

Cloud-supported self-triggered control for multi-agent circumnavigation

Clara Cavaliere*, Dario Mariniello*, Antonio Adaldo, Francesco Lo Iudice,
Dimos V. Dimarogonas, Karl H. Johansson, & Mario di Bernardo

Abstract—In this paper, we propose a cloud-supported control framework for multi-agent circumnavigation missions. We consider a network of planar autonomous agents. Our objective is for the agents to circumnavigate a target with a desired angular speed, while forming a regular polygon around the target. We propose self-triggered rules to schedule the bearing measurements and the cloud accesses for each agent.

I. INTRODUCTION

The problem of tracking and circumnavigating a target with a network of autonomous agents finds numerous applications in mobile robotics. This problem has been studied for a single agent in [1], [2], while the case of taking a network of autonomous agents to a circling formation is the subject of a vast body of work, including, for example, [3]–[7]. Recently, these two problems have been merged in [8], [9]. The vast majority of the existing works on circumnavigation are based on the underlying assumption that each agent may perform measurements and/or exchange information with the other agents in a continuous-time fashion. However, in reality, both the measurements and the exchange of information happen through wireless communication channels with limited bandwidth capacity. Therefore, the frequency with which the agents may perform these actions is limited. In this paper we propose a control framework for multi-agent circumnavigation where bearing measurements and communication are event-triggered and self-triggered, respectively [10]. These schemes are applied to multi-agent systems to achieve the desired coordination while reducing the amount of information exchanged among different agents [11]. Moreover, instead of letting the agents communicate directly with each other, we let them transfer data over a shared information repository hosted on a cloud [12]–[15]. The effectiveness of the proposed algorithms is verified analytically where possible or by numerical simulations.

C. Cavaliere, D. Mariniello, F. Lo Iudice, and M. di Bernardo are with the Department of Electrical Engineering and Information Technology, University of Naples, Italy. Emails: {clar.cavaliere, da.mariniello}@studenti.unina.it, {francesco.loiudice2, mario.dibernardo}@unina.it.

A. Adaldo, D. V. Dimarogonas, and K. H. Johansson are with the Department of Automatic Control, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Osquldas väg 10, 10044, Stockholm, Sweden. Emails: {adaldo, dimos, kallej}@kth.se

This work has received funding from the European Union Horizon 2020 Research and Innovation Programme under the Grant Agreement No. 644128, AEOROWORKS; from the Swedish Foundation for Strategic Research; from the Swedish Research Council; and from the Knut och Alice Wallenberg foundation.

C. Cavaliere and D. Mariniello wish to thank the EU for providing funding to visit KTH under the Erasmus program.

* These two authors have contributed equally to this work.

II. PRELIMINARIES

Given two scalars, i and j , we denote by $i\%j$ the remainder of the division i/j . Given two vectors $u, v \in \mathbb{R}^2$, the angle between u and v computed counterclockwise from u to v is denoted $\angle(u, v)$. For any vector $u \in \mathbb{R}^2$, $\text{rot}(u, \theta)$ denotes the vector obtained rotating u by θ radians counterclockwise. For brevity, we denote $u^\perp = \text{rot}(u, -\pi/2)$.

Let us consider a digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{1, \dots, N\}$ is the set of the N graph vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of the graph edges. The i, j -th element \mathcal{L}_{ij} of the Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N \times N}$ of the graph \mathcal{G} is obtained as

$$\mathcal{L}_{ij} = \begin{cases} -1 & \text{if } (j, i) \in \mathcal{E}, \\ \sum_{k \neq i} -\mathcal{L}_{ik} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Let $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$. Then, we can define the incidence matrix of the digraph \mathcal{G} as the matrix $B \in \mathbb{R}^{N \times M}$ such that its i, k -th element is defined as

$$B_{i,k} := \begin{cases} 1 & \text{if } e_k = (j, i) \text{ for some } j \in \mathcal{V}, \\ -1 & \text{if } e_k = (i, j) \text{ for some } j \in \mathcal{V}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We define the in-incidence matrix $B^\circ \in \mathbb{R}^{N \times M}$ as the matrix obtained from B by substituting zero to each of its negative entries. A directed path is a sequence of edges $\{e_{k_1}, e_{k_2}, \dots, e_{k_L}\}$ such that if $e_{k_l} = (i, j)$ for some vertices i and j , then $e_{k_{l+1}} = (j, m)$ for some vertex m . A spanning tree (if it exists) of the graph \mathcal{G} is a subgraph $\mathcal{G}^T(\mathcal{V}, \mathcal{T})$ of \mathcal{G} encompassing $N - 1$ edges and a node, say r , such that there exists a directed path from r to any other node in \mathcal{V} . Assume \mathcal{G} has a spanning tree \mathcal{G}^T . Accordingly, without loss of generality, we can partition the network edges in the set $\mathcal{T} = \{e_1, e_2, \dots, e_{N-1}\}$ and $\bar{\mathcal{T}} = \{e_{N+1}, e_2, \dots, e_M\}$. Accordingly, we have that $B = [B_{\mathcal{T}} \ B_{\bar{\mathcal{T}}}]$, and $B^\circ = [B_{\mathcal{T}}^\circ \ B_{\bar{\mathcal{T}}}^\circ]$. Finally, as $B_{\mathcal{T}}$ is full column rank, it has a left pseudo-inverse $B_{\mathcal{T}}^\dagger$ we have that the reduced edge Laplacian

$$\mathcal{L}_r := B_{\mathcal{T}}^\top (B_{\mathcal{T}}^\circ + B_{\bar{\mathcal{T}}}^\circ (B_{\mathcal{T}}^\dagger B_{\bar{\mathcal{T}}})^\top) \quad (3)$$

is positive definite [16]. The following Lemma 1 is used to prove our main result.

Lemma 1: Let N distinct vectors $\varphi_1, \dots, \varphi_N \in \mathbb{R}^2$ be such that $\angle(\varphi_i, \varphi_{i+1}) = \angle(\varphi_j, \varphi_{j+1})$ for any two $i, j \in \{1, \dots, N\}$, where we have denoted $\varphi_{N+1} = \varphi_1$. Then, for all $i \in \{1, \dots, N\}$, we have $\min_{j \neq i} \angle(\varphi_i, \varphi_j) = 2\pi/N$. ■

Proof: Let $\bar{\beta} = \angle(\varphi_i, \varphi_{i+1})$ and $\theta_i = \min_{j \neq i} \angle(\varphi_i, \varphi_j)$. Denote as ν_i the index j that attains

the minimum $\angle(\varphi_i, \varphi_j)$. Then, we have $\angle(\varphi_i, \varphi_{i+1}) = \beta = \angle(\varphi_{\nu_i}, \varphi_{\nu_{i+1}})$, and, consequently, $\angle(\varphi_{i+1}, \varphi_{\nu_{i+1}}) = \angle(\varphi_i, \varphi_{\nu_i}) = \theta_i$. However, since $\varphi_{\nu_{i+1}}$ must precede φ_{ν_i+1} when proceeding counterclockwise from φ_i , we have $\theta_{i+1} = \angle(\varphi_{i+1}, \varphi_{\nu_{i+1}}) \leq \angle(\varphi_i, \varphi_{\nu_i}) = \theta_i$, where we have denoted $\theta_{N+1} = \theta_1$. Since the indexes are circular, we must conclude that $\theta_i = \theta_j$ for all $i, j \in \{1, \dots, N\}$. ■

III. SYSTEM MODEL

We consider a network of N autonomous vehicles modeled as planar, first-order integrators, described by

$$\dot{y}_i(t) = u_i(t), \quad i \in \{1, \dots, N\}, \quad (4)$$

where $y_i(t) \in \mathbb{R}^2$ is the position of the agent, and $u_i(t) \in \mathbb{R}^2$ is the decentralized control action exerted on the agent, thus having the dimensions of a velocity. The agents are required to locate and circumnavigate a target, whose position is denoted by x , while forming a balanced circular formation around the target. We define the *counterclockwise angle* between two agents i and j as the angle $\beta_i^j(t)$ subtended at x by $y_i(t)$ and $y_j(t)$, evaluated counterclockwise from the $y_i(t)$ to $y_j(t)$. To simplify the notation, we let

$$\beta_i(t) := \beta_i^{i\%N+1}(t). \quad (5)$$

In other words, $\beta_i(t)$ denotes the counterclockwise angle between agent i and the agent with the consecutive index $(i+1)$ in a circular fashion. Finally, we denote as $\varphi_i(t)$ the bearing of the target with respect to the position $y_i(t)$ of agent i . Namely, we set

$$\varphi_i(t) := \frac{x - y_i(t)}{\|x - y_i(t)\|}. \quad (6)$$

Note that, for any two agents i and j , we have $\beta_i^j(t) = \angle(\varphi_i(t), \varphi_j(t))$. The bearing vector is well defined if and only if $y_i(t) \neq x$. Therefore, we need to make sure that our control law guarantees that the agents do not travel indefinitely close to the target. Also, the initial positions of the agents must not coincide with the position of the target.

Assumption 1: For all agents $i \in \{1, \dots, N\}$, $y_i(0) \neq x$. ■

Our objective is to design a decentralized control action $u_i(t)$, with $i \in \{1, \dots, N\}$ such that

$$\lim_{t \rightarrow \infty} \|x - y_i(t)\| = D^*, \quad i \in \{1, \dots, N\}, \quad (7a)$$

$$\lim_{t \rightarrow \infty} (\beta_i(t) - \beta_j(t)) = 0, \quad i, j \in \{1, \dots, N\}, \quad (7b)$$

where $D^* > 0$ is a desired distance. Note that, by Lemma 1 the control objective (7) implies that the agents tend to become equally spaced on the circle with center x and radius D^* .

To reach the control objective (7), we assume that the agents can measure the bearing of the target and that they can exchange data over a shared repository hosted on a cloud server. However, since the bearing measurements and the exchange of information over the cloud rely on wireless communication, we do not assume that they can be executed continuously. Instead, we model these communication instances

i	$\tau_{i,l_i(t)}$	$\hat{\varphi}_i(\tau_{i,l_i(t)})$	$\omega_i(\tau_{i,l_i(t)})$	$\tau_{i,l_i(t)+1}$
1	$\tau_{1,l_1(t)}$	$\hat{\varphi}_1(\tau_{1,l_1(t)})$	$\omega_1(\tau_{1,l_1(t)})$	$\tau_{1,l_1(t)+1}$
2	$\tau_{2,l_2(t)}$	$\hat{\varphi}_2(\tau_{2,l_2(t)})$	$\omega_2(\tau_{2,l_2(t)})$	$\tau_{2,l_2(t)+1}$
\vdots	\vdots	\vdots	\vdots	\vdots
N	$\tau_{N,l_N(t)}$	$\hat{\varphi}_1(\tau_{N,l_N(t)})$	$\omega_1(\tau_{N,l_N(t)})$	$\tau_{N,l_N(t)+1}$

TABLE I: Information contained in the cloud at a generic time instant.

as instantaneous events, that are triggered by appropriately designed conditions. We let $t_{i,k}$ denote the time when agent i measures the bearing of the target for the k -th time. Similarly, we let $\tau_{i,k}$ denote the time when agent i accesses the cloud repository for the k -th time.

The distributed control law that we propose takes the following form:

$$\omega_{i,k} = \kappa(\alpha + \hat{\beta}_{i,k}), \quad (8a)$$

$$u_i(t) = D^* \omega_{i,k} \hat{\varphi}_i(t)^\perp, \quad t \in (\tau_{i,k}, \tau_{i,k+1}), \quad (8b)$$

where κ and α are positive constants, while $\hat{\varphi}_i(t)$ and $\hat{\beta}_{i,k}$ are local estimates of $\varphi_i(t)$ and $\beta_i(t)$, respectively. As we shall see in the following, $\omega_i > 0$ represents the angular speed with which agent i rotates around its current estimate of the position of the target.

For the control law (8) to be completely defined, we need to specify how the estimates $\hat{\varphi}_i(t)$ and $\hat{\beta}_{i,k}$ are computed.

A. Estimate of the bearing vectors

The estimates $\hat{\varphi}_i(t)$ of the bearing vectors are obtained as follows. For all $t \in (t_{i,k}, t_{i,k+1})$, we let

$$\hat{x}_i(t) = y_i(t_{i,k}) + D^* \varphi_i(t_{i,k}), \quad (9a)$$

$$\hat{\varphi}_i(t) = \frac{\hat{x}_i(t) - y_i(t)}{\|\hat{x}_i(t) - y_i(t)\|}. \quad (9b)$$

The estimation law (9) can be interpreted as follows: between two consecutive bearing measurements $\varphi_i(t_{i,k})$ and $\varphi_i(t_{i,k+1})$, agent i assumes that the target is located on the direction defined by the most recent bearing measurement $\varphi_i(t_{i,k})$, at a distance D^* from the position $y_i(t_{i,k})$ of the agent at the measurement time. Note that, differentiating (9b) and using (4) and (8), we find easily

$$\dot{\hat{\varphi}}_i(t) = \omega_{i,k} \hat{\varphi}_i(t)^\perp, \quad t \in (\tau_{i,k}, \tau_{i,k+1}), \quad t \notin \{t_{i,k}\}_{k \in \mathbb{N}}. \quad (10)$$

B. Estimate of the counterclockwise angles

To define the estimates of the counterclockwise angles, we need first to define the pattern by which the agents exchange information over the cloud repository. To this aim, let $l_i(t)$ denote the cardinality of the most recent access to the cloud of agent i before time t . In other words, let

$$l_i(t) = \max\{k \in \mathbb{N} : \tau_{i,k} < t\}. \quad (11)$$

The information contained in the cloud repository at a generic time instant is illustrated in Table I. From Table I, we can see that each row contains information about one agent. Namely, the i -th row contains: the time $\tau_{i,l_i(t)}$ of the most recent access of agent i to the cloud; the estimated bearing

vector $\hat{\varphi}_i(\tau_{i,l_i(t)})$ of agent i at said time; the angular velocity $\omega_i(\tau_{i,l_i(t)})$ applied to agent i at said time, and; the time of the following access $\tau_{i,l_i(t)+1}$. Whenever agent i accesses the cloud repository, it downloads the information relative to agent $i\%N+1$, and computes the angular speed $\omega_{i,k}$ and the time $\tau_{i,k+1}$ of its next access; then, it uploads the quadruple $(\tau_{i,k}, \hat{\varphi}_i(\tau_{i,k}), \omega_i(\tau_{i,k}), \tau_{i,k+1})$. This quadruple overwrites the corresponding row in the repository, so that the cloud contains updated information about agent i . In this way, the amount of information contained in the repository does not grow over time, and the capacity of the repository can be proportional to the number of agents in the network. The estimates of the counterclockwise angles are generated as follows. Let

$$\hat{\beta}_{i,k}^j = \angle(\hat{\varphi}_i(\tau_{i,k}), \text{rot}(\varphi_j(\tau_{j,l_j(\tau_{i,k})}), \omega_j(\tau_{j,l_j(\tau_{i,k})})(\tau_{i,k} - \tau_{j,l_j(\tau_{i,k})}))), \quad (12)$$

and, in particular, $\hat{\beta}_{i,k} = \hat{\beta}_{i,k}^{i\%N+1}$.

Simply put, $\hat{\beta}_{i,k}$ is an estimate of $\beta_i(\tau_{i,k})$ based on the data available in the cloud. (In fact, $\hat{\beta}_{i,k} = \beta_i(\tau_{i,k})$ if the estimates of the bearing vectors contained in the cloud coincide with the actual bearing vectors at the access times.) Note that $\hat{\beta}_{i,k}$ can be computed by agent i using only information downloaded from the cloud at time $\tau_{i,k}$.

Remark 1: In order to compute $\hat{\beta}_{i,k}$, agent i needs to download only the information related to agent $i\%N+1$. ■

To conclude the definition of our control law, we must now give the rules that trigger the measurements of the bearing vector and the accesses to the cloud. In the following Sections IV and V we illustrate appropriate scheduling rules to attain the control objective (7).

IV. TRIGGERING OF THE BEARING MEASUREMENTS

The bearing measurements are scheduled according to the following recursive rule:

$$t_{i,k+1} = \inf\{t \geq t_{i,k} : \angle(\hat{\varphi}_i(t), \varphi(t_{i,k})) \geq \pi/2\}. \quad (13)$$

Theorem 1: Consider a generic agent i with kinematics (4) and under the control law (8). Let the bearing measurements be scheduled as prescribed by (13). Then, under Assumption 1, we have $y_i(t) \neq x$ for all $t \geq 0$ and $\lim_{t \rightarrow \infty} \hat{x}_i(t) = x$. Moreover, the interval $t_{i,k+1} - t_{i,k}$ between two consecutive bearing measurements is lower-bounded. ■

Proof: First note that $\hat{\varphi}_i(t)$ rotates with angular speed $\omega_{i,k}$, which is upper-bounded by $\kappa(\alpha + 2\pi)$. Hence, the interval between two consecutive measurements is lower-bounded by $(\pi/2)/(\alpha + 2\pi)$. Let us consider the discrete time system obtained by integrating (4) over the time interval between two consecutive measurements $t_{i,k}$ and $t_{i,k+1}$ of the bearing vector $\varphi_i(t)$. Substituting in (4) the expression of $u_i(t)$ given in (8), and noting that $\alpha > 0$ by design, and $\beta_i(t)$ is nonnegative by definition, we have that, at any time instant, agent i rotates about its current estimate of the target

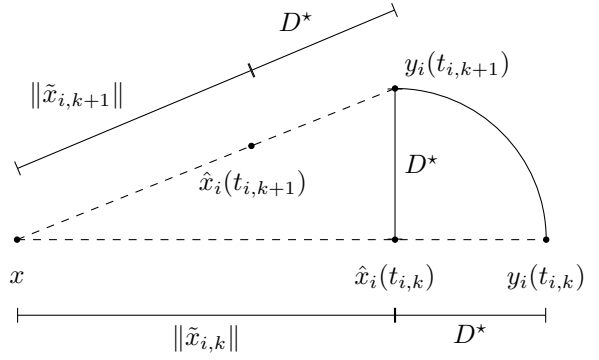


Fig. 1: Illustration of eq. (16)

with strictly positive (counterclockwise) angular velocity, and thus $t_{i,k+1}$ is finite for all $k \in \mathbb{N}$. Hence,

$$y_i(t_{i,k+1}) = \hat{x}_i(t_{i,k}) + (y_i(t_{i,k}) - \hat{x}_i(t_{i,k}))^\perp. \quad (14)$$

Substituting in (14) the expressions of $\hat{x}_i(t)$ and $\hat{\varphi}_i(t)$ in (9), we obtain

$$y_i(t_{i,k+1}) = y_i(t_{i,k}) + \varphi_i(t_{i,k})D^* - \varphi(t_{i,k})^\perp D^*. \quad (15)$$

Now, consider the estimation error $\tilde{x}_{i,k} := \hat{x}_i(t_{i,k+1}) - x$, and the distance between the agent and the target $z_{i,k} = y_i(t_{i,k}) - x$. From Figure 1, and leveraging (15), we see that

$$\|\tilde{x}_{i,k}\| = \sqrt{\|\tilde{x}_{i,k}\|^2 + D^{*2}} - D^* \quad (16)$$

$$\|z_{i,k+1}\| = \sqrt{\|\tilde{x}_{i,k}\|^2 + D^{*2}} \quad (17)$$

From (17), we can see that, as long as $z_{i,0} \neq 0$, we have $z_{i,k} \neq 0$ for all $k \in \mathbb{N}$. Since under Assumption 1 we have $z_{i,0} = y_i(0) - x \neq 0$, we can conclude that $z_{i,k} = y_i(t_{i,k}) - x \neq 0$ for all $k \in \mathbb{N}$. Moreover, subtracting $\|\tilde{x}_{i,k}\|$ from both sides in (16), we have

$$\|\tilde{x}_{i,k+1}\| - \|\tilde{x}_{i,k}\| = \sqrt{\|\tilde{x}_{i,k}\|^2 + D^{*2}} - (\|\tilde{x}_{i,k}\| + D^*). \quad (18)$$

From (18), we can see that $\|\tilde{x}_{i,k+1}\| - \|\tilde{x}_{i,k}\| \leq 0$, and $\|\tilde{x}_{i,k+1}\| - \|\tilde{x}_{i,k}\| = 0$ if and only if $\|\tilde{x}_{i,k}\| = 0$. Hence, from LaSalle's invariance principle for discrete systems [17], we know that $\|\tilde{x}_{i,k}\|$ must converge to zero for $k \rightarrow \infty$. Hence, $\hat{x}_i(t)$ must converge to x , which concludes the proof. ■

Theorem 1 ensures the control law in (8) is capable of achieving the control goal in (7a), that is, ensuring the agents circumnavigate the target. This is evident if the reader recalls that the agents are modeled as simple integrators, and thus $u_i(t)$ is the velocity of agent i . Then, if $\hat{x}_i(t)$ converges to x , from (6) and (9b) we have that $\hat{\varphi}_i$ tends to φ_i and thus from (8) we have that agent i rotates about x at a distance D^* with positive (counterclockwise) angular velocity.

V. TRIGGERING OF THE ACCESS TO THE CLOUD

Having taken care of the circumnavigation task, we can now turn our attention to ensuring that the agents achieve a balanced circular formation; that is, the control goal in (7b).

Achieving such a goal is intrinsically tied to the selection of the rule that triggers accesses to the cloud.

We start by noting that accessing the cloud to download the information to compute $\hat{\beta}_{i,k}$ only affects the goal of achieving a balanced circular formation. By Theorem 1, we know all agents converge to the circle of radius D^* centered in the position of the target, regardless of the particular values assumed by the angular speed $\omega_{i,k}$. Therefore, we can as well study the triggering of the cloud accesses at an instant. In this case, from (8) we have that the planar velocities $u_i(t)$ of the agents are substantially tangential to the circle. Hence, we can reason directly on the angular speeds, which we have denoted as $\omega_{i,k}$. Note that, since $\hat{\beta}_{i,k} \geq 0$ by definition, we have that each agent circumnavigates the target counterclockwise.

Without loss of generality, let $t \in (\tau_{i,k}, \tau_{i,k+1})$, and let $j = i\%N + 1$, $h = l_j(\tau_{i,k})$. With this notation, when agent i accesses the cloud at time $\tau_{i,k}$, it downloads the quadruple $(\tau_{j,h}, \hat{\varphi}_j(\tau_{j,h}), \omega_{j,h}, \tau_{j,h+1})$. When all agents have reached the desired circle, we have $\hat{\varphi}_i(t) = \varphi_i(t)$ for all $i \in \{1, \dots, N\}$. Hence, from (10), we have $\hat{\beta}_i(t) = \omega_{j,l_j(t)} - \omega_{i,k}$, which, using (8a), becomes

$$\hat{\beta}_i(t) = \kappa(\hat{\beta}_{j,l_j(t)} - \hat{\beta}_{i,k}). \quad (19)$$

Denoting $e_i(t) = \hat{\beta}_{i,l_i(t)} - \beta_i(t)$, (19) becomes

$$\hat{\beta}_i(t) = \kappa(\beta_j(t) - \beta_i(t)) + \kappa(e_j(t) - e_i(t)). \quad (20)$$

From (20), it is clear that $\beta_i(t)$ evolves according to a diffusive coupling (over a ring graph) with additive disturbances $e_i(t)$. Since a ring graph has a spanning tree, the variables $\beta_i(t)$ will reach consensus as long as the additive disturbances vanish quickly enough.

Rule A

Our first rule is inspired by the results in [14], and takes explicitly into account that part of the error $e_i(t)$ arises from the possible existence, in the time interval $(\tau_{i,k}, \tau_{i,k+1})$, of a smaller interval $(\tau_{j,h+1}, \tau_{i,k+1})$ in which the angular speed of agent j is different from $\omega_{j,h}$ downloaded by agent i from the cloud. This indeed happens if $\tau_{j,h+1} \in (\tau_{i,k}, \tau_{i,k+1})$. Namely, Rule A is defined by the following set of equations:

$$\tau_{i,k+1} = \inf\{t > \tau_{i,k} : \sigma_{i,k}(t) \geq \varsigma_i(t)\}, \quad (21)$$

where

$$\begin{aligned} \sigma_{i,k}(t) = & \kappa \left(\|\omega_{i,k}(t - \tau_{i,k}) - \omega_{j,h}(\min\{t, \tau_{j,h+1}\} - \tau_{i,k})\| \right. \\ & \left. + \int_{\tau_{j,h+1}}^{\max\{t, \tau_{j,h+1}\}} \mu_j(\tau) d\tau \right), \end{aligned} \quad (22)$$

$$\varsigma_i(t) = \varsigma_{i,0} e^{-\lambda t}, \quad (23)$$

$$\mu_i(t) = \gamma_i \eta(t) + \varsigma_i(t), \quad (24)$$

$$\begin{aligned} \eta(t) = & \delta \left(\eta_0 e^{-\kappa \lambda t} + \right. \\ & \left. + \|B_{\mathcal{T}}\| \int_0^t e^{-\kappa \lambda (t-\tau)} \|\varsigma(\tau)\| d\tau \right), \end{aligned} \quad (25)$$

$$\eta_0 \geq \|B_{\mathcal{T}}^\top \beta(0)\|, \quad (26)$$

$$\gamma_i = \kappa \|\{B_{\mathcal{T}}^\circ + B_c^\circ (B_{\mathcal{T}}^\dagger B_C)^\top\}_i\|, \quad (27)$$

and where $\varsigma(t) = [\varsigma_1(t), \dots, \varsigma_N(t)]^\top$, $\beta(t) = [\beta_1(t), \dots, \beta_N(t)]^\top$, $\varsigma_{i,0} > 0$ for all $i \in \{1, \dots, N\}$, $\lambda = \min(\text{real}(\text{eig}(L_r)))$, $\lambda_c \in (0, \lambda)$, and $\delta = \|V\| \|V^{-1}\|$, with V being the matrix of the eigenvectors of L_r .

Taken altogether, (21)–(27) indicate that an access to the cloud performed by agent i is scheduled as soon as $\sigma_{i,k}(t)$ reaches the threshold $\varsigma_i(t)$. The rationale behind Rule A is that $\sigma_{i,k}(t)$ constitutes an upper bound for $e_i(t)$ that agent i can compute at time $\tau_{i,k}$. Hence, by ensuring that $\sigma_{i,k}(t) \leq \varsigma_i(t)$, one automatically ensures that $e_i(t)$ converges to zero exponentially, with a rate of at least λ_c . It is also possible to show that, under this scheduling rule, the interval between two consecutive cloud accesses is lower-bounded.

Lemma 2: The scheduling proposed in (21)–(27) guarantees that the dynamics of $\beta_i(t)$ in (20) satisfy the control objective (7b). Moreover, the interval $\tau_{i,k+1} - \tau_{i,k}$ between two consecutive cloud accesses is lower-bounded. ■

The proof of Lemma 2 is similar to that of Theorem V.2 in [14], where the angles $\beta_i(t)$ are used as the consensus variables and is therefore omitted for brevity.

Rule B

In section VI we will provide a formal proof ensuring the agents achieve the control objective in (7b) under Rule A. However, as we will numerically show in section VII, this comes at the price of having the agents access the cloud at a relatively high frequency. To reduce such frequency we propose the following heuristic rule triggering the accesses to the cloud:

$$\tau_{i,k+1} := \inf\{t > \tau_{i,k} : (t - \tau_{i,k})\omega_{i,k} = \hat{\beta}_{i,k}\} \quad (28)$$

Roughly speaking, Rule B prescribes that agent i accesses the cloud whenever it sweeps, around its current estimate of the target \hat{x} , an angle equal to $\hat{\beta}_{i,k}$.

VI. MAIN RESULT

The main theoretical result of this paper can be formalized as follows.

Theorem 2: Consider a network of agents with kinematics (4) and under the control law (8). Let the bearing measurements $t_{i,k}$ be scheduled according to (13) and the cloud accesses $\tau_{i,k}$ be scheduled according to (21)–(27). Then, the agents reach the control objective in (7b). Moreover, the interval $t_{i,k+1} - t_{i,k}$ between two consecutive bearing measurements and the interval $\tau_{i,k+1} - \tau_{i,k}$ between two consecutive accesses to the cloud are lower-bounded. ■

Proof: The proof follows directly from Theorem 1 and Lemma 2 and is omitted here for the sake of brevity. ■

VII. SIMULATIONS AND PRELIMINARY EXPERIMENTS

A. Comparison between Rule A and Rule B

To demonstrate the effectiveness of our approach, for each of the two rules proposed in Section V, we carried out extensive numerical simulations involving a set of $N = 5$

agents. Specifically, we performed 100 simulations of the duration of $T = 25s$ each. As the asymptotic convergence of the agents to the circle of radius D^* centered in x is guaranteed regardless of the rule triggering the accesses to the cloud we assumed the agents have already converged onto the circle surrounding the target and started all simulations accordingly. Therefore, we focused on their angular speed $\omega_{i,k} = \kappa(\alpha + \hat{\beta}_{i,k})$ and, throughout our simulations, we have set $\kappa = 1$ and $\alpha = 0.5$. To ensure the comparison between Rule A and Rule B is independent of the initial conditions, we randomly selected a set of 100 different values of the vector $y(0)$ and assigned the i -th initial condition to both the i -th simulation performed under Rule A, and the i -th simulation performed under Rule B. As for the parameters of Rule A, we have $\gamma_i = 1$ $i = 1, \dots, 4$, $\gamma_5 = 2$, and $\lambda = \min(\text{real}(\text{eig}(\mathcal{L}_r))) = 0.69$. Moreover, we set $\lambda_\zeta = 0.68$, $\varsigma_{i,0} = 10$, $\eta_0 = 2.12$, and $\delta = 2.24$.

In all simulations performed under Rule B we have observed convergence of the angles β_i $i = 1, \dots, n$, in a time t_c computed as the first time instant such that an agent accesses the cloud and $|\beta_i(t) - 2\pi/N| \leq 0.01(2\pi/N)$. Having provided numerical evidence demonstrating the effectiveness of Rule B in allowing the agents to achieve the control goal in (7b), we next compare the performances of the two rules regulating the accesses to the cloud. We start by highlighting that, as is the case for the two simulations with identical initial conditions shown in Figs. 2 and 3, Rule A ensures faster convergence than Rule B. Namely, we find that the observed average convergence time under the former, $\langle t_{c,A} \rangle$ is equal to $6.32s$, while $\langle t_{c,B} \rangle = 14.98$. The results of an hypothesis test on the difference between $\langle t_{c,A} \rangle$ and $\langle t_{c,B} \rangle$ confirms the observed difference in the convergence time is statistically significant. Conversely, we find that under Rule A, the total number of accesses to the cloud before convergence is achieved averages at 316 accesses against the average of 103 observed under Rule B. Again, we find that this difference is statistically significant. Moreover, we observe that the frequency of the total accesses to the cloud under Rule A averages to 50Hz, while the frequency under Rule B to 7Hz. Finally, as expected, we observe a substantial difference in the frequency of the accesses after convergence is achieved. Namely, under Rule A such frequency averages at 203Hz, while under Rule B it averages at 7Hz.

B. ROS Implementation and experimental evaluation

To demonstrate the use of the proposed control algorithm when the agents have to approach the target, we run a simulation in the ROS [18] environment, where each agent is simulated as a different ROS node. Each agent is a ROS node itself and interacts with an additional ROS node that represents the cloud repository by means of ROS messages and services. In this simulation, we consider $N = 5$ agents, with $\alpha = 0.4$ and $\kappa = 0.2$, and the cloud accesses are scheduled according to Rule B. The results of the simulation are summarized in Figure 4, where we can see that the agents converge to the desired circle around the target while reaching a regular formation, and that each agent accesses

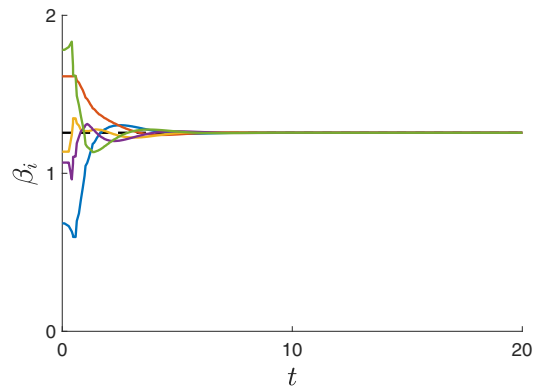


Fig. 2: Plot of $\beta_i(t)$ $i = 1, \dots, 5$ for a simulation performed under Rule A. The black dashed line denotes the target value for $\beta_i(t)$, that is $2\pi/5$.

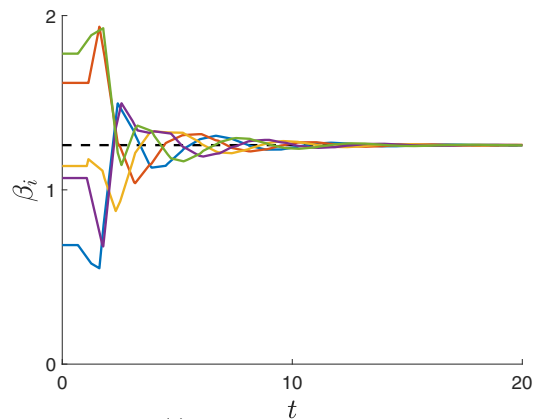


Fig. 3: Plot of $\beta_i(t)$ $i = 1, \dots, 5$ for a simulation performed under Rule B. The black dashed line denotes the target value for $\beta_i(t)$, that is $2\pi/5$.

the cloud at approximately 0.3Hz. Although the ROS implementation requires some overhead, it presents the following significant advantages: as each agent is implemented as a ROSnode, the distributed nature of the algorithm is reproduced in the simulation; the code used to implement the controller in the simulation can be re-used tout-court in the experimental implementations.

In order to further test the proposed control strategies, we have performed some experiments in the flying arena of the Smart Mobility Lab at KTH. In the experiments, the networked agents are three Bitcraze CrazyFlye 2.0, which are open-source and open-hardware nano quadcopters. We implemented the control and communication features in ROS, exploiting its modularity and robot-aimed tools. We used a custom USB radio dongle called CrazyRadio through which we sent setpoints consisting in thrust and attitude (roll, pitch and yaw angles); these were then transformed into commands for the four propellers by an on-board micro-controller. The real-time positions and orientations of quadrotors are tracked by a motion capture system with an update rate of 100Hz, capable of detecting the position of some reflective spherical markers that we applied on the quadcopters.

The logical structure of the control architecture is de-

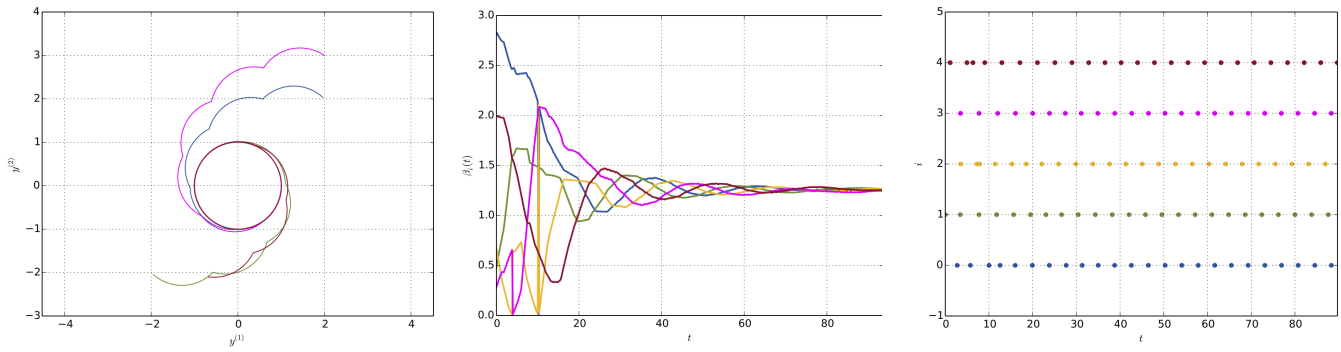


Fig. 4: Results of the simulation described in Section VII-B. From left to right: trajectories of the agents; angles $\beta_i(t)$, cloud access times for each agent.

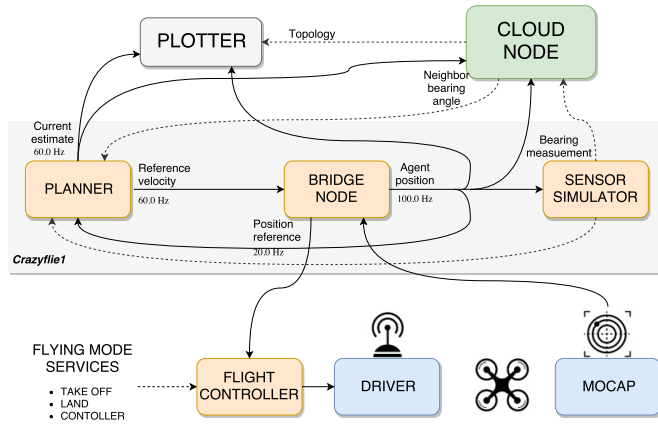


Fig. 5: Logical structure of the implemented control solutions for experiments. The dashed arrows correspond to ROS services, the continue ones correspond to ROS topics. The orange nodes represent features that are replicated for each agent of the network; the green node is the cloud and is able to get information from every network agent; the plotter, in white, gets informations both about the agents and the designed task. It represents the interface for the human user. The blue nodes are related to the actuation of the control and to the acquisition of the feedback from the ground.

scribed in Figure 5. As a preliminary implementation, we integrate the velocity command $u_i(t)$ generated by the controller according to (8), producing a certain number of intermediate goal positions that are fed to a PID flight controller, as shown in figure (5). In these experiments, both Rule A and Rule B were able to achieve the desired target behavior. Convergence was achieved despite some limitations of the experimental set-up, whose improvement will be the subject of future work. Moreover, the PID controller used on-board as a flight controller could not always cope with the unmodelled dynamics leading to possible instabilities that made the experimental results worse than the numerical ones. A video of a representative experiment is on Zenodo [19].

REFERENCES

- [1] I. Shames, S. Dasgupta, B. Fidan, and B. D. O. Anderson, "Circumnavigation Using Distance Measurements Under Slow Drift," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 889–903, 2012.
- [2] M. Deghat, I. Shames, B. D. O. Anderson, and L. Fellow, "Localization and Circumnavigation of a Slowly Moving Target Using Bearing Measurements," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2182–2188, 2014.
- [3] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.
- [4] J. A. Marshall, M. E. Broucke, and B. A. F. Macdonald, "Pursuit formations of unicycles," *Automatica*, vol. 42, pp. 3–12, 2006.
- [5] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion: All-to-all communication," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 811–824, 2007.
- [6] T.-H. Kim and T. Sugie, "Cooperative control for target-capturing task based on a cyclic pursuit strategy," *Automatica*, vol. 43, pp. 1426–1431, 2007.
- [7] I. Shames, B. Fidan, and B. D. O. Anderson, "Close target reconnaissance with guaranteed collision avoidance," *International Journal of Robust and Nonlinear Control*, vol. 21, pp. 1823–1840, 2011.
- [8] J. O. Swartling, I. Shames, K. H. Johansson, and D. V. Dimarogonas, "Collective Circumnavigation," *Unmanned Systems*, vol. 2, no. 3, pp. 219–229, 2014.
- [9] A. Boccia, A. Adaldo, D. V. Dimarogonas, M. di Bernardo, and K. H. Johansson, "Tracking a mobile target by multi-robot circumnavigation using bearing measurements," *IEEE Conference on Decision and Control*, 2017.
- [10] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An Introduction to Event-triggered and Self-triggered Control," in *IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2012, pp. 3270–3285.
- [11] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.
- [12] S. L. Bowman, C. Nowzari, and G. J. Pappas, "Coordination of multi-agent systems via asynchronous cloud communication," in *IEEE Conference on Decision and Control*, 2016, pp. 2215–2220.
- [13] R. Patel, P. Frasca, J. W. Durham, R. Carli, and F. Bullo, "Dynamic partitioning and coverage control with asynchronous one-to-base-station communication," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 5589–5594, 2016.
- [14] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Multi-Agent Trajectory Tracking with Event-Triggered Cloud Access," in *IEEE Conference on Decision and Control*, 2016, pp. 2207–2214.
- [15] A. Adaldo, D. V. Dimarogonas, and K. H. Johansson, "Cloud-supported effective coverage of 3D structures," in *European Control Conference*, 2018.
- [16] D. Zelazo, Amirreza Rahmani, and Mehran Mesbahi, "Agreement via the edge laplacian," in *IEEE Conference on Decision and Control*, 2007, pp. 2309–2314.
- [17] J. P. LaSalle, *The Stability and Control of Discrete Processes*, 1986.
- [18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, "ROS: an open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation*, 2009.
- [19] D. Mariniello, A. Adaldo, D. V. Dimarogonas, and K. H. Johansson, "Cooperative circumnavigation with event-triggered bearing measurements." [Online]. Available: <https://doi.org/10.5281/zenodo.1067181%0A>