eta_0, 0, t)$.

V. Main Result

Our main result is formalized as the following theorem.

Theorem V.1. Consider the multiagent 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_v be such that H(14) is Hurwitz. If $\varsigma_\infty > 0$, the closed-loop system does not exhibit Zeno behavior and achieves practical consensus over the network graph with

$$\chi = \frac{\sqrt{N}\|B_{\mathcal{T}}\|(\varsigma_\infty + \delta_\infty)}{\lambda}, \quad (23)$$

where ς_∞ is the asymptotic value of the threshold function (10), δ_∞ is the asymptotic value of the disturbance bound (2), λ is defined in Section IV, and $B_{\mathcal{T}}$ is the incidence matrix of the network graph. If $\delta_\infty = 0$, $\varsigma_\infty = 0$ and $\lambda_\varsigma < \min\{\lambda, \lambda_\delta\}$, 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 multiagent system; see, for example, [21]. Here, however, convergence is obtained by using an asynchronously accessed repository, rather than by direct interagent communication. Note also that the convergence error represented by χ is distributed across the whole network, which is reflected in that χ grows with $\sqrt{N}$. Such dependence vanishes if we focus on the mean square error attained by a single agent, which is bounded by $\sqrt{\chi^2/N}$. Finally, note that the edge weights w_{ij} influence the convergence radius χ through the parameter λ.

The proof of Theorem V.1 is given in the following three sections of this 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

$$z_i(t) = -(L_i^\mathsf{T} \otimes I_n)(k_p p(t) + k_v v(t)) \tag{24}$$

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

$$z(t) = -(L \otimes I_n)(k_p p(t) + k_v v(t)). \tag{25}$$

Now recall that $L = CB^\mathsf{T}$, and that, since $B_\mathcal{T}$ is full column rank, there exists a matrix T such that $B = B_\mathcal{T} T$ (namely, $T = B_\mathcal{T}^\mathsf{T}(B_\mathcal{T} B_\mathcal{T}^\mathsf{T})^{-1} B$). Therefore

$$z(t) = -(CT^\mathsf{T} B_\mathcal{T}^\mathsf{T} \otimes I_n)(k_p p(t) + k_v v(t)). \tag{26}$$

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

$$z(t) = -(CT^\mathsf{T} \otimes I_n)(k_p x(t) + k_v y(t)). \tag{27}$$

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

$$\tilde{u}_i(t) = u_i(t) - z_i(t). \tag{28}$$

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

$$\tilde{u}(t) = u(t) - z(t). \tag{29}$$

From (29) and (26), we have

$$u(t) = (CT^\mathsf{T} \otimes I_n)(k_p x(t) + k_v y(t)) + \tilde{u}(t) \tag{30}$$

which substituted in (8) yields

$$\dot{x}(t) = y(t) \tag{31a}$$

$$\dot{y}(t) = -(B_\mathcal{T}^\mathsf{T} \otimes I_n)(CT^\mathsf{T} \otimes I_n)(k_p x(t) + k_v y(t)) + (B_\mathcal{T}^\mathsf{T} \otimes I_n)(\tilde{u}(t) + d(t)). \tag{31b}$$

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

$$\dot{x}(t) = y(t) \tag{32a}$$

$$\dot{y}(t) = -(R \otimes I_n)(k_p x(t) + k_v y(t)) + (B_\mathcal{T}^\mathsf{T} \otimes I_n)(\tilde{u}(t) + d(t)). \tag{32b}$$

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

$$z(t) = -((CT^\mathsf{T} K) \otimes I_n)\xi(t) \tag{33}$$

$$\dot{\xi}(t) = (H \otimes I_n)\xi(t) + (G \otimes I_n)(\tilde{u}(t) + d(t)) \tag{34}$$

where H is the Hurwitz matrix defined in (14), $G = [0^\mathsf{T}_{(N-1)\times N}, \; B_\mathcal{T}]^\mathsf{T}$, and $K = I_N \otimes [k_p, k_v]$.

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

Lemma VI.1: Consider the multiagent system (1), and let Assumptions III.1 and III.2 hold. Suppose that

$$\|\tilde{u}_i(\tau)\| \leq \varsigma(\tau) \tag{35}$$

for all $\tau \in [0, t)$ and all $i \in \mathcal{V}$, where $\tilde{u}_i(\cdot)$ is defined by (28) and $\varsigma(\cdot)$ is the threshold function (10). Then, for all $\tau \in [0, t)$, we have $\|\xi(\tau)\| \leq \eta(\eta_0, 0, \tau)$, where $\eta(\cdot,\cdot,\cdot)$ 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

$$\xi(\tau) = e^{F_{e,r}\tau}\xi(t_0) + \int_0^\tau e^{F_{e,r}(\tau-\theta)}(G \otimes I_n) \times (\tilde{u}(\theta) + d(\theta))d\theta. \tag{36}$$

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 $\|e^{F_{e,r}(\tau-t_0)}\| \leq e^{-\lambda(\tau-t_0)}$, and that $\|G\| = \|B_\mathcal{T}\|$, we have $\|\xi(t)\| \leq \eta(\eta_0, 0, t)$. 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 jth agent in (33), we have $z_j(\tau) = ((C_j T^\mathsf{T} K) \otimes I_n)\xi(\tau)$, where C_j denotes the jth row of C. Taking norms of both sides, and substituting the result in the previous inequality, we have $\|u_j(\tau)\| \leq \beta_j\|\xi(\tau)\| + \|\tilde{u}_j(\tau)\|$. The proof is concluded by using $\|\xi(\tau)\| \leq \eta(\eta_0, 0, \tau)$ and (35), to obtain $\|u_j(\tau)\| \leq \mu_j(\tau)$. ■

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

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

Proof: Since (19) guarantees $\sigma_{i,l_i(t)}(t) \leq \varsigma(t)$ for all $t \geq 0$ and all $i \in \mathcal{V}$, we only need to show that $\|\tilde{u}_i(t)\| \leq \sigma_{i,l_i(t)}(t)$ for all $t \geq 0$ 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

$$\tilde{u}_i(t) = u_{i,k} - \sum_{j \in \mathcal{N}_i} w_{ij}(k_p(p_j(t) - p_i(t)) + k_v(v_j(t) - v_i(t))). \tag{37}$$

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

$$v_i(t) = \hat{v}_i^{i,k}(t) + \int_{t_{i,k}}^t d_i(t)d\tau. \tag{38}$$

Second, consider the term $p_i(t)$. Integrating (1a) in $[t_{i,k}, t)$, we have

$$p_i(t) - p_{i,k} = \int_{t_{i,k}}^t v_i(\tau)d\tau \tag{39}$$

which using (38) and (6c) can be rewritten as

$$p_i(t) = \hat{p}_i^{i,k}(t) + \int_{t_{i,k}}^t \int_{t_{i,k}}^\tau d_i(\theta)d\theta d\tau. \tag{40}$$

Third, consider the term $v_j(t)$. Without loss of generality, let $l_j(t_{i,k}) = h$. Integrating (1b) for agent j in $[t_{j,h}, t)$, we have

$$v_j(t) = v_{j,h} + \int_{t_{j,h}}^t u_j(\tau)d\tau + \int_{t_{j,h}}^t d_j(\tau)d\tau. \tag{41}$$

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 $u_j(\tau) = u_{j,l_j(t_{i,k})}$ for all $\tau \in [t_{i,k}, t)$, and therefore, (41) becomes, also using (6a),

$$v_j(t) = \hat{v}_j^{i,k}(t) + \int_{t_{j,h}}^t d_j(\tau)d\tau. \tag{42}$$

In the second case, we have $u_j(\tau) = u_{j,l_j(t_{i,k})}$ for $\tau \in [t_{i,k}, t_{j,l_j(t_{i,k}+1)})$, so we can rewrite (41) as

$$v_j(t) = \hat{v}_j^{i,k}(t) + \int_{t_{j,h+1}}^t u_j(\tau)d\tau + \int_{t_{j,h}}^t d_j(\tau)d\tau \tag{43}$$

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

$$\begin{aligned} p_j(t) &= p_{j,h} + (t - t_{j,h})v_{j,h} \\ &\quad + \int_{t_{j,h}}^t \int_{t_{j,h}}^\tau (u_j(\theta) + d_j(\theta))d\theta d\tau. \end{aligned} \tag{44}$$

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 $\theta \in [t_{j,h}, \tau)$ and all $\tau \in [t_{i,k}, t)$; therefore, (44) becomes, using (6c),

$$p_j(t) = \hat{p}_j^{i,k}(t) + \int_{t_{j,h}}^t \int_{t_{j,h}}^\tau d_j(\theta)d\theta d\tau. \tag{45}$$

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 $\hat{p}_j^{i,k}(t)$, namely, a few passages show that in this case, using (6c), (44) becomes

$$\begin{aligned} p_j(t) &= \hat{p}_j^{i,k}(t) + \int_{t_{j,h+1}}^t \int_{t_{j,h+1}}^\tau u_j(\theta)d\theta d\tau \\ &\quad + \int_{t_{j,h}}^t \int_{t_{j,h}}^\tau d_j(\theta)d\theta d\tau. \end{aligned} \tag{46}$$

Using (38), (40), (42), (43), (45), and (46) in (37), we have

$$\begin{aligned} \tilde{u}_i(t) &= u_{i,k} - \sum_{j \in \mathcal{N}_i} w_{ij}\Bigg(k_p\Bigg(\hat{p}_j^{i,k}(t) + \int_{t_{j,h}}^t \int_{t_{j,h}}^\tau d_j(\theta)d\theta d\tau \\ &\quad - \hat{p}_i^{i,k}(t) - \int_{t_{i,k}}^t \int_{t_{i,k}}^\tau d_i(\theta)d\theta d\tau\Bigg) + k_v\Bigg(\hat{v}_j^{i,k}(t) \\ &\quad + \int_{t_{j,h}}^t d_j(\tau)d\tau - \hat{v}_i^{i,k}(t) - \int_{t_{i,k}}^t d_i(\tau)d\tau\Bigg)\Bigg) \\ &\quad - \sum_{j \in \mathcal{N}'_{i,k}(t)} w_{ij}\Bigg(k_v \int_{t_{j,h+1}}^t u_j(\tau)d\tau + k_p \int_{t_{j,h+1}}^t \\ &\quad \times \int_{t_{j,h+1}}^\tau u_j(\theta)d\theta d\tau\Bigg). \end{aligned} \tag{47}$$

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

$$\|\tilde{u}_i(t)\| \leq \Theta_{i,k}(t) + \Psi_{i,k}(t) + \Phi_{i,k}(t) = \sigma_{i,k}(t) \tag{48}$$

where $\Theta_{i,k}(\cdot)$, $\Psi_{i,k}(\cdot)$, $\Phi_{i,k}(t)$, 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 ς_∞ is the asymptotic value of the threshold function (10).

Lemma VII.1: Consider the multiagent 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 $\Omega_{i,k}(t) \geq (\alpha/\nu_i)\varsigma(t)$. We can use (2) and (11) to compute $\Omega_{i,k}(t)$ explicitly as

$$\Omega_{i,k}(t) = \frac{\delta_0 - \delta_\infty}{\lambda_\delta} e^{-\lambda_\delta t_{i,k}} \left(k_v \left(1 - e^{-\lambda_\delta (t - t_{i,k})}\right) + k_p \left((t - t_{i,k}) + \frac{1 - e^{-\lambda_\delta (t-t_{i,k})}}{\lambda_\delta} \right)\right) + \delta_\infty (t - t_{i,k})(k_v + 0.5 k_p (t - t_{i,k})). \quad (49)$$

Since $\varsigma(t)$ can be lower-bounded as $\varsigma(t) \geq \varsigma_\infty$, a necessary condition to have $\Omega_{i,k}(t) \geq (\alpha/\nu_i)\varsigma(t)$ is that the right-hand side of (49) is larger than ς_∞. This condition implies that $t \geq t_{i,k} + \tau_i^*$, where τ_i^* is the smallest (strictly) positive solution of

$$\frac{\delta_0 - \delta_\infty}{\lambda_\delta} \left(k_v (1 - e^{-\lambda_\delta \tau}) + k_p \left(\tau + \frac{1 - e^{-\lambda_\delta \tau}}{\lambda_\delta}\right)\right) + \delta_\infty \tau (k_v + 0.5 k_p \tau) = (\alpha/\nu_i)\varsigma_\infty. \quad (50)$$

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 $[t_{j,h}, t_{i,k})$ and $[t_{i,k}, t)$, we have

$$\Omega_{j,h}(t) = \Omega_{j,h}(t_{i,k}) + k_v \int_{t_{i,k}}^{t} \delta(\tau) d\tau + k_p \int_{t_{i,k}}^{t} \int_{t_{j,h}}^{\tau} \delta(\theta) d\theta d\tau. \quad (51)$$

Splitting the integration interval $[t_{j,h}, \tau)$ further in $[t_{j,h}, t_{i,k})$ and $[t_{i,k}, t)$, we have

$$\Omega_{j,h}(t) = \Omega_{j,h}(t_{i,k}) + \Omega_{i,k}(t) + k_p \int_{t_{i,k}}^{t} \int_{t_{j,h}}^{t_{i,k}} \delta(\theta) d\theta d\tau. \quad (52)$$

Observing that $\delta(\theta) \leq \delta_0$, and that $t_{i,k} - t_{j,h} \leq t_{j,h+1} - t_{j,h}$, we have $\int_{t_{j,h}}^{t_{i,k}} \delta(\theta) d\theta \leq \delta_0 \tau_j^*$, where τ_j^* denotes the smallest (strictly) positive solution of (50), evaluated for agent j. Hence, we can upper-bound (52) as

$$\Omega_{j,h}(t) \leq \Omega_{j,h}(t_{i,k}) + \Omega_{i,k}(t) + k_p \delta_0 \tau_j^* (t - t_{i,k}). \quad (53)$$

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

$$\Omega_{j,h}(t) \leq (\alpha/\nu_j)\varsigma(t_{i,k}) + \Omega_{i,k}(t) + k_p \tau_j^* (t - t_{i,k}). \quad (54)$$

Using (54) in (20), we can upper-bound $\Psi_{i,k}(t)$ as

$$\Psi_{i,k}(t) \leq \sum_{j \in \mathcal{N}_i} w_{ij} \left(\frac{\alpha}{\nu_j} \varsigma(t_{i,k}) + \Omega_{i,k}(t) + k_p \delta_0 \tau_j^* (t - t_{i,k})\right). \quad (55)$$

Since, by the definition of ν_j, $\sum_{j \in \mathcal{N}_i} (w_{ij}/\nu_j) \leq 1$, we can further bound (55) as

$$\Psi_{i,k}(t) \leq \alpha \varsigma(t_{i,k}) + \left(\sum_{j \in \mathcal{N}_i} w_{ij} \right) \Omega_{i,k}(t) + k_p \delta_0 \left(\sum_{j \in \mathcal{N}_i} w_{ij} \tau_j^* \right) (t - t_{i,k}). \quad (56)$$

Next, consider the term $\Phi_{i,k}(t)$. Since $\eta(\eta_0, t_0, t)$ is an upper-bounded function of t, we can denote as $\bar{\eta}$ largest value of $\eta(\eta_0, t_0, t)$ for $t \geq 0$, which, by observing also that $\varsigma(t) \leq \varsigma_0$, allows us to bound $\mu_j(t)$ as $\mu_j(t) \leq \beta_j \bar{\eta} + \varsigma_0$. Consequently, from (17), we have

$$\Phi_{i,k}(t) \leq \sum_{j \in \mathcal{N}'_{ik}(t)} w_{ij} (\beta_j \bar{\eta} + \varsigma_0)((t - t_{j,h+1}) + 0.5(t - t_{j,h+1})^2). \quad (57)$$

Last, consider the term $\Theta_{i,k}(t)$, and note that, using (7), it can be written as

$$\Theta_{i,k}(t) = \left\| \sum_{j \in \mathcal{N}_i} w_{i,j} (k_p (\hat{p}_j^{i,k}(t) - \hat{p}_j^{i,k}(t_{i,k}) - \hat{p}_i^{i,k}(t) + p_{i,k}) + k_v (\hat{v}_j^{i,k}(t) - \hat{v}_j^{i,k}(t_{i,k}) - \hat{v}_i^{i,k}(t) + v_{i,k})) \right\|. \quad (58)$$

Using (6), the right-hand side of (58) can be rewritten as

$$\Theta_{i,k}(t) = \left\| \sum_{j \in \mathcal{N}_i} w_{ij} \left(k_v \left(\int_{t_{i,k}}^{t'_{j,h}} u_{j,h} d\tau - \int_{t_{i,k}}^{t} u_{i,k} d\tau \right) + k_p \left(\int_{t_{i,k}}^{t'_{j,h}} \int_{t_{i,k}}^{\tau} u_{j,h} d\theta d\tau - \int_{t_{i,k}}^{t} \int_{t_{i,k}}^{\tau} u_{i,k} d\theta d\tau \right)\right) \right\| \quad (59)$$

where we have denoted $t'_{j,h} = \min\{t, t_{j,h+1}\}$ for brevity. Since each control input is upper-bounded as $\|u_i(t)\| \leq \mu_i(t) \leq \beta_i \bar{\eta} + \varsigma_0$, using the triangular inequality on the right-hand side of (59), then summing side by side with (57), yields

$$\Theta_{i,k}(t) + \Phi_{i,k}(t) \leq \sum_{j \in \mathcal{N}_i} w_{ij} ((\beta_j + \beta_i)\bar{\eta} + 2\varsigma_0) \cdot (k_v (t - t_{i,k}) + 0.5 k_p (t - t_{i,k})^2). \quad (60)$$

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 $1 - e^{\lambda_\delta (t - t_{i,k})}$, plus the term $\alpha \varsigma(t_{i,k})$, so that we can write

$$\sigma_{i,k}(t) \leq \sigma_1 (t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 (1 - e^{-\lambda_\delta (t - t_{i,k})}) + \alpha \varsigma(t_{i,k}) \quad (61)$$

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

$$\sigma_1 (t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 (1 - e^{-\lambda_\delta (t - t_{i,k})}) + \alpha \varsigma(t_{i,k}) \geq \varsigma(t). \quad (62)$$

Since $\varsigma(t) \geq \varsigma(t_{i,k}) - \lambda_\varsigma (\varsigma_0 - \varsigma_\infty)(t - t_{i,k})$, (62) implies

$$(\sigma_1 + \lambda_\varsigma (\varsigma_0 - \varsigma_\infty))(t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 (1 - e^{-\lambda_\delta (t - t_{i,k})}) \geq (1 - \alpha)\varsigma(t_{i,k}). \quad (63)$$

Finally, observing that $\varsigma(t_{i,k}) \geq \varsigma_\infty$, (63) implies $t_{i,k+1} \geq t_{i,k} + \tau_i^{**}$, where τ_i^{**} is the smallest (strictly) positive solution of $(\sigma_1 + \lambda_\varsigma (\varsigma_0 - \varsigma_\infty))\tau + \sigma_2 \tau^2 + \sigma_3 (1 - e^{-\lambda_\delta \tau}) \geq (1 - \alpha)\varsigma_\infty$. Since $t_{i,k+1}$ is defined as the smallest time when either $\Omega_{i,k}(t) \geq \alpha \varsigma(t)$ 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. ■

Lemma VII.2: Consider the multiagent system (1), with control law (5)–(7), and cloud accesses scheduled by (19). Let Assumptions III.2 and III.1 hold, with $\delta_\infty = 0$ in Assumption III.1. Choose k_p and k_v such that H is Hurwitz, and choose $\varsigma_\infty = 0$ and $\lambda_\varsigma < \min\{\lambda, \lambda_\delta\}$. Then, the closed-loop system does not exhibit Zeno behavior.

Proof: Using $\delta_\infty = 0$ and $\lambda_\delta \geq \lambda_\varsigma$ in (11), we obtain

$$\Omega_{i,k}(t) \leq \frac{\delta_0}{\lambda_\varsigma} e^{-\lambda_\varsigma t_{i,k}} \left(k_v (1 - e^{-\lambda_\varsigma (t - t_{i,k})}) + k_p \left((t - t_{i,k}) + \frac{1 - e^{-\lambda_\varsigma (t - t_{i,k})}}{\lambda_\varsigma} \right) \right). \quad (64)$$

Moreover, observe that, with $\varsigma_\infty = 0$, the threshold function can be written as

$$\varsigma(t) = \varsigma_0 e^{-\lambda_\varsigma t} = \varsigma_0 e^{-\lambda_\varsigma t_{i,k}} e^{-\lambda_\varsigma (t - t_{i,k})}. \quad (65)$$

Inequalities (64) and (65) show that the triggering condition $\Omega_{i,k}(t) \geq (\alpha/\nu_i)\varsigma(t)$ implies that $t \geq t_{i,k} + \tau_i^*$, where τ_i^* is the smallest (strictly) positive solution of

$$\frac{\delta_0}{\lambda_\varsigma} \left(\left(k_v + \frac{k_p}{\lambda_\varsigma} \right) (1 - e^{-\lambda_\varsigma \tau}) + k_p \tau \right) \geq \frac{\varsigma_0}{\nu_i} e^{-\lambda_\varsigma \tau}. \quad (66)$$

Reasoning as in Lemma VII.1, we find again that (52) holds, but since by hypothesis $\delta(\theta) = \delta_0 e^{-\lambda_\delta \theta} \leq \delta_0 e^{-\lambda_\varsigma \theta}$, we compute therein

$$\int_{t_{i,k}}^{t} \int_{t_{j,h}}^{t_{i,k}} \delta(\theta) d\theta d\tau \leq \frac{\delta_0}{\lambda_\varsigma} (e^{\lambda_\varsigma (t_{i,k} - t_{j,h})} - 1) e^{-\lambda_\varsigma t_{i,k}} (t - t_{i,k}). \quad (67)$$

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 $(\delta_0/\lambda_\varsigma)(e^{\lambda_\varsigma \tau_j^*} - 1) e^{-\lambda_\varsigma t_{i,k}} (t - t_{i,k})$, which substituted in (52) yields

$$\Omega_{j,h}(t) \leq \Omega_{j,h}(t_{i,k}) + \Omega_{i,k}(t) + k_p (\delta_0/\lambda_\varsigma)(e^{\lambda_\varsigma \tau_j^*} - 1) e^{-\lambda_\varsigma t_{i,k}} (t - t_{i,k}). \quad (68)$$

Similarly, as done in Lemma VII.1, we can now use (68) in (20), obtaining

$$\Psi_{i,k}(t) \leq \alpha\varsigma(t_{i,k}) + \left(\sum_{j \in \mathcal{N}_i} w_{ij} \right) t \Omega_{i,k}(t) + \frac{k_p \delta_0 e^{-\lambda_\varsigma t_{i,k}}}{\lambda_\delta} \left(\sum_{j \in \mathcal{N}_i} w_{ij} \tau_j^* \right) (t - t_{i,k}). \quad (69)$$

Now, note that, using (15) with $\delta_\infty = 0$ and $\varsigma_\infty = 0$, $\eta(t)$ can be upper-bounded by the slowest exponential among $e^{-\lambda t}$, $e^{-\lambda_\varsigma t}$, and $e^{-\lambda_\delta t}$. Since by hypothesis $\lambda_\varsigma \leq \min\{\lambda, \lambda_\delta\}$, we can write $\eta(t) \leq \bar{\eta} e^{-\lambda_\varsigma t}$, where $\bar{\eta} > 0$ depends on the initial conditions. Consequently, we can upper-bound the control inputs as

$$\|u_i(t)\| \leq \beta_i \eta(t) + \varsigma(t) \leq (\beta_i \bar{\eta} + \varsigma_0) e^{-\lambda_\varsigma t_{i,k}}. \quad (70)$$

Using (70) in (17), we have

$$\Phi_{i,k}(t) \leq e^{-\lambda_\varsigma t_{i,k}} \sum_{j \in \mathcal{N}'_{ik}(t)} w_{ij} (\beta_j \bar{\eta} + \varsigma_0) \cdot \left((t - t_{j,h+1}) + 0.5 (t - t_{j,h+1})^2 \right). \quad (71)$$

Similarly, as done in Lemma VII.1, we can now sum (59) and (71) side by side and, then, use the triangular inequality and (71) to obtain

$$\Theta_{i,k}(t) + \Phi_{i,k}(t) \leq e^{-\lambda_\varsigma t_{i,k}} \sum_{j \in \mathcal{N}_i} w_{ij} ((\beta_j + \beta_i)\bar{\eta} + 2\varsigma_0) \cdot \left(\left(k_v + \frac{k_p}{\lambda_\varsigma} \right) (1 - e^{-\lambda_\varsigma (t - t_{i,k})}) + k_p (t - t_{i,k}) \right). \quad (72)$$

By summing the right-hand sides of (69) and (72), considering also (49), and observing that $\varsigma(t_{i,k}) = \varsigma_0 e^{-\lambda_\varsigma t_{i,k}}$, we have an upper bound for $\sigma_{i,k}(t)$ in the form

$$\sigma_{i,k}(t) \leq (\sigma_1 (t - t_{i,k}) + \sigma_2 (t - t_{i,k})^2 + \sigma_3 (1 - e^{-\lambda_\varsigma (t - t_{i,k})})) e^{-\lambda_\varsigma t_{i,k}} + \alpha\varsigma(t_{i,k}). \quad (73)$$

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 τ_i^{**} is the smallest (strictly) positive solution of $\sigma_1 \tau + \sigma_2 \tau^2 + \sigma_3 (1 - e^{-\lambda_\varsigma \tau}) + \alpha\varsigma_0 = \varsigma_0 e^{-\lambda_\varsigma \tau}$. Since $t_{i,k+1}$ is defined as the smallest time when either $\Omega_{i,k}(t) \geq \alpha\varsigma(t)$ 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. ■

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 χ given by (23). If $\delta_\infty = 0$, $\varsigma_\infty = 0$, and $\lambda_\varsigma < \min\{\lambda, \lambda_\delta\}$, 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 the other for a scenario where asymptotic convergence is reached. For both simulations, we consider a multiagent system made up of $N = 4$ agents with state in $\mathbb{R}^2$, which exchange information through a cloud repository according to the graph $\mathcal{G}$ illustrated in Fig. 1, where all the edges are assigned unitary weights. The assigned graph contains a spanning tree $\mathcal{T}$ made up of the first three edges. The

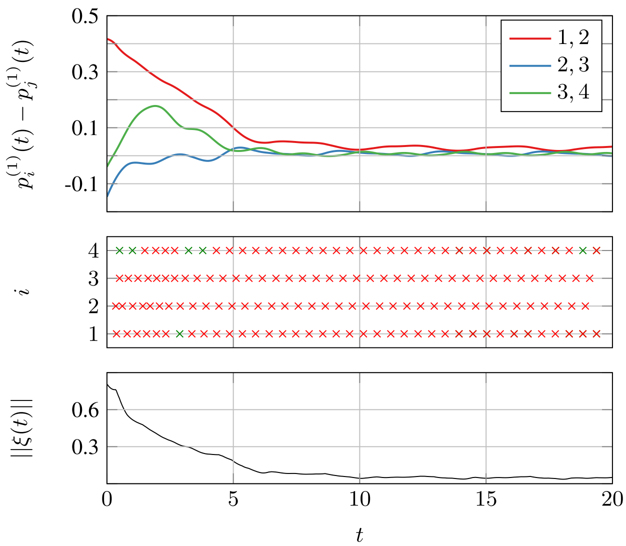


Fig. 3. Simulation with persistent disturbances. (Top) Position mismatches across the edges (j, i) in the spanning tree over time. (Middle) Time instants when each agent accesses the cloud; a green cross denotes an access triggered by $\Omega_{i,k}(t) \geq \alpha/\nu_i\varsigma(t)$; a red cross denotes an access triggered by $\sigma_{i,k}(t) \geq \varsigma(t)$. (Bottom) Norm of the global disagreement vector $\xi(t)$ over time.

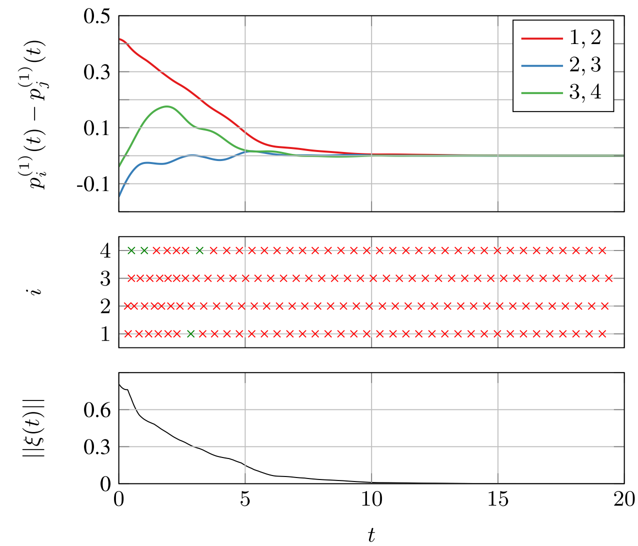


Fig. 4. Simulation with asymptotically vanishing disturbances. Same subplots as in Fig. 3.

corresponding matrix R is

$$R = \begin{bmatrix} 2 & 0 & -1 \\ -1 & 2 & 1 \\ 0 & -1 & 1 \end{bmatrix}. \tag{74}$$

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 $\|B_{\mathcal{T}}\| \simeq 2.45$. The disturbances are chosen as

$$d_i(t) = \delta(t) \begin{bmatrix} \cos(2\pi(i/N)t + 2\pi((N-i)/N)) \\ \sin(2\pi(i/N)t + 2\pi((N-i)/N)) \end{bmatrix} \tag{75}$$

where $\delta(t)$ is defined by Assumption III.1 with $\delta_0 = 0.2$, $\lambda_\delta = 0.45$, $\delta_\infty = 0.02$ 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 α 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 Fig. 3. From Fig. 3, it looks clear that the multiagent system only achieves practical convergence, but the norm of the disagreement vector is significantly reduced. From Fig. 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 Fig. 4. From Fig. 4, it looks clear that $\xi(t) \to 0$, which means that asymptotic convergence is reached. From Fig. 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.

X. Conclusion and Future Developments

This paper has addressed a self-triggered control problem for multiagent coordination of a team of agents with second-order dynamics. Coordination has been achieved by having the agents asynchronously deposit and retrieve data on a cloud repository, rather than by interagent communication. Two control objectives have been considered, namely practical and asymptotic convergence. It has been shown that the proposed control strategy achieves practical convergence in the presence of unknown bounded persistent disturbances and asymptotic convergence in the presence of unknown disturbances if they slowly vanish. Well-posedness of the closed-loop system has been 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 multiagent 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

$$\mathcal{P}(\lambda) = \det(\lambda I_{2N} - F) = \det(\lambda^2 I_N + (\lambda k_v + k_p)L) \tag{76}$$

which for $\lambda \neq 0$ can be written as

$$\mathcal{P}(\lambda) = \lambda^{2N} \det(I_N + (\lambda k_v + k_p)/\lambda^2 L). \tag{77}$$

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 $L_e = B^\intercal C$ is the edge Laplacian of $\mathcal{G}$. Similarly, as done for F, we can compute the characteristic polynomial of F_e as

$$\mathcal{P}_e(\lambda) = \det(\lambda^2 I_M + (\lambda k_v + k_p)L_e) \tag{78}$$

which for $\lambda \neq 0$ can be rewritten as

$$\mathcal{P}_e(\lambda) = \lambda^{2M} \det(I_M + (\lambda k_v + k_p)/\lambda^2 L_e). \tag{79}$$

Since $L = CB^\intercal$ and $L_e = B^\intercal C$, 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

$$S = \begin{bmatrix} I_{N-1} & 0_{(N-1)\times(M-N+1)} \\ -T^\intercal & I_{M-N+1} \end{bmatrix} \tag{80}$$

and note that

$$SL_eS^{-1} = \begin{bmatrix} R & * \\ 0_{(M-N+1)\times(N-1)} & 0_{(M-N+1)} \end{bmatrix}. \tag{81}$$

Multiplying the right-hand side of (78) by $\det(S)\det(S^{-1}) = 1$, and using (81), we have

$$\begin{aligned} \mathcal{P}_e(\lambda) &= \det(S(\lambda^2 I_M + (\lambda k_v + k_p)L_e)S^{-1}) \\ &= \lambda^{2(M-(N-1))} \det(\lambda^2 I_{N-1} + (\lambda k_v + k_p)R) \qquad (82) \\ &= \lambda^{2(M-(N-1))} \mathcal{P}_{e,r}(\lambda) \end{aligned}$$

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

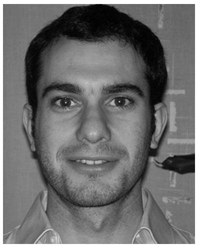
References

[1] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, no. 473, pp. 167–73, 2011.
[2] Y. Shoham and L.-B. Kevin, *Multiagent Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[4] P. V. Teixeira, D. V. Dimarogonas, and K. H. Johansson, "Multi-agent coordination with event-based communication," in *Proc. Amer. Control Conf.*, 2011, pp. 824–829.
[5] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.
[6] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030–2042, Sep. 2010.
[7] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st Conf. Decision Control*, 2012, pp. 3270–3285.
[8] D. V. Dimarogonas and K. H. Johansson, "Event-triggered control for multi-agent systems," in *Proc. IEEE 48th Conf. Decision Control*, 2009, pp. 7131–7136.
[9] C. De Persis and P. Frasca, "Robust self-triggered coordination with ternary controllers," *IEEE Trans. Autom. Control*, vol. 58, no. 12, pp. 3024–3038, Dec. 2013.
[10] R. Patel, J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 24–33, Mar. 2016.
[11] M. T. Hale and M. Egerstedt, "Differentially private cloud-based multi-agent optimization with constraints," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2015, pp. 1235–1240.
[12] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 544–555, Mar. 2011.
[13] Z. Zeng, X. Wang, and Z. Zheng, "Convergence analysis using the edge Laplacian: Robust consensus of nonlinear multi-agent systems via ISS method," *Int. J. Robust Nonlinear Control*, vol. 26, pp. 1051–1072, 2015.
[14] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Control of multi-agent systems with event-triggered cloud access," in *Proc. Eur. Control Conf.*, Linz, Austria, 2015, pp. 954–961.
[15] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Multi-agent trajectory tracking with event-triggered cloud access," in *Proc. IEEE 55th Conf. Decision Control*, 2016, pp. 2207–2214.
[16] C. Nowzari and G. J. Pappas, "Multi-agent coordination with asynchronous cloud access," in *Proc. Amer. Control Conf.*, 2016, pp. 4649–4654.
[17] S. L. Bowman, C. Nowzari, and G. J. Pappas, "Coordination of multi-agent systems via asynchronous cloud communication," in *Proc. IEEE 55th Conf. Decision Control*, 2016, pp. 2215–2220.
[18] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1804–1816, Sep. 2008.
[19] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[20] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[21] A. Adaldo *et al.*, "Event-triggered pinning control of switching networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 204–213, Jun. 2015.
[22] K. H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry, "On the regularization of Zeno hybrid automata," *Syst. Control Lett.*, vol. 38, pp. 141–150, 1999.
[23] K. Liu, G. Xie, and L. Wang, "Consensus for second-order multi-agent systems with inherent nonlinear dynamics under directed topologies," *Syst. Control Lett.*, vol. 62, 2013.
[24] J. R. Silvester, "Determinants of block matrices," *Math. Gazette*, vol. 84, no. 501, pp. 460–467, 2000.
[25] C. Pozrikidis, *An Introduction to Grids, Graphs, and Networks*. Oxford, U.K.: Oxford Univ. Press, 2014.

[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].

Antonio Adaldo received the M.Sc. degree in automation engineering from the University of Naples Federico II, Naples, Italy, in 2013 and is currently pursuing the Ph.D. degree at the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden.

His research interests include hybrid systems, nonlinear control, and coordination in multiagent systems.



Davide Liuzza received the Ph.D. degree in automation engineering from the University of Naples Federico II, Naples, Italy, in 2013.

He was a visiting Ph.D. student with the Department of Applied Mathematics, University of Bristol, Bristol, U.K., and with the ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden. From 2013 to 2015, he had a Postdoctoral scholarship with the Automatic Control Laboratory, KTH Royal Institute of Technology. He is currently a Postdoctoral Researcher with the Department of Engineering, University of Sannio, Benevento, Italy.

Dimos V. Dimarogonas (M'10) received the Graduate degree in electrical and computer engineering and the Ph.D. degree in mechanical engineering from the National Technical University of Athens, Athens, Greece, in 2001 and 2007, respectively.

He held postdoctoral positions at KTH Royal Institute of Technology, Stockholm, Sweden, and Massachusetts Institute of Technology, Cambridge, MA, USA, between 2007 and 2010. In 2010, he joined the Department of Automatic Control, KTH Royal Institute of Technology, Stockholm, Sweden, as an Assistant Professor, and is currently a Professor of automatic control.

Dr. Dimarogonas is an Associate Editor for *Automatica*, *IET Control Theory and Applications*, and IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

Karl H. Johansson (F'13) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Lund, Sweden.

He is the Director of the Stockholm Strategic Research Area ICT—The Next Generation and a Professor with the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include networked control systems, cyberphysical systems, and applications in transportation, energy, and automation.

Dr. Johansson 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 a member of the Royal Swedish Academy of Engineering Sciences and a Distinguished Lecturer of the IEEE.