

# Control of Multi-Agent Systems with Event-Triggered Cloud Access

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

**Abstract**—This paper investigates a multi-agent formation control problem with event-triggered control updates and additive disturbances. The agents communicate only by exchanging information in a cloud repository. The communication with the cloud is considered a shared and limited resource, and therefore it is used intermittently and asynchronously by the agents. The proposed approach takes advantage of having a shared asynchronous cloud support while guaranteeing a reduced number of communication. More in detail, each agent schedules its own sequence of cloud accesses in order to achieve a coordinated network goal. A control law is given with a criterion for scheduling the control updates recursively. The closed loop scheme is proven to be effective in achieving the control objective and a numerical simulation corroborates the theoretical results.

## I. INTRODUCTION

The study of networked control systems (NCS) is motivated by the fact that, nowadays, heterogeneous and geographically distributed devices can be connected with cheap and reliable wireless technologies. Specifically, consensus algorithms have been investigated [1], [2] and tailored for platooning and formation control [3], [4]. On the other hand, several recent papers consider the possibility of distributed wireless sensors and actuators in NCS, devoting the research effort in coordinating the data packets and guaranteeing desired performances [5], [6]. Motivated by the need of saving hardware and software resources and reducing the transmitted data, event-triggered and self-triggered control strategies have been introduced [7]–[9], and later extended to multi-agent coordination [10]–[12]. These strategies do not require a fixed sampling period for the feedback loop, but the control input is updated only when a specific condition related to the stability or to some control performance is violated.

In the current paper a novel event-scheduled cloud access approach is introduced to solve the problem of formation control for a fleet of systems modeled with simple integrator dynamics. We consider a setup where each agent processes information locally. However, all the agents use the same communication channel and database hosted in the cloud, which are both shared resources. The cloud keeps a reduced centralized amount of information and can be accessed by the agents in an asynchronous way under a publish–subscribe paradigm [13], [14]. Basically, the agents can intermittently read/write information on the cloud in an asynchronous way, while being in an idle mode (no communication and no

computing) between any two consecutive accesses. Such control infrastructure appears particularly convenient when communication constraints severely limit the possibility of a direct exchange of information among agents. The use of shared resources hosted in the cloud is widely studied in computer science, where problems such as cloud access management, resource allocations control and content delivery are studied [15], [16], while recently the scheduling of a common computational resource for control system architectures has also been considered [17], [18].

As motivating example, we consider the problem of waypoint generation for control formation of a fleet of autonomous underwater vehicles (AUVs) [19]–[21]. For this kind of agents the communication problems are indeed particularly severe. Specifically, underwater communication is achieved by means of expensive and power hungry acoustic modems and it is considerably limited both in range and bandwidth [19]. Furthermore, the GPS signal is interdicted when the vehicle is underwater and accurate inertial platforms are expensive. Acoustic positioning by means of baselines is also difficult to be adopted in wide open sea scenarios. For this reason, in low cost applications the AUVs are supposed to now and then surface to get their exact feedback position by GPS in order to compensate the effect of ocean currents and external disturbances. This kind of scenario is considered in [22], [23] where, taking advantage of periodic surfacing, wireless communication is used. However, the drawback is that all AUVs have to surface at the same instant in order to communicate with a leader and receive the next waypoint (in terms of time and position). Furthermore, the marine current disturbance is supposed to be the same for all the agents, thus resulting in conservative results when agents are far from each other, possibly experiencing different sea conditions.

The main contribution of this paper is to introduce an asynchronous cloud access for the control system and we exploit a shared database to coordinate a formation of AUVs on the horizontal plane. Specifically, each vehicle surfaces asynchronously with respect to the others, gets its current GPS position and a forecast on the maritime conditions related to its region. Using this information, the AUV computes its new control input and its next surfacing instant. Then, it uploads this new information on the cloud and starts a new underwater navigation segment without being able to communicate or to get GPS until its next surfacing. Note that, between any two consecutive surfacing instants, the others could possibly surface an arbitrary number of times, thus changing their control inputs without the AUVs underwater being able to detect such change. However, despite the pres-

The authors are with the ACCESS Linneaus Center and School of Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden.

This work was partly supported by the Swedish Research Council and the *Knut och Alice Wallenberg* Foundation.

ence of asynchronous and outdated information, we prove that adopting a suitable scheduling rule the fleet converges to the desired formation keeping a residual error within a given bound.

The rest of this paper is organized as follows. In Section II some notations and background concepts are introduced. In Section III the mathematical model of the multi-agent system is presented, the control objective is defined and the proposed control algorithm is outlined. In Section IV sufficient conditions for achieving the control objective are identified and in Section V it is shown how these conditions are attained by scheduling the control updates in an opportunistic way. Section VI shows the application of the proposed algorithm to a simulated vehicle formation problem. Section VII concludes the paper with some possible future developments.

## II. PRELIMINARIES

### A. Notation

For any  $n \in \mathbb{N}$ ,  $\mathbf{1}_n$  denotes the vector in  $\mathbb{R}^n$  whose entries are all equal to one, while  $I_n$  denotes the identity matrix of order  $n$ . Operator  $\|\cdot\|$  used on a vector denotes the Euclidean norm, while if used on a matrix denotes the corresponding induced norm. Operator  $\otimes$  denotes the Kronecker product. For the definition and properties of this operator see for example [24].

### B. Graph Theory

For the purposes of this paper a *graph* is a tuple  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  made up of a set  $\mathcal{V} = \{1, \dots, N\}$  of *nodes* and a set  $\mathcal{E}$  of *edges* connecting distinct nodes. The edge connecting nodes  $i$  and  $j$  is denoted as  $(i, j)$  or  $(j, i)$  indifferently. For each node  $i \in \mathcal{V}$ , the set  $\mathcal{V}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$  of the nodes that are connected to node  $i$  by an edge is called the *neighborhood* of node  $i$  and the nodes  $j \in \mathcal{V}_i$  are called *neighbors* of node  $i$ . The number of neighbors  $d_i$  of a node  $i$  is called the *degree* of that node. A *path* between nodes  $i$  and  $j$  is defined as a sequence  $i, k_1, \dots, k_m, j$  of nodes such that any two consecutive nodes in the sequence are connected by an edge. A graph is said to be *connected* if all possible pairs of nodes are connected by a path. The matrix  $A = \{a_{ij}\}$  such that

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

is called *adjacency matrix* of the graph, the matrix  $D = \text{diag}\{d_1, \dots, d_N\}$  is called *degree matrix* of the graph and finally the matrix  $L = D - A$  is called *Laplacian matrix* of the graph. The Laplacian matrix is symmetric and positive semidefinite and  $\mathbf{1}_N$  is an eigenvector with zero eigenvalue. Moreover, the zero eigenvalue has multiplicity one if and only if the graph is connected [1]. Therefore in a connected graph the eigenvalues of the Laplacian can be denoted as  $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$ . A graph is typically used to describe networked multi-agent systems: each node in the graph represents one agent in the network and an edge between nodes  $i$  and  $j$  represents a possible interaction between the corresponding agents.

## III. SETUP DESCRIPTION

Consider a set of  $N$  dynamical agents described by

$$\dot{x}_i(t) = u_i(t) + \omega_i(t), \quad i = 1, \dots, N, \quad (1)$$

with  $t \geq 0$  and  $x_i(t), u_i(t), \omega_i(t) \in \mathbb{R}^n$ . Here  $x_i(t)$  is the state of agent  $i$ ,  $u_i(t)$  is the control input applied to it and  $\omega_i(t)$  is a disturbance acting on it. Denoting

$$\begin{aligned} x(t) &:= [x_1(t)^\top, \dots, x_N(t)^\top]^\top, \\ u(t) &:= [u_1(t)^\top, \dots, u_N(t)^\top]^\top, \\ \omega(t) &:= [\omega_1(t)^\top, \dots, \omega_N(t)^\top]^\top, \end{aligned}$$

we can rewrite (1) as

$$\dot{x}(t) = u(t) + \omega(t). \quad (2)$$

We consider the *rendezvous* problem, i.e., the problem of driving the states of the agents close to each other in the state space. More precisely, let  $\bar{x}(t)$  be the average of the agents' states,

$$\bar{x}(t) := \frac{1}{N} \sum_{i=1}^N x_i(t),$$

let  $e_i(t)$  be the mismatch between the state of agent  $i$  and the average state,

$$e_i(t) := \bar{x}(t) - x_i(t), \quad i = 1, \dots, N,$$

and collect the signals  $e_i(t)$  into the stack vector

$$e(t) := [e_1(t)^\top, \dots, e_N(t)^\top]^\top.$$

*Definition 1:* We say that the multi-agent system (1) achieves *practical consensus* if

$$\limsup_{t \rightarrow +\infty} \|e(t)\| \leq \varepsilon, \quad (3)$$

where  $\varepsilon > 0$  is a given positive constant.

We assume that the agents cannot directly communicate with each other, but have access to a shared database and a measurement system hosted on a cloud. The communication channel between the agents and the cloud is considered a shared resource with limited throughput capacity, and therefore it must be used at discrete time instants and asynchronously by different agents. We consider piecewise constant control signals with event-triggered updates. The time instants when agent  $i$  updates its control input are denoted as  $t_{i,k}$ , with  $k \in \mathbb{N}$  and we set  $t_{i,0} = 0$ . Namely, we have

$$u_i(t) = u_i(t_{i,k}) \quad \text{for } t \in [t_{i,k}, t_{i,k+1}).$$

For convenience, we introduce the functions  $l_j(t)$  corresponding to the latest update of  $u_j(\cdot)$  before time  $t$  [11],

$$l_j(t) = \max_{k \in \mathbb{N}} \{t_{j,k} : t_{j,k} \leq t\}$$

Note in particular that  $l_i(t_{i,k}) = k$ . It is assumed that when agent  $i$  connects to the cloud, say at time  $t = t_{i,k} = t_{i,l_i(t)}$ , it receives a measurement of its current state  $x(t_{i,k})$ , some information about the other agents stored in the database, and some estimate of the disturbances to which it is subject.

$i$	1	2	...	$N$
$t_{i,l_i(t)}$	$t_{1,l_1(t)}$	$t_{2,l_2(t)}$	...	$t_{N,l_N(t)}$
$x_i(t_{i,l_i(t)})$	$x_1(t_{1,l_1(t)})$	$x_2(t_{2,l_2(t)})$	...	$x_N(t_{N,l_N(t)})$
$u_i(t_{i,l_i(t)})$	$u_1(t_{1,l_1(t)})$	$u_2(t_{2,l_2(t)})$	...	$u_N(t_{N,l_N(t)})$
$\gamma_{i,l_i(t)}$	$\gamma_{1,l_1(t)}$	$\gamma_{2,l_2(t)}$	...	$\gamma_{N,l_N(t)}$
$\rho_{i,l_i(t)}$	$\rho_{1,l_1(t)}$	$\rho_{2,l_2(t)}$	...	$\rho_{N,l_N(t)}$
$t_{i,l_i(t)+1}$	$t_{1,l_1(t)+1}$	$t_{2,l_2(t)+1}$	...	$t_{N,l_N(t)+1}$

TABLE I

DATA STORED IN THE SHARED DATABASE AT A GENERIC TIME  $t$ .

The estimate of the disturbances is given in the form of two coefficients  $\gamma_{i,k}$  and  $\rho_{i,k}$  such that

$$\|\omega_i(t)\| \leq \hat{\omega}_{i,k}(t) := \begin{cases} \gamma_{i,k} & t \in [t_{i,k}, t_{i,k+1}), \\ \gamma_{i,k} + \rho_{i,k}(t - t_{i,k+1}) & t \geq t_{i,k+1}. \end{cases} \quad (4)$$

*Remark 1:* Disturbance estimates (4) are different for different agents and for different update times of the same agent, taking into account that disturbances can vary both in time and space within the operating region. The case of a known global upper bound on the disturbances can still be derived as a particular case, by setting  $\gamma_{i,k} = \gamma$  and  $\rho_{i,k} = 0$  for all  $i = 1, \dots, N$  and all  $k \in \mathbb{N}$ .

This particular model of disturbance estimation is inspired by our motivating example, described at the end of this section. Agent  $i$  uses all such information to compute its new control input  $u_i(t_{i,k})$ , and the time  $t_{i,k+1}$  of the next update. Before closing the connection to the cloud, agent  $i$  uploads the values of  $t_{i,k}$ ,  $x(t_{i,k})$ ,  $u(t_{i,k})$ ,  $\gamma_{i,k}$ ,  $\rho_{i,k}$ , and  $t_{i,k+1}$  on the shared database, so that they can be used later by other agents. Such values may replace the corresponding old ones uploaded by agent  $i$  at the time of the previous connection, so that the dimension of the database does not increase over time. Table I shows the data stored in the shared database at a generic time  $t$ . The control signals are obtained as linear diffusive feedback from other agents in the network. The topology of the interactions is described by a graph  $\mathcal{G}$  where each node represents an agent and the edge  $(i, j)$  represents a feedback interaction between agents  $i$  and  $j$ . Namely, we set

$$u_i(t) = c \sum_{j=1}^N a_{ij} (\hat{x}_j(t_{i,k}) - x_i(t_{i,k})), \quad t \in [t_{i,k}, t_{i,k+1}), \quad (5)$$

where  $A = \{a_{ij}\}$  is the adjacency matrix of  $\mathcal{G}$ ,  $c$  is a positive scalar gain and  $\hat{x}_j(t_{i,k})$  is an estimate of the state of agent  $j$  available at time  $t_{i,k}$ . The estimate  $\hat{x}_j(t_{i,k})$  is obtained by using the data available in the cloud at time  $t_{i,k}$ . Namely, (1) is considered for agent  $j$  under null disturbances, and it is integrated in the interval  $[t_{j,l_j(t_{i,k})}, t)$ , with  $t \leq t_{j,l_j(t_{i,k})+1}$ , yielding

$$\hat{x}_j(t) = x_j(t_{j,l_j(t)}) + u_j(t_{j,l_j(t)})(t - t_{j,l_j(t)}). \quad (6)$$

*Remark 2:* The control input  $u_i(t_{i,k})$  can be computed by agent  $i$  at time  $i$  by using only information downloaded from the cloud at time  $t_{i,k}$ .

For the purposes of the forthcoming analysis, consider also the following signals,

$$z_i(t) := \sum_{j=1}^N a_{ij} (x_j(t) - x_i(t)), \quad (7)$$

and the mismatches  $\tilde{u}_i(t)$  between a control signal  $u_i(t)$  and the corresponding  $z_i(t)$ ,

$$\tilde{u}_i(t) := u_i(t) - cz_i(t). \quad (8)$$

The setup proposed above is suitable to describe a formation control problem for a network of autonomous vehicles under strict communication constraints. The motivating example of the paper is the problem of a waypoint generation algorithm for a two-dimensional formation of AUVs. Each agent represents a vehicle, and the state of agent  $i$  is  $x_i(t) = p_i(t) - b_i \in \mathbb{R}^2$ , where  $p_i(t)$  is the horizontal waypoint trajectory of vehicle  $i$ , i.e. we do not care about the vertical coordinate, and  $b_i$  is a constant offset term with respect to the average point of the fleet, so it describes the position assigned to vehicle  $i$  within the formation. Since radio communication is interdicted underwater (no GPS and no relative information exchange can occur), and since we have assumed that the AUVs are not equipped with expensive sonar modems, the vehicles are completely isolated during the navigation, but that they can surface at discrete time instants to exchange information with a remote repository hosted on a cloud. The disturbances included in the model account for the marine currents influencing the motion of the vehicles. The position measurements may be obtained by GPS and the forecasts (4) on the marine currents may be computed from a MAFOR bulletin obtained from a wireless weather station. In fact, forecasts become more conservative for more distant times in the future, a characteristic which is embedded into model (4).

The proposed control algorithm is summarized below. The algorithm is initialized by setting, for all  $i \in \mathcal{V}$ ,  $t_{i,0} = 0$ ,  $\hat{x}_i(0) = x_i(0)$  and

$$u_i(0) = c \sum_{j=1}^N a_{ij} (x_j(0) - x_i(0)).$$

Each agent  $i \in \mathcal{V}$  at each update time  $t_{i,k}$  performs the following operations.

1. Agent  $i$  connects to the cloud at time  $t = t_{i,k}$ , as scheduled at time  $t_{i,k-1}$ .
2. Agent  $i$  receives the measurement of its current state  $x_i(t_{i,k})$  and uploads it on the shared database.
3. From the database, agent  $i$  downloads  $t_{j,l_j(t_{i,k})}$ ,  $x_j(t_{j,l_j(t_{i,k})})$ ,  $u_j(t_{j,l_j(t_{i,k})})$ ,  $t_{j,l_j(t_{i,k})+1}$ ,  $\gamma_{j,l_j(t_{i,k})}$  and  $\rho_{j,l_j(t_{i,k})}$  for each  $j \in \mathcal{V}_i$ .
4. Using  $x_i(t)$ ,  $t_{j,l_j(t_{i,k})}$ ,  $x_j(t_{j,l_j(t_{i,k})})$  and  $u_j(t_{j,l_j(t_{i,k})})$  for  $j \in \mathcal{V}_i$ , agent  $i$  computes its new control input  $u_i(t_{i,k})$  according to (5).

5. Agent  $i$  uploads its new control input  $u_i(t_{i,k})$  on the cloud.
6. Agent  $i$  computes the parameters  $\gamma_{i,k}, \rho_{i,k}$  by elaborating available information on the disturbances, and uploads them on the cloud.
7. Using  $\gamma_{i,k}, t_{j,l_j(t_{i,k})}, x_j(t_{j,l_j(t_{i,k})}), u_j(t_{j,l_j(t_{i,k})}), t_{j,l_j(t_{i,k})+1}, \gamma_{j,l_j(t_{i,k})}$  and  $\rho_{j,l_j(t_{i,k})}$  for  $j \in \mathcal{V}$ , agent  $i$  schedules the time  $t_{i,k+1}$  of its next update and uploads it on the cloud. An appropriate scheduling rule will be given later in the paper.
8. Agent  $i$  disconnects from the cloud and will be unable to communicate until its next update at time  $t_{i,k+1}$ .

When the particular problem of AUVs coordination is considered,  $t_{i,k}$  is the  $k$ -th surfacing instant for the  $i$ -th vehicle, and step 8 corresponds to the underwater navigation segment between the surfacing times  $t_{i,k}$  and  $t_{i,k+1}$ .

#### IV. PRACTICAL CONSENSUS

In this section our bounded convergence result is derived. Later on, this result will be related to the scheduling of the control updates. The following assumptions are needed.

*Assumption 1:* The graph  $\mathcal{G}$  that describes the feedback interactions is connected, and its Laplacian has eigenvalues  $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$ .

*Assumption 2:* The disturbances  $\omega_i(t)$  acting on each agent  $i = 1, \dots, N$  are uniformly bounded by  $\|\omega_i(t)\| \leq \Omega$ .

*Remark 3:* Assumption 2 is not related to the estimation model (4). The upper bound  $\Omega$  is not used in the disturbance estimation nor in scheduling and computing the control updates, and it is only introduced to characterize the convergence radius  $\varepsilon$  in (3).

*Assumption 3:* There exists a *threshold function*

$$\sigma(t) = \sigma_0 + \sigma_1 e^{-\lambda_\sigma t},$$

with positive constants  $\sigma_0, \sigma_1$  and  $\lambda_\sigma$  such that at any time instant  $t \geq 0$  it holds that

$$\|\tilde{u}_i(t)\| \leq c\sigma(t). \quad (9)$$

*Remark 4:* Assumption 3 can be fulfilled by scheduling the control updates opportunely. This is shown in Section V.

*Theorem 1:* Consider the multi-agent system (1) under controls (5). Suppose Assumptions 1 to 3 hold. Then practical consensus is achieved with

$$\varepsilon = \frac{\lambda_N}{\lambda_2^2} \left( \sqrt{N}\sigma_0 + \frac{\Omega}{c} \right) \quad (10)$$

*Proof:* Consider the following Lyapunov candidate function for the error stack vector  $e(t)$ <sup>1</sup>,

$$V(t) = \sqrt{e(t)^\top (L^2 \otimes I_n) e(t)}, \quad (11)$$

where  $L$  is the Laplacian of the graph  $\mathcal{G}$  that describes the agents' interactions. Denoting

$$z(t) = [z_1(t)^\top, \dots, z_N(t)^\top]^\top$$

we have

$$z(t) = -(L \otimes I_n)x(t) = (L \otimes I_n)e(t). \quad (12)$$

Since  $L$  is symmetric we can write

$$\begin{aligned} e(t)^\top (L^2 \otimes I_n) e(t) &= e(t)^\top (L \otimes I_n)^2 e(t) \\ &= ((L \otimes I_n)e(t))^\top ((L \otimes I_n)e(t)) \\ &= z(t)^\top z(t) = \|z(t)\|^2. \end{aligned}$$

Hence, (11) can be rewritten as

$$V(t) = \|z(t)\|.$$

Consider now the dynamics of this candidate function along the system trajectories. Using (2) and (12) we can write

$$\dot{z}(t) = -(L \otimes I_n)\dot{x}(t) = -(L \otimes I_n)(u(t) + \omega(t)). \quad (13)$$

Now denote

$$\tilde{u}(t) = [\tilde{u}_1(t)^\top, \dots, \tilde{u}_N(t)^\top]^\top,$$

so that we have

$$u(t) = \tilde{u}(t) + cz(t),$$

which substituted into (13) yields

$$\dot{z}(t) = -(L \otimes I_n)(\tilde{u}(t) + cz(t) + \omega(t)).$$

Consequently, we have

$$\begin{aligned} \dot{V}(t) &= \frac{d}{dt} \|z(t)\| = \frac{z(t)^\top \dot{z}(t)}{\|z(t)\|} \\ &= - \frac{z(t)^\top (L \otimes I_n)(\tilde{u}(t) + cz(t) + \omega(t))}{\|z(t)\|} \end{aligned} \quad (14)$$

By the properties of the Kronecker product and the Euclidean norm we have

$$\begin{aligned} -z(t)^\top (L \otimes I_n)z(t) &\leq -\lambda_2 \|z(t)\|^2 \\ -z(t)^\top (L \otimes I_n)\tilde{u}(t) &\leq \|z(t)\| \lambda_N \|\tilde{u}(t)\| \\ -z(t)^\top (L \otimes I_n)\omega(t) &\leq \|z(t)\| \lambda_N \|\omega(t)\| \end{aligned}$$

Substituting these inequalities into (14) yields

$$\dot{V}(t) \leq -c\lambda_2 V(t) + \lambda_N \|\tilde{u}(t)\| + \lambda_N \|\omega(t)\|. \quad (15)$$

By Assumption 3 we have

$$\|\tilde{u}(t)\| \leq c\sqrt{N}\sigma(t),$$

which substituted into (15) yields

$$\dot{V}(t) \leq -c\lambda_2 V(t) + c\lambda_N \sqrt{N}\sigma(t) + \lambda_N \|\omega(t)\|. \quad (16)$$

Accounting for Assumptions 1 and 2, (16) implies that

$$\limsup_{t \rightarrow \infty} V(t) \leq \frac{\lambda_N}{\lambda_2} \left( \sqrt{N}\sigma_0 + \frac{\Omega}{c} \right) \quad (17)$$

Now observe that, from the Rayleigh-Ritz theorem [24], we have

$$V(t) = \|(L \otimes I_n)e(t)\| \geq \lambda_2 \|e(t)\|,$$

<sup>1</sup>When a planar formation problem is considered,  $n = 2$

or equivalently

$$\|e(t)\| \leq \frac{V(t)}{\lambda_2}. \quad (18)$$

Therefore, using (17) into (18), and taking the limit for  $t \rightarrow +\infty$ , we have

$$\begin{aligned} \limsup_{t \rightarrow +\infty} \|e(t)\| &\leq \frac{\lambda_N}{\lambda_2^2} \limsup_{t \rightarrow +\infty} \left( \sqrt{N} \sigma(t) + \frac{\|\omega(t)\|}{c} \right) \\ &\leq \frac{\lambda_N}{\lambda_2^2} \left( \sqrt{N} \sigma_0 + \frac{\Omega}{c} \right). \end{aligned}$$

*Remark 5:* The convergence radius (10) can be arbitrarily reduced by increasing the control gain  $c$ , reducing the threshold  $\sigma_0$  or considering a better connected network, which corresponds to a smaller ratio  $\lambda_N/\lambda_2^2$ .

## V. SCHEDULING CONTROL UPDATES

In this section we present our main result. Specifically, we give a criterion for recursive scheduling of the control updates  $t_{i,k}$  such that Assumption 3 for Theorem 1 holds, and so practical consensus is achieved. Denote

$$\hat{z}_i(t) := \sum_{j=1}^N a_{ij} (\hat{x}_j(t) - \hat{x}_i(t)), \quad (19)$$

$$\hat{z}(t) := [\hat{z}_1(t)^\top, \dots, \hat{z}_N(t)^\top]^\top, \quad (20)$$

$$\alpha := q\sigma_0, \quad (20)$$

$$\beta(t) := (1-q)\sigma_0 + \sigma_1 e^{-\lambda\sigma t}, \quad (21)$$

where  $0 < q < 1$ . Let us introduce the following functions:

$$\begin{aligned} \hat{\Omega}_{i,k}(t) &:= \int_{t_{i,k}}^t \hat{\omega}_{i,k}(\tau) d\tau \\ &= \begin{cases} \gamma_{i,k}(t - t_{i,k}) & t \in [t_{i,k}, t_{i,k+1}), \\ \gamma_{i,k}(t - t_{i,k}) + \frac{1}{2} \rho_{i,k}(t - t_{i,k+1})^2 & t \geq t_{i,k+1}, \end{cases} \end{aligned}$$

$$B_{i,k}(t) := \sqrt{\sum_{j=1}^N \hat{\omega}_{j,l_j(t_{i,k})}(t)^2},$$

$$R_{i,j,k}(t) := \max \left\{ \frac{\|u(t_{i,k})\|}{c} + \sqrt{N} \sigma(t_{i,k}), \frac{\lambda_N}{\lambda_2} \left( \sqrt{N} \sigma(t_{j,l_j(t_{i,k})+1}) + \frac{B_{i,k}(t)}{c} \right) \right\},$$

$$\begin{aligned} S_{i,k}(t) &:= \|d_i u_i(t_{i,k})(t - t_{i,k}) \\ &\quad - \sum_{j:t < t_{j,l_j(t_{i,k})+1}}^N u_j(t_{j,l_j(t_{i,k})})(t - t_{i,k}) \\ &\quad - \sum_{j:t \geq t_{j,l_j(t_{i,k})+1}}^N u_j(t_{j,l_j(t_{i,k})})(t_{j,l_j(t_{i,k})+1} - t_{i,k}) \Big\| \\ &\quad + c \sum_{j:t \geq t_{j,l_j(t_{i,k})+1}}^N \int_{t_{j,l_j(t_{i,k})+1}}^t R_{i,j,k}(\tau) + \sigma(\tau) d\tau, \end{aligned}$$

where  $d_i$  is the degree of node  $i$  in the graph  $\mathcal{G}$ . Now recalling (20) and (21), consider the following scalars:

$$\begin{aligned} T_{i,j,k} &= \inf \left\{ \tau > t_{i,k} : \hat{\Omega}_{j,l_j(t_{i,k})}(\tau) \geq \frac{\alpha}{2d_{\max}} \right\}, \\ T'_{i,k} &= \inf \left\{ \tau > t_{i,k} : S_{i,k}(\tau) \geq \beta(\tau) \right\}, \end{aligned}$$

where we denoted  $d := \max\{d_1, \dots, d_N\}$ . With this notation, the control updates are recursively scheduled as

$$t_{i,k+1} = \min_{j \in \mathcal{V}_i \cup \{i\}} \{T_{i,j,k}, T'_{i,k}\}. \quad (22)$$

The scheduling rule (22) introduces a degree of freedom in the choice of the next connection  $t_{i,k+1}$  to the cloud.

*Remark 6:* The time instant  $t_{i,k+1}$  can be computed by agent  $i$  at time  $t_{i,k}$  by only using information available on the cloud at time  $t_{i,k}$ . In particular, the values  $u_i(t_{i,k})$  and  $u_j(t_{j,l_j(t_{i,k})})$  are directly available, cfr. Table I. Together with Remark 2, this implies that no centralized computation is required to implement the proposed control algorithm. The cloud is only used as a data repository and it does not need to process information. All the necessary computing can be done by the agents in a decentralized way accessing the cloud asynchronously.

*Remark 7:* Note that  $\hat{\Omega}_{i,k}(t_{i,k}) = 0$  and  $S_{i,k}(t_{i,k}) = 0$  for any  $k \in \mathbb{N}$  and for any  $i \in \mathcal{V}$ . Moreover,  $\hat{\Omega}(t)$  and  $S_{i,k}(t)$  are continuous in  $t$  with upper-bounded slope. Since  $\alpha/d_{\max}$  is a positive constant and  $\beta(\tau)$  is lower-bounded by a positive constant, this implies that times  $T_{i,j,k}$  and  $T'_{i,k}$  cannot be infinitely close to  $t_{i,k}$ . Consequently the inter-update times  $t_{i,k+1} - t_{i,k}$  are lower-bounded by some positive constant and the updates do not present accumulation points.

*Theorem 2:* Consider the multi-agent system (1) under controls (5). Suppose Assumptions 1 and 2 hold and let the control updates  $t_{i,k}$  be scheduled according to (22). Then practical consensus is achieved with  $\varepsilon$  as in (10).

*Proof:* We are going to prove that if the control updates are scheduled according to (22), then (9) holds for all the agents  $i \in \mathcal{V}$  and at all the time instants  $t \geq 0$ . Then the claim is obtained from Theorem 1.

Since  $t_{i,0} = 0$  for all  $i \in \mathcal{V}$ , we have  $\tilde{u}_i(0) = 0 < \sigma(0)$ , and therefore at time zero (9) holds for all the agents. Now suppose by contradiction that at a finite time  $t$  some agent  $i$  attains  $\|\tilde{u}_i(t)\| > c\sigma(t)$  for the first time, while  $\|\tilde{u}_j(\tau)\| \leq c\sigma(\tau)$  for all  $\tau \in [0, t)$  and all  $j \in \mathcal{V}$ . Denote also  $k = l_i(t)$ , i.e., let  $t_{i,k}$  be the latest update for agent  $i$  before  $t$ . Adding and subtracting  $c\hat{z}_i(t)$  on the right-hand side of (8) we obtain

$$\tilde{u}_i(t) = c(\hat{z}_i(t_{i,k}) - \hat{z}_i(t) + \hat{z}_i(t) - z_i(t)).$$

Taking the norm of both sides and applying the triangular inequality yields

$$\|\tilde{u}_i(t)\| \leq c \|\hat{z}_i(t_{i,k}) - \hat{z}_i(t)\| + c \|\hat{z}_i(t) - z_i(t)\|. \quad (23)$$

By the contradiction hypothesis we have  $\|\tilde{u}_i(t)\| > c\sigma(t)$ , therefore (23) implies

$$\sigma(t) < \|\hat{z}_i(t_{i,k}) - \hat{z}_i(t)\| + \|\hat{z}_i(t) - z_i(t)\|. \quad (24)$$

First consider the term  $\|\hat{z}_i(t) - z_i(t)\|$ . We have

$$\hat{z}_i(t) - z_i(t) = \sum_{j=1}^N a_{ij} (\hat{x}_j(t) - \hat{x}_i(t) - x_j(t) + x_i(t)),$$

and consequently

$$\|\hat{z}_i(t) - z_i(t)\| \leq d_i \|\hat{x}_i(t) - x_i(t)\| + \sum_{j \in \mathcal{Y}_i} \|\hat{x}_j(t) - x_j(t)\|. \quad (25)$$

Consider now the terms  $\|\hat{x}_j(t) - x_j(t)\|$ . Integrating (1) in  $[t_{j,l_j(t_{i,k})}, t)$ , with  $t < t_{j,l_j(t_{i,k})+1}$ , we have

$$\begin{aligned} x_j(t) &= x_j(t_{j,l_j(t_{i,k})}) \\ &+ u_j(t_{j,l_j(t_{i,k})})(t - t_{j,l_j(t_{i,k})}) + \int_{t_{j,l_j(t_{i,k})}}^t \omega_j(\tau) d\tau. \end{aligned} \quad (26)$$

On the other hand,  $\hat{x}_j(t)$  can be computed as in (6). Therefore, using (6) and (26), we have

$$\|\hat{x}_j(t) - x_j(t)\| \leq \left\| \int_{t_{j,l_j(t_{i,k})}}^t \omega_j(\tau) d\tau \right\| \leq \int_{t_{j,l_j(t_{i,k})}}^t \|\omega_j(\tau)\| d\tau,$$

which by (4) also implies

$$\|\hat{x}_j(t) - x_j(t)\| \leq \hat{\Omega}_{j,l_j(t_{i,k})}(t).$$

The same reasoning can be carried out for the term  $\|\hat{x}_i(t) - x_i(t)\|$ , yielding

$$\|\hat{x}_i(t) - x_i(t)\| \leq \hat{\Omega}_{i,k}(t).$$

Substituting the two previous inequalities into (25) yields

$$\|\hat{z}_i(t) - z_i(t)\| \leq d_i \hat{\Omega}_{i,k}(t) + \sum_{j \in \mathcal{Y}_i} \hat{\Omega}_{j,l_j(t_{i,k})}(t).$$

Since (22) is applied, we have  $\hat{\Omega}_{i,k}(t) \leq \frac{\alpha}{2d_{\max}}$ , and consequently

$$\|\hat{z}_i(t) - z_i(t)\| \leq d_i \frac{\alpha}{2d_{\max}} + d_i \frac{\alpha}{2d_{\max}} \leq \alpha. \quad (27)$$

Consider now the term  $\|\hat{z}(t_{i,k}) - \hat{z}_i(t)\|$  in (23). Recalling (7) and noting that  $\hat{x}_i(t_{i,k}) = x_i(t_{i,k})$ , because at time  $t_{i,k}$  vehicle  $i$  receives the exact measurement of its state, we have

$$\hat{z}_i(t_{i,k}) - \hat{z}_i(t) = \sum_{j=1}^N a_{ij} (\hat{x}_j(t_{i,k}) - x_i(t_{i,k}) - \hat{x}_j(t) + \hat{x}_i(t)). \quad (28)$$

Focusing on the term  $\hat{x}_i(t) - x_i(t_{i,k})$ , by (6) applied for  $j = i$ , we have

$$\hat{x}_i(t) - x_i(t_{i,k}) = u_i(t_{i,k})(t - t_{i,k}). \quad (29)$$

Similar reasoning can be applied to the terms  $\hat{x}_j(t_{i,k}) - \hat{x}_j(t)$ . However, since the control updates are asynchronous,  $u_j(\tau)$  may be updated one or multiple times during the time interval  $[t_{i,k}, t_{i,k+1})$ . Namely, in the time interval  $[t_{i,k}, t_{j,l_j(t_{i,k})+1})$ ,  $u_j$  has value  $u_j(t_{j,l_j(t_{i,k})})$ , which is available in the cloud at time  $t_{i,k}$ , but the possible future values assumed by  $u_j(\tau)$  for  $\tau \geq t_{j,l_j(t_{i,k})+1}$  are unknown at time  $t_{i,k}$ . Hence we can write

$$\hat{x}_j(t) - \hat{x}_j(t_{i,k}) = \int_{t_{i,k}}^t u_j(\tau) d\tau$$

$$= \begin{cases} u_j(t_{j,l_j(t_{i,k})})(t - t_{i,k}) & t \leq t_{j,l_j(t_{i,k})+1}, \\ u_j(t_{j,l_j(t_{i,k})}) \cdot (t_{j,l_j(t_{i,k})+1} - t_{i,k}) \\ + \int_{t_{j,l_j(t_{i,k})+1}}^t u_j(\tau) d\tau & t > t_{j,l_j(t_{i,k})+1}. \end{cases} \quad (30)$$

Substituting (29) and (30) into (28), taking norms of both sides, and applying the triangular inequality yields

$$\begin{aligned} \|\hat{z}_i(t_{i,k}) - \hat{z}_i(t)\| &\leq \|d_i u_i(t_{i,k})(t - t_{i,k}) \\ &- \sum_{j:t < t_{j,l_j(t_{i,k})+1}}^N u_j(t_{j,l_j(t_{i,k})})(t - t_{i,k}) \\ &- \sum_{j:t \geq t_{j,l_j(t_{i,k})+1}}^N u_j(t_{j,l_j(t_{i,k})}) (t_{j,l_j(t_{i,k})+1} - t_{i,k})\| \\ &+ \sum_{j:t \geq t_{j,l_j(t_{i,k})+1}}^N \int_{t_{j,l_j(t_{i,k})+1}}^t \|u_j(\tau)\| d\tau. \end{aligned} \quad (31)$$

Consider now an agent  $j$  that updates its control at least once before time  $t$ , and focus on the term  $\|u_j(\tau)\|$  with  $\tau \in [t_{j,l_j(t_{i,k})+1}, t)$ . Since (9) holds for all the agents until time  $t$ , we can write

$$\|u_j(\tau)\| \leq c \|z_j(\tau)\| + \|\tilde{u}_j(\tau)\| \leq c (\|z(\tau)\| + \sigma(\tau)). \quad (32)$$

Moreover, in  $\tau \in [t_{j,l_j(t_{i,k})+1}, t)$ , since (9) holds, the state of the system converges to the region described by (17). Therefore, taking into account that  $\|\omega(\tau)\| \leq B_{i,k}(\tau)$  and  $\sigma(\tau) \leq \sigma(t_{i,k})$ , we can write

$$\|z(\tau)\| \leq \max \left\{ \|z(t_{i,k})\|, \frac{\lambda_N}{\lambda_2} \left( \sqrt{N} \sigma(t_{i,k}) + \frac{B_{i,k}(\tau)}{c} \right) \right\}. \quad (33)$$

Also, since  $t_{i,k} \leq \tau$ , (9) holds for all the agents at time  $t_{i,k}$ , and we can write

$$\|z(t_{i,k})\| \leq \frac{\|u(t_{i,k})\|}{c} + \sqrt{N} \sigma(t_{i,k}). \quad (34)$$

Using (33) and (34) into (32) we can write

$$\|u_j(\tau)\| \leq c (R_{i,j,k}(\tau) + \sigma(\tau)),$$

which substituted into (31) yields

$$\|\hat{z}_i(t_{i,k}) - \hat{z}_i(t)\| \leq S_{i,k}(t).$$

Now since (22) is applied, we have  $S_{i,k}(t) \leq \beta(t)$ , and consequently

$$\|\hat{z}_i(t_{i,k}) - \hat{z}_i(t)\| \leq \beta(t). \quad (35)$$

Now substituting (27) and (35) into (24), we have

$$\sigma(t) < \alpha + \beta(t)$$

This is a contradiction, since  $\alpha$  and  $\beta(t)$  are defined so that  $\sigma(t) = \alpha + \beta(t)$ . We can conclude that (9) holds for all the agents  $i$  at all times  $t \geq 0$ .

Now since (9) holds for all the agents uniformly, Assumptions 1 to 3 hold, and Theorem 1 can be applied. Hence practical consensus is attained with radius (10). ■

*Remark 8:* Since Theorem 2 only requires that  $t_{i,k+1} \leq T_{i,j,k}$  for all  $j \in \mathcal{V}$  and  $t_{i,k+1} \leq T'_{i,k}$ , the scheduling rule (22) may be relaxed to

$$t_{i,k} < t_{i,k+1} \leq \min_{j \in \mathcal{V}_i \cup \{i\}} \{T_{i,j,k}, T'_{i,k}\}. \quad (36)$$

This gives each agent a degree of freedom in the scheduling of the next connection to the cloud. Such degree of freedom may be exploited to avoid cloud congestion due to multiple contemporary accesses. In fact, if (36) is enforced, agent  $i$  is free to choose  $t_{i,k+1}$  in a given interval, and since it is aware of the subsequent update times  $t_{j,l_j(t_{i,k}+1)}$  of all the other agents  $j \neq i$ , it can schedule  $t_{i,k+1}$  so that it does not coincide with any of these instants.

*Remark 9:* An alternative upper bound to (34) is

$$\|z(t_{i,k})\| \leq \|\hat{z}(t_{i,k})\| + \sqrt{N}\alpha.$$

Therefore, function  $R_{i,j,k}(t)$  can also be designed as

$$R_{i,j,k}(t) := \max \left\{ \min \left\{ \frac{\|u(t_{i,k})\|}{c} + \sqrt{N}\sigma(t_{i,k}), \|\hat{z}(t_{i,k})\| + \sqrt{N}\alpha \right\}, \frac{\lambda_N}{\lambda_2} \left( \sqrt{N}\sigma(t_{j,l_j(t_{i,k}+1)}) + \frac{B_{i,k}(t)}{c} \right) \right\},$$

In this case, when scheduling the update time  $t_{i,k+1}$ , agent  $i$  needs to compute  $\|\hat{z}(t_{i,k})\|$ . If the topology of the connections among the vehicles is known,  $\|\hat{z}(t_{i,k})\|$  can be computed by using (19) with  $t = t_{i,k}$ .

## VI. NUMERICAL SIMULATIONS

In order to corroborate the theoretical results, we applied the proposed control algorithm to a formation problem on a simulated network made up of  $N = 5$  planar vehicles. The topology of the connections is described by a complete graph, so that every agent receives feedback from every other agent. The desired formation is described by the offsets  $[(-25, -25), (-25, 25), (0, 0), (25, -25), (25, 25)]$ . The simulation takes place the time span  $[0, 50]$ . The agents are spawned in initial positions randomly extracted in a square of 200 by 200. We pick a control gain  $c = 0.01$  and a threshold function  $\sigma(t) = 1.4 \cdot 10^3 + 0.8 \cdot 10^3 e^{-0.05t}$ . A different value of the additive disturbance is chosen for each agent, randomly extracted in the range  $(-1.0, 1.0)$  on both coordinates. At each hundredth of second this value is changed with probability  $5 \cdot 10^{-3}$ , by randomly extracting a new value from the same range. To model the forecast that the agents receive about such disturbances, at each update of a vehicle  $i$  we assign to  $\gamma_{i,k}$  a value of  $\sqrt{2} \cdot (1.0 + r_\gamma)$ , where  $r_\gamma$  is randomly extracted in  $(0.0, 1.0)$ , while we assign value zero to  $\rho_{i,k}$ . With these choices we have that the norm of the disturbances is always below the estimate that the agents receive. Figure 1 illustrates the convergence of the first position variable for each vehicle during the simulation.

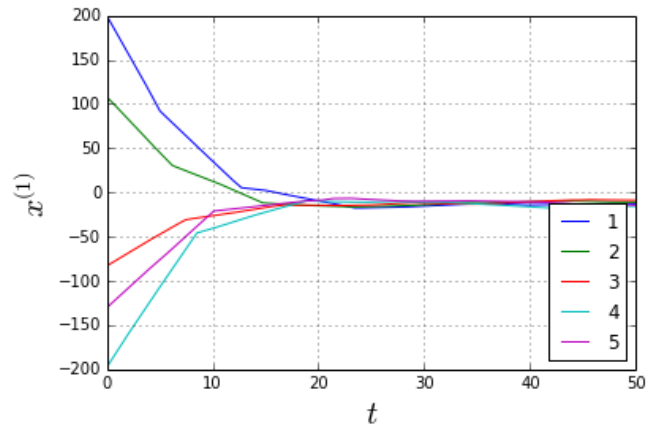


Fig. 1. Trend of the first consensus variable  $x_i^{(1)} = p_i^{(1)} - b_i^{(1)}$ , for each agent  $i = 1, \dots, 5$  during the simulation.

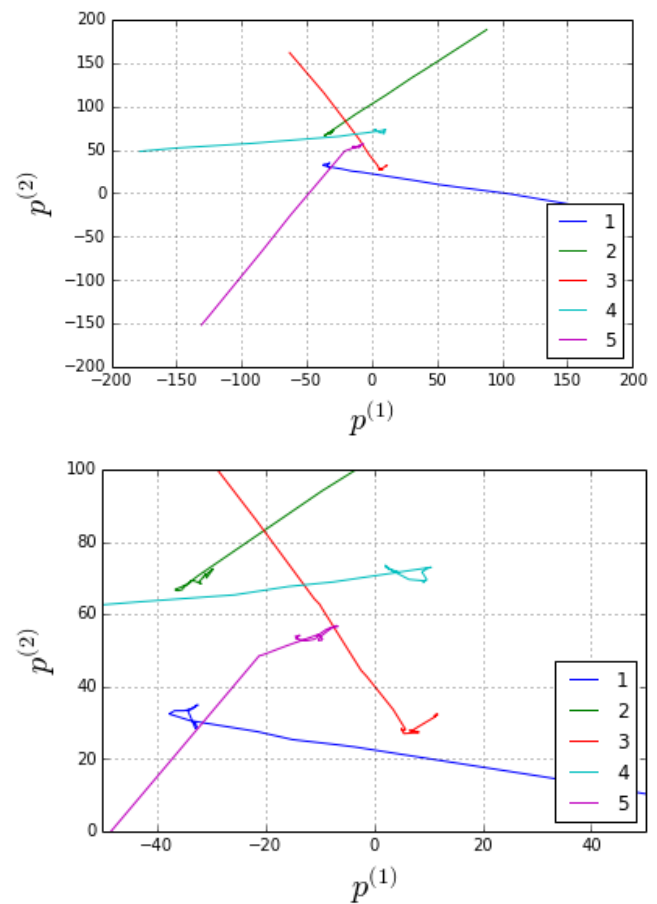


Fig. 2. Paths  $p_i(t)$  executed by each vehicle  $i = 1, \dots, 5$  during the simulation (upper) and detail (lower).

Figure 2 shows the two-dimensional paths. Finally, Table II shows the update times  $t_{i,k}$  in the time span  $[0, 50]$  for each agent  $i = 1, \dots, 5$ .

## VII. CONCLUSIONS

A cloud-based control algorithm has been proposed for practical consensus of a network of agents with integrator

$k$	$t_{1,k}$	$t_{2,k}$	$t_{3,k}$	$t_{4,k}$	$t_{5,k}$
0	0.00	0.00	0.00	0.00	0.00
1	5.01	6.21	7.41	8.51	10.11
2	12.72	14.72	16.72	18.81	21.31
3	23.32	25.82	28.02	30.41	32.61
4	34.92	37.23	39.63	41.92	44.22
5	46.53	48.84			

TABLE II  
UPDATE TIMES IN THE TIME SPAN  $[0, 50]$ .

dynamics under event-triggered updates and additive disturbances. Sufficient conditions for convergence have been identified in terms of the network topology and of the scheduling of the control updates. The proposed approach combines the benefits of event/self-triggered control schemes with the advantage of having a shared asynchronous cloud support. Specifically, each agent schedules its own sequence of cloud accesses in order to achieve a coordinated network formation. The setup is particularly convenient for those applications where direct communication among agents is not always feasible, such as formation control for AUVs. For this problem, the control algorithm overcomes the limitation of having a pre-assigned trajectory for the whole fleet as well as the synchronization of the surfacing of all the agents [20], [23]. Future work will further develop the approach of the paper considering different scheduling laws for the cloud accesses as well as other control objectives, e.g. leader–follower control. Furthermore, more complex agent dynamics and more complex models for the forecast on the disturbances will be studied.

#### REFERENCES

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[2] W. Ren, R. W. Beard, and E. M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference*, Portland, Oregon, USA, 2005.

[3] M. Arcak. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, 2007.

[4] Dimos V. Dimarogonas and Kostas J. Kyriakopoulos. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica*, 44(10):2648–2654, 2008.

[5] D. Nesić and A. R. Teel. Input-output stability properties of networked control systems. *IEEE Transaction on Automatic Control*, 49(10):1650–1667, 2004.

[6] M. Mazo and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2456–2461, Oct 2011.

[7] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transaction on Automatic Control*, 52(9):1680–1685, 2007.

[8] A. Anta and P. Tabuada. To sample or not to sample: self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55:2030–2042, 2010.

[9] X. Wang and M. D. Lemmon. Self-triggered feedback control systems with finite-gain L2 stability. *IEEE Transactions on Automatic Control*, 54:452–467, 2009.

[10] O. Demir and J. Lunze. Event-based synchronisation of multi-agent systems. In *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems*, 2012.

[11] D. Liuzza, D. V. Dimarogonas, M. di Bernardo, and K. H. Johansson. Distributed model-based event-triggered control for synchronization of multi-agent systems. In *IFAC Conference on Nonlinear Control Systems (NOLCOS)*, Toulouse, France, 2013.

[12] C. De Persis and P. Frasca. Self-triggered coordination with ternary controllers. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2012.

[13] P. T. Eugster, P. A. Felber, R. Guerraoui, and Kermarrec A. M. The many faces of publish/subscribe. *ACM Computing Surveys*, 35:114–131, 2003.

[14] G. Cugola and H. A. Jacobsen. Using publish/subscribe middleware for mobile systems. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6:25–33, 2002.

[15] M. Ansbjerg Kjer, M. Kihl, and A. Robertsson. Resource allocation and disturbance rejection in web servers using SLAs and virtualized servers. *IEEE Transactions on Network Service Management*, 6:226–239, 2009.

[16] H. C. Lim, S. Babu, J. S. Chase, and S. S. Parekh. Automated control in cloud computing: challenges and opportunities. In *Workshop on Automated control for datacenters and clouds, ACDC*, Barcelona, Spain, 2009.

[17] S. Samii, P. Eles, Z. Peng, Tabuada P., and A. Cervin. Dynamic scheduling and control-quality optimization of self-triggered control applications. In *IEEE Real-Time Systems Symposium*, San Diego, California, US, 2010.

[18] Y. Xu, K.-E. Årzén, Bini E., and A. Cervin. Response time driven design of control systems. In *The 19th World Congress of the International Federation of Automatic Control*, 2014.

[19] N. A. Cruz, B. M. Ferreira, Kebkal O., A. C. Matos, C. Petrioli, R. Petrocchia, and D. Spaccini. Investigation of underwater acoustic networking enabling the cooperative operation of multiple heterogeneous vehicles. *Marine Technology Society Journal*, 47:43–58, 2013.

[20] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in Monterey bay. *IEEE Journal of Oceanic Engineering*, 31:935–948, 2006.

[21] W. Yan, R. Cui, and D. Xu. Formation control of underactuated autonomous underwater vehicles in horizontal plane. In *IEEE International Conference on Automation and Logistics*, Qingdao, China, 2008.

[22] P. V. Teixeira, D. V. Dimarogonas, K. H. Johansson, and J. Sousa. Event-based motion coordination of multiple underwater vehicles under disturbances. In *OCEANS’10 IEEE*, Sidney, Australia, 2010.

[23] P. V. Teixeira, D. V. Dimarogonas, K. H. Johansson, and J. Sousa. Multi-agent coordination with event-based communication. In *American Control Conference*, Baltimore, Maryland, US, 2010.

[24] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.