Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Self-triggered adaptive control for multi-agent systems with timed constraints and connectivity maintenance ⋆

Maria Guinaldo * **Dimos V. Dimarogonas** **

* *Department of Computer Science and Automatic Control, UNED, Madrid, Spain (e-mail: mguinaldo@dia.uned.es).*
** *Division of Decision and Control, School of EECS, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: dimos@kth.se)*

Abstract: This paper presents a distributed control strategy for a multi-agent system commanded by a set of leaders that has to accomplish a high-level plan consisting of a sequence of tasks specified by a state-space region and a timed constraint. The agents are also subject to relative-distance constraints with its neighbors. The solution consists in an adaptive distributed mechanism to update the feedback gains for the leader agents, which is executed following a self-triggered algorithm. The results show how the proposed approach provides less conservative results than if feedback gains are held constant, and are illustrated with a simulation example.

Keywords: Multi-agent systems, distributed control, self-triggering, adaptive control.

1. INTRODUCTION

In the recent years, cooperative control for multi-agent systems has attracted the attention of both the robotics and control communities, due to its wide applications (Ren, 2007; Lafferriere et al., 2005; Hong et al., 2006). In general, each agent serves to accomplish a global objective or fulfill a simple local goal such as reachability. However, in practice, a group of agents encounters the request of a sequence of tasks. Furthermore, deadline constraints on the completion of each task is a common requirement (Gombolay et al., 2018). One common approach to express temporal specifications relies on Linear Temporal Logic (LTL), and it has been applied to multi-agent systems (Nikou et al., 2016). LTL allows proving the accomplishment of the high-level tasks but it lies on the assumption that simple control laws, e.g. turn-and-forward switching control (Guo and Zavlanos, 2017), can be used, leaving out the possibility of reusing advanced control strategies designed in the single task case and applying their properties.

In this paper, we propose a novel strategy for the control of a multi-agent systems, modeled as single-integrators, in which the aim of the system is to complete a *high level plan*, consisting of a sequence of cooperative tasks that need to be accomplished before some given deadlines. This plan is partially known by a subgroup of the agents (called *leaders*). Moreover, the agents do not have communication capabilities but are equipped with low-cost hardware to detect close objects (other robots) that allow the measurement of relative distances. This constraint requires that neighboring agents maintain geographical closeness, and this problem is similar to maintaining connectivity in robotic networks (Ji and Egerstedt, 2007). Hence, the leaders have to command the rest of the agents to the objective to reach it on time, but at the same time ensure that the whole group remains connected. To achieve this goal, a distributed self-triggered algorithm (Heemels et al., 2012) is proposed to tune the parameters of the leaders' controllers. Based on local measurements (distance and remaining time to reach the objective, and the state of the neighboring agents) that are taken only a discrete instants of time which are computed from the last measurements acquired, the leaders calculate an upper bound for the feedback gain that ensures connectivity. Self-triggered control allows reducing the use of computational resources since continuous monitoring is avoided. Hence, the contributions of the paper in hand can be summarized as the design of a self-triggered algorithm to update the distributed controllers in order to achieve each control objective that corresponds to any individual task in the high level plan, while guaranteeing the maintenance of relative-distance constraints in each team of robots. This paper partially builds on our earlier work Guinaldo and Dimarogonas (2017), where all the agents are aware of the planning, feedback gains are held constant, and the connectivity maintenance is not considered.

2. PRELIMINARIES

Consider a set $\mathcal{N}$ of N agents. The topology of the multi-agent system can be modeled as a static undirected graph $\mathcal{G}$. This section reviews some facts from algebraic graph theory Godsil and Royle (2001). The graph $\mathcal{G}$ is described by the set of agent-nodes $\mathcal{V}$ and the set of edges $\mathcal{E}$. For each agent i, $\mathcal{N}_i$ represents the neighborhood of i, i.e., $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. A path graph of N vertices, denoted

⋆ The work of the first author was supported by the Spanish Ministry of Science, Innovation and Universities under Projects DPI2017-84259-C2-2-R and CICYT RTI2018-094665-B-I00. The work of second author was supported by the H2020 ERC Starting Grant BUCOPHSYS, the EU H2020 Co4Robots project, the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR) and the Knut och Alice Wallenberg Foundation (KAW).

by P_N, is a graph whose vertices can be listed in the order v_1, v_2, ..., v_N such that the edges are (v_j, v_{j+1}) where $j = 1, 2, , N-1$. A path is a particularly simple example of a tree, since no vertex has degree 3 or more. Assume that the edges has been labeled as e_k and arbitrarily oriented. Then the incidence matrix $E(\mathcal{G}) = [\varepsilon_{ik}]$ is defined as $\varepsilon_{ik} = -1$ if v_i is the tail of the edge e_k, $\varepsilon_{ik} = 1$ if v_i is the head of e_k, and $\varepsilon_{ik} = 0$ otherwise. The Laplacian matrix $L(\mathcal{G}) \in \mathbb{R}^{N\times N}$ of a network of agents is defined as $L(\mathcal{G}) = E(\mathcal{G})E(\mathcal{G})^\top$. The Laplacian matrix $L(\mathcal{G})$ is positive semidefinite, and if $\mathcal{G}$ is connected and undirected, then $0 = \lambda_1(\mathcal{G}) < \lambda_2(\mathcal{G}) \leq \cdots \leq \lambda_N(\mathcal{G})$, where $\{\lambda_j(\mathcal{G})\}$ are the eigenvalues of $L(\mathcal{G})$. Both matrices $E(\mathcal{G})$ and $L(\mathcal{G})$ can be simply denoted by E and L, respectively, when it is clear from the context.

The edge Laplacian is defined as $L_e(\mathcal{G}) = E(\mathcal{G})^\top E(\mathcal{G})$, and has the following properties (Zelazo et al., 2007): 1) The non-zero eigenvalues of L_e are equal to the non-zero eigenvalues of L; 2) the rank of L_e depends only on the number of connected components; 3) the null space of L_e, $\mathcal{N}(L_e)$, depends on the number of cycles in the graph and it holds that $\mathcal{N}(L_e) = \mathcal{N}(E)$. Furthermore, $\mathcal{N}(E)$ is spanned by all the linearly independent *signed path vectors* [1] corresponding to the cycles of E; 4) if $\mathcal{G}$ is a spanning tree, then L_e has no zero eigenvalues and, hence, $\mathcal{N}(L_e) = \emptyset$. The next lemma follows from Zelazo et al. (2007).In the following, $\|\cdot\|$ denotes the Euclidean norm for vectors and the induced 2-norm for matrices, respectively, while $|\cdot|$ is the cardinality of a set.

Lemma 1. Suppose $L_e \in \mathbb{R}^{N_e\times N_e}$ is the edge Laplacian of an undirected connected graph $\mathcal{G}$. Then, for all $t \geq 0$ and all vectors $z \in \mathbb{R}^{N_e}$ with $z = E^\top x$ and $x \in \mathbb{R}^N$, it holds that $\|e^{-L_e t} z\| \leq e^{-\lambda_2(\mathcal{G})t}\|z\|$.

Proof. Let us assume that the graph $\mathcal{G}$ has n_c independent cycles. Then, the multiplicity of the zero eigenvalues of L_e is n_c. Since L_e is symmetric, the eigenvectors of L_e can always be chosen such that they form and orthonormal basis T and it holds that $L_e = T \cdot \mathrm{diag}(0, \ldots, 0, \lambda_2, \ldots, \lambda_N) \cdot T^\top$, where the first n_c vectors in T correspond to the n_c zero eigenvalues, and the rest corresponds to the eigenvalues $\lambda_2, \ldots, \lambda_N$ of L. Then $e^{-L_e t} = T\ \mathrm{diag}(1, \ldots, 1, e^{-\lambda_2 t}, \ldots, e^{-\lambda_N t})T^\top$. For $z = E^\top x \in \mathbb{R}^{N_e}$, it holds $e^{-L_e t} z = T\ \mathrm{diag}(1, \ldots, 1, 0, \ldots, 0)T^\top z + T\ \mathrm{diag}(0, \ldots, 0, e^{-\lambda_2 t}, \ldots, e^{-\lambda_N t})T^\top z$. The first term is 0 using the property 3) described above Zelazo et al. (2007) and, hence $\|e^{-L_e t} z\| = \|T\ \mathrm{diag}(0, \ldots, 0, e^{-\lambda_2 t}, \ldots, e^{-\lambda_N t})T^\top z\|$. Note that $\|A \cdot B\| \leq \|A\|\|B\|$ for any two matrices and that $\|T\| = \|T^\top\| = 1$ for an orthonormal basis, then it follows $\|e^{-L_e t} z\| \leq \|\mathrm{diag}(0, \ldots, 0, e^{-\lambda_2 t}, \ldots, e^{-\lambda_N t})\|\|z\| = e^{-\lambda_2 t}\|z\|$, which completes the proof. □

3. PROBLEM FORMULATION

3.1 Planning description

For a multi-agent system, we assume the existence of a high level plan ϕ consisting of a sequence of cooperative tasks. This plan is partially known by a subgroup of the agents (called *leaders* and denoted by $\mathcal{L} \in \mathcal{N}$), whereas the rest of the agents do not have a priori knowledge of the planning and follow the leaders. We can denote ϕ as a sequence of $N_{task} \in \mathbb{N}_{>0}$ objectives $\phi_1, \phi_2, ..., \phi_{N_{task}}$. Once ϕ_p, $p = 1, \ldots, N_{task}-1$, is completed, the agents proceed to the next task ϕ_{p+1}. We consider that the set of agents have to visit consecutively a set of regions $\{\mathcal{B}_p\}$ defined as

$$\mathcal{B}_p = \{y \in \mathbb{R}^n :\ \|y - c_p\| \leq r_p\}, \tag{1}$$

and denoted as $\mathcal{B}_p = (c_p, r_p)$. The task ϕ_p is completed once all the agents lie inside the region $\mathcal{B}_p$ within some time constraints, i.e., before some deadline $T_p = t_p - t_{p,}^0$, where t_p^0 is the starting time of the task ϕ_p.

3.2 Agents model

Let the N agents obey the single-integrator dynamics:

$$\dot{x}_i(t) = u_i(t), \quad i = 1, \ldots, N, \tag{2}$$

where $x_i(t), u_i(t) \in \mathbb{R}^n$ are the state and the control inputs of agent i, respectively. For each agent of the form (2), let us consider a control law of the form

$$u_i(t) = g_i(x_i(t), \bigcup_{j\in\mathcal{N}_i} x_j(t), \pi_i), \quad i = 1, \ldots, N, \tag{3}$$

where each $g_i(\cdot)$ is locally Lipschitz and π_i represents a set of additional parameters, which may include information about the objective regions (1). The team of agents is commanded by N_{lead} leaders, which are aware of the planning description. The design of (3) has to be such that the sequence of objectives is accomplished. Then, g_i may switch in every ϕ_p. Furthermore, the agents are equipped with hardware that allows the observation of another agents' state only if their distance is below R:

$$(i, j) \in \mathcal{E} \iff \|x_i - x_j\| \leq R. \tag{4}$$

The control law should additionally guarantee that connectivity is maintained, that we define as follows:

Definition 2. The connectivity is maintained if $(i, j) \in \mathcal{E}$, that is, (4) holds, at time $t = 0$, then $(i, j) \in \mathcal{E}\ \forall t > 0$.

3.3 Problem statement

Given a team of N agents commanded by a set of leaders $\mathcal{L}$, subject to dynamics (2) and interconnected over a connected undirected $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, synthesize for each $i \in \mathcal{V}$ control laws u_i of the form (3) such that the high level plan ϕ is completed while the connectivity is maintained.

4. PROBLEM SOLUTION

The proposed solution follows five steps: 1) the design of the control law; 2) the convergence in finite time to the objective regions with the proposed control law; 3) the derivation of constraints over the feedback gains that ensure that the connectivity is maintained for all time; 4) a self-triggered algorithm for the update of the feedback gains; and 5) the accomplishment of the high level plan under certain restrictions on the initial conditions.

4.1 Control law

Two types of agents are distinguished in the network and, hence, two control laws are defined according to each role. The task ϕ_p given in Section 3.1 is achieved once all agents lie inside the region $\mathcal{B}_p$ (1) before some given deadline T_p.

[1] See Definition 3.1 (Zelazo et al., 2007)

Hence, we propose a continuous controller which includes one or two terms, depending on the role of the agent:

$$u_i(t) = \begin{cases} \sum_{j\in\mathcal{N}_i} \kappa(x_j(t)-x_i(t)) - a_i(t)(x_i(t)-c_p) & \text{if } i\in\mathcal{L} \\ \sum_{j\in\mathcal{N}_i} \kappa(x_j(t)-x_i(t)) & \text{otherwise,} \end{cases} \tag{5}$$

where $\mathcal{L}$ is the set of leaders. The gain $\kappa \in \mathbb{R}_{>0}$ is assumed to be given and constant, while the gain $a_i(t) \in \mathbb{R}_{>0}$ is to be designed such that the group remains connected and the region $\mathcal{B}_p$ is reached on time. The mechanism to update the value of a_i will be defined later, but it will follow a self-triggered policy such that a_i remains constant between two sampling instants t_k^i and t_{k+1}^i:

$$a_i(t) = a_i(t_k^i),\ t_k^i \le t < t_{k+1}^i. \tag{6}$$

Denote by x the stack vector of x_i's, $i\in\mathcal{N}$. Then, it holds

$$\dot{x} = -\kappa(L(\mathcal{G})\otimes I_n)x - (A\otimes I_n)(x - \mathbf{1}_N\otimes c_p), \tag{7}$$

where A is a diagonal matrix whose diagonal elements are defined as $A_{ii} = a_i$ if $i\in\mathcal{L}$ and $A_{ii}=0$ otherwise, where a_i is the feedback gain defined above. Moreover, the incidence matrix allows rewriting the variables of the vertex $\mathcal{V}$ in terms of the edges $\mathcal{E}$. According to the definition of the incidence matrix $E(\mathcal{G})$, we can define a vector z for the state of the edges in $\mathcal{E}$ such that $z = (E^\top(\mathcal{G})\otimes I_n)x$, and its dynamics are given by

$$\dot{z} = -\kappa(L_e\otimes I_n)z - (E^\top A\otimes I_n)(x-\mathbf{1}_N\otimes c_p), \tag{8}$$

4.2 Convergence analysis

Before presenting the main results of this section, the following lemma studies the spectral properties of the matrix $M = \kappa L + A \in \mathbb{R}^{N\times N}$, $\kappa\in\mathbb{R}_{>0}$ and $A\in\mathbb{R}^n_{\ge 0}$.

Lemma 3. The eigenvalues of the matrix M defined above $\lambda_N(M)\ge\cdots\ge\lambda_1(M)$ are lower bounded by

$$\underline{\lambda_1}(M) = \Big(\frac{N-1}{N(N-1)+(\sum_{i=1}^N a_i)/\kappa}\Big)^{N-1}(\sum_{i=1}^N a_i) > 0. \tag{9}$$

Proof. The proof is provided in the appendix.

The next proposition analyzes the convergence to each control objective represented by regions $\mathcal{B}_p$ based on the results of Lemma 3. At this point, we are not concerned about the values of $a_i > 0$ that will be designed later to ensure that the connectivity of the team is maintained.

Proposition 4. Consider the system (7). The distance to the center of the region $\mathcal{B}_p$ denoted by $\|\delta(t)\|$ converges asymptotically to zero, where $\delta(t) = x(t) - \mathbf{1}_N\otimes c_p$, and each region $\mathcal{B}_p$ is reached in finite time for any feedback gains $a_i>0$, $\forall i\in\mathcal{L}$.

Proof. Since $(L(\mathcal{G})\otimes I_n)(\mathbf{1}_N\otimes c_p) = 0$ ($\mathbf{1}_N$ is an eigenvector of $L(\mathcal{G})$), then $\dot{x}(t) = -(\kappa L(\mathcal{G})+A(t))\otimes I_n(x(t)-\mathbf{1}_N\otimes c_p)$. If we rewrite this in terms of $\delta(t)$, it follows that $\dot{\delta}(t) = -(\kappa L(\mathcal{G})+A(t))\otimes I_n\delta(t)$. Let us denote by t_p^0 and $x(t_p^0)$ ($\delta(t_p^0) = x(t_p^0) - \mathbf{1}_N\otimes c_p$) the starting time and the initial conditions, respectively, for the task ϕ_p. According to (6), $A(s)$ is a piecewise constant function $A(s):\mathbb{R}_{\ge0}\to\mathbb{R}^n_{\ge0}$. Let us denote by $\{t_p^\ell,\ \ell\in\mathbb{N}_{>0}\}$ the sequence of updating times for the set of $\{a_i\}$ while the task p is being executed, i.e.

$$t_p^\ell = \bigcup_{i\in\mathcal{L}}\{t_k^i\} : t_p^0 < t_k^i < t_p^{\text{end}}, \tag{10}$$

where t_p^{end} is the instant of time when the task p is completed and it might occur that $t_k^i = t_{k'}^{i'}$ for some $k,k'\in\mathbb{N}$ and $i,i'\in\mathcal{L}$. However, those coincident instants are denoted by a unique t_p^ℓ and, thus, $t_p^0 < t_p^1 < t_p^2 < \cdots < t_p^{\text{end}}$. Let also denote by A_p^ℓ the value of $A(t)$ in the interval $t_p^\ell \le t < t_p^{\ell+1}$. Then, it holds $\delta(t_p^{\ell+1}) = e^{-((\kappa L(\mathcal{G})+A_p^\ell)\otimes I_n)(t_p^{\ell+1}-t_p^\ell)}\delta(t_p^\ell)$. Let $n_p(t)\in\mathbb{N}$ and denote by $t_p^{n_p(t)}$ the last updating time at time t. We further introduce $M_p^\ell = \kappa L(\mathcal{G}) + A_p^\ell$ and $\tau_p^\ell = t_p^{\ell+1} - t_p^\ell$. Then, using the expression above recursively:

$$\delta(t) = e^{-(M_p^{n_p(t)}\otimes I_n)(t-t_p^{n_p(t)})}\Big(\prod_{\ell=1}^{n_p(t)} e^{-(M_p^{\ell-1}\otimes I_n)\tau_p^{\ell-1}}\Big)\delta(t_p^0). \tag{11}$$

Remark 5. Note that the matrices M_p^ℓ do not commute in general, and therefore the order of terms in (11) is important. Thus, we use the notation of *time-ordered product* (Pazy, 2012): for a family of operators $U_1,\ldots,U_n$, the time-ordered product of these operators is defined by $\prod_{\ell=1}^n U_\ell = U_n\ldots U_1$ and $\prod_{\ell=n}^1 U_\ell = U_1\ldots U_n$.

We next apply the results of Lemma 3 to the set of matrices M_p^ℓ. Note that the eigenvalues of $e^{-M_p^\ell\otimes I_n}$ are all negative (since the eigenvalues of M_p^ℓ are all positive), and this guarantees that $\delta(t)$ in (11) converges to zero ($\delta\to0$ when $t\to\infty$). Moreover, because the set of matrices M_p^ℓ has an orthonormal basis of eigenvectors, it holds that $\|\delta(t)\| = \|e^{-(M_p^{n_p(t)}\otimes I_n)(t-t_p^{n_p(t)})}\prod_{\ell=1}^{n_p(t)} e^{-(M_p^{\ell-1}\otimes I_n)\tau_p^{\ell-1}}\delta(t_p^0)\| \le e^{-\lambda_1(M_p^{n_p(t)})(t-t_p^{n_p(t)})}\prod_{\ell=1}^{n_p(t)} e^{-\lambda_1(M_p^{\ell-1})\tau_p^{\ell-1}}\|\delta(t_p^0)\|$, and then it follows

$$\|\delta(t)\| \le e^{-\underline{\lambda_1}(M_p^{n_p(t)})(t-t_p^{n_p(t)})}\prod_{\ell=1}^{n_p(t)} e^{-\underline{\lambda_1}(M_p^{\ell-1})\tau_p^{\ell-1}}\|\delta(t_p^0)\|, \tag{12}$$

where $\underline{\lambda_1}(M_p^\ell)$ is given by (9). Then, $\delta(t)$ converges asymptotically to 0. Furthermore, the region $\mathcal{B}_p$ is reached in finite time, since there exist a value of $t<\infty$ such that $\|\delta(t)\| < r_p$, for any initial condition $\|\delta(t_p^0)\| < \infty$ □

The previous result proves that each region is reached in finite time. However, δ (i.e., the state of the overall system) cannot be measured since only the leaders are able to compute the distance to the corresponding c_p, i.e., δ_i. Then, the following proposition provides an estimation that each leader can compute based on local measurements that is an upper bound for $\|\delta\|$ at any time t.

Proposition 6. Let us assume that the set of values of $\{a_i\}$ ensures connectivity, according to the Definition 2. Then, the distance to the center of the region $\mathcal{B}_p$ can be estimated by each leader $i\in\mathcal{L}$ at any time t as

$$\hat{d}_i = (N - |\mathcal{N}_i| - 1)R + \sum_{j\in\mathcal{N}_i}\|x_i - x_j\| + \sqrt{N}\|\delta_i\|, \tag{13}$$

and it holds that $\|\delta\| \le \hat{d}_i\ \forall i\in\mathcal{L}$.

Proof. The proof immediately holds taking into account that, for each leader $i \in \mathcal{L}$ it holds $\|\delta\| = \|x - \mathbf{1}_N \otimes c_p\| = \|x - \mathbf{1}_N \otimes x_i + \mathbf{1}_N \otimes x_i - \mathbf{1}_N \otimes c_p\| \leq \|x - \mathbf{1}_N \otimes x_i\| + \|\mathbf{1}_N \otimes (x_i - c_p)\| \leq\leq (N - |\mathcal{N}_i| - 1)\|z\| + \sum_{j\in\mathcal{N}_i} \|x_i - x_j\| + \sqrt{N}\|\delta_i\|$, and then

$$\|\delta\| \leq (N - |\mathcal{N}_i| - 1)R + \sum_{j\in\mathcal{N}_i} \|x_i - x_j\| + \sqrt{N}\|\delta_i\|. \quad (14)$$

The following proposition defines the measured distance by the leader that ensures that all the agents are inside $\mathcal{B}_p$, using the overapproximation of Proposition 6.

Proposition 7. Let us assume that the connectivity radius R satisfies $(N-1)R < r_p$. If the distance of the leader i to the center of the objective region $\mathcal{B}_p$ satisfies

$$\|\delta_i\| \leq \frac{r_p - (N-1)R}{\sqrt{N}} = \bar{r}_p, \quad (15)$$

then all agents of the team are inside $\mathcal{B}_p$.

Proof. The proof is provided in the appendix.

4.3 Connectivity maintenance

The next proposition provides a constraint on the design of the feedback gains a_i to ensure the connectivity of the team. We further make the following assumption.

Assumption 8. The initial conditions satisfy $\|z(t_p^0)\| \leq R$.

Remark 9. Assumption 8 only imposes real constraints over the initial conditions at $t = 0$, since for the rest of the tasks, $\|z(t_p^0)\| \leq R$ can be guaranteed with an adequate choice of the switching rule, as it will be illustrated later.

Proposition 10. Let Assumption 8 holds. Then the graph $\mathcal{G}$ remains connected and no links are lost during the duration of the task ϕ_p if the feedback gains a_i in (5) updated according to (6) satisfy the following constraint:

$$a_i(t_k^i) \leq \frac{R\lambda_2(\mathcal{G})\kappa}{\sqrt{N}\hat{d}_i(t_k^i)}, \quad (16)$$

where $\hat{d}_i$ is defined in (13) and t_k^i denotes the updating instants of a_i.

Proof. Let us assume that no edges are removed or added in the group and that none of the feedback gains a_i have been updated yet in the interval of time considered. Moreover, without loss of generality, let $t_p^0 = 0$. Then, according to (8), the state of the edges z at time t is $z(t) = e^{-\kappa(L_e \otimes I_n)t}z(0) - \int_0^t e^{-\kappa(L_e\otimes I_n)(t-s)}(E^\top A(s) \otimes I_n)(x(s) - \mathbf{1}_N \otimes c_p)ds$. Taking norms, using the result of Lemma 1, and noting that $\|E^\top A(s)\| \leq \sqrt{N}\max_{i\in\mathcal{L}} a_i(s)$, it follows that $\|z(t)\| \leq e^{-\kappa\lambda_2 t}\|z(0)\| + \int_0^t \sqrt{N}\max_{i\in\mathcal{L}} a_i(s)e^{-\kappa\lambda_2(t-s)}\|\delta(s)\|ds$. From Proposition 4, $\|\delta(s)\| \leq \|\delta(0)\|$, and since $a_i(s)$ remains constant in the interval of time considered, then $\|z(t)\| \leq e^{-\kappa\lambda_2 t}\|z(0)\| + \frac{\sqrt{N}\max_{i\in\mathcal{L}} a_i(0)\|\delta(0)\|}{\kappa\lambda_2}(1 - e^{-\kappa\lambda_2 t})$. From Assumption 8, it holds that $\|z(0)\| \leq R$, and the connectivity is maintained if $\|z_j(t)\| \leq R$ for all the edges j in $\mathcal{E}$. Since $\|z_j(t)\| \leq \|z(t)\|$, and applying the results of Proposition 6 it follows that $\|\delta(0)\| \leq \hat{d}_i(0)$, $\forall i \in \mathcal{L}$, the choice of feedback gains $a_i(0) \leq \frac{R\lambda_2(\mathcal{G})\kappa}{\sqrt{N}\hat{d}_i(0)}$, guarantees connectivity maintenance and that no edges are removed in $[0, t)$.

The previous argument can be used for any time t and considering that the last instance of time any feedback gain is updated is t_p^ℓ (see (10)). Similar steps yield $\|z(t)\| \leq e^{-\kappa\lambda_2(t-t_p^\ell)}\|z(t_p^\ell)\| + \frac{\sqrt{N}\max_{i\in\mathcal{L}} a_i(t_p^\ell)\|\delta(t_p^\ell)\|}{\kappa\lambda_2}(1 - e^{-\kappa\lambda_2(t-t_p^\ell)})$, and thus, the choice of (16) guarantees that the connectivity is not broken, which completes the proof. □

Remark 11. In the proof of Proposition 10, it is assumed that no edges are added to the graph $\mathcal{G}$. This can be relaxed without altering the results, since adding new edges in the graph can only increase the value of λ_2 (Zelazo et al., 2007) and, hence, all the upper bounds still hold if the value of λ_2 at the beginning of each interval is considered.

4.4 Time constraint analysis

The planning description given in section 3.1 establishes that the team reaches the objective before some given deadlines T_p. The following proposition provides a value for the maximum distance to the objective that ensures the time constraint is satisfied (and, of course, connectivity maintained) if a_i is held constant in the control law (5). An example will illustrate the conservatism of the results and will motivate the adaptive mechanism for updating a_i and the self-triggered algorithm proposed afterwards.

Proposition 12. Consider the multi-agent system (2) and control law (5) with feedback gains a_i with constant values set as in (16) computed at t_p^0. If for each leader $i \in \mathcal{L}$ the initial estimated distance to the center c_p is such that $\hat{d}_i(t_p^0) \leq d_{max}$, where $d_{max} > 0$ is the solution of

$$\left(\frac{N-1}{N(N-1) + \alpha(d_{max})}\right)^{N-1} \frac{R\lambda_2\kappa|\mathcal{L}|}{\sqrt{N}d_{max}} = \frac{1}{T_p}\log\frac{d_{max}}{r_p}, \quad (17)$$

and $\alpha(d_{max}) = \frac{|\mathcal{L}|R\lambda_2}{\sqrt{N}d_{max}}$, then the team completes the task ϕ_p before the deadline T_p.

Proof. The proof follows from Propositions 4 and 10 (considering that a_i remains constant $\forall t$) and the result of Lemma 3. If a_i remains constant, from (12) it holds that at time $t = t_p^0 + T_p$ $\|\delta(t_p^0 + T_p)\| \leq e^{-\underline{\lambda_1}(M_p)T_p}\|\delta(t_p^0)\|$.. Using the constraint (16) that guarantees the connectivity at time t_p^0 and introducing this into (9), it yields $\underline{\lambda_1}(M_p) = \left(\frac{N-1}{N(N-1)+\alpha(d_{max})}\right)^{N-1}\frac{R\lambda_2\kappa|\mathcal{L}|}{\sqrt{N}d_{max}}$, since $\hat{d}_i(t_p^0) \leq d_{max}$. The fact that the task ϕ_p is completed if $\|\delta(t_p^0 + T_p)\| \leq r_p$, and that $\|\delta(t_p^0)\| \leq \hat{d}_i((t_p^0))$ from (14), completes the proof. □

Remark 13. The existence of a positive d_{max} is illustrated graphically in Fig. 1. Let us denote the left and right hand sides of (17) as f_1 and f_2, respectively. Note that f_1 behaves approximately like the reciprocal function for d_{max}, when $d_{max} > 0$ it takes positive values, and it goes to $+\infty$ when $d_{max} \to 0$ and to 0 when $d_{max} \to +\infty$. f_2 is also a positive function for $d_{max} > r_p$ that goes to 0 when $d_{max} \to r_p$ and to $+\infty$ when $d_{max} \to +\infty$. Then, both functions intersect for some value $r_p < d_{max} < +\infty$.

Example 14. Consider a four agents system evolving in $\mathbb{R}^2$. The graph defining the neighbor's relations is a path graph with a single leader, v_1. The rest of the parameters here are $r_p = 0.2$ m, $T_p = 20$ s, $R = 0.05$ s, $\kappa = 100$. This yields a value of $d_{max} = 0.58$ m as the solution of (17) and a feedback gain $a_1 = 2.54$. Note that d_{max} is an upper bound for $\|\delta\|$, i.e., the norm of the state of the whole team of agents. We can, however, estimate a value for the admissible distance to the objective measured by

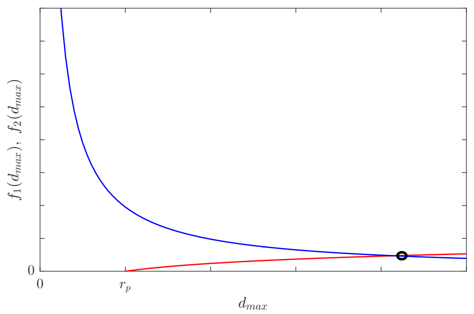

Fig. 1. Graphical representation of the solution d_{max} of (17). f_1 represents the left hand side (blue line), f_2 represents the right hand side (red line) of (17). The solution is represented by a black circle.

the leader $\|\delta_i(t_p^0)\|$ as $\|\delta_i(t_p^0)\| \leq \frac{d_{max}-(N-1)R}{\sqrt{N}} = (0.58 - 3 \cdot 0.05)/\sqrt{4} = 0.22$ m, i.e., the leader being at 2 cm to the border of $\mathcal{B}_p$, which is a really conservative result and motivates the next subsection.

4.5 Self-triggered algorithm

In order to avoid the conservatism of the previous result, an adaptive mechanism for the update of the feedback gains a_i is defined in Algorithm 1. The instances of time for the update follow a self-triggered policy. We first present the algorithm executed by the leaders, and then we provide the sufficient conditions for the measured distance at the beginning of the task that guarantees the task is completed. However, it might occur that the Algorithm 1 ends successfully, that is, there is no deadline violation, even if the constraint over $\|\delta_i(t_p^0)\|$ is not satisfied.

Algorithm 1 can be summarized as follows. Each leader measures the distance to the center of the objective region c_p and computes the feedback gain according to (16) (lines 1-2). Then, it checks if the measured distance guarantees that the task will be completed before T_p, according to the results of Proposition 12 (line 3). In the affirmative case, the computation ends. If the constraint cannot be satisfied, each leader makes an optimistic estimation of the traveling time to reach d_{max} (the solution of (17)) with the initial computation of a_i, and the team keeps moving without any further calculations (lines 6-7). Once this time has elapsed, the leader takes new measurements and executes the procedure again with the updated parameters (lines 8-12). Note that the parameter τ_i represents the remaining time to complete the task before T_p. The computation ends in three situations: 1) when the time constraint can be satisfied with the current values of a_i ($d_{max} \geq \hat{d}_i$); 2) the team is inside the objective region before the deadline T_p ($\hat{d}_i \leq r_p$); 3) the deadline T_p has elapsed without completing the task ($\tau_i \leq 0$). The first two represent a success, while the third case is a fail.

Remark 15. The updating times defined in Algorithm 1 (line 6) are lower bounds of the traveling time required by the leader to reach the distance given as a solution of (17) if the cooperative term is left out, that is, if the dynamics of the leader agents are given by $\dot{x}_i(t) = -a_i(t_k^i)(x_i(t) - c_p)$ for $t \in [t_k^i, t_{k+1}^i)$. Then, it holds that $\delta_i(t_{k+1}^i) = e^{-a_i(t_{k+1}^i - t_k^i)}\delta_i(t_k^i)$. Since $\|\delta_i(t_{k+1}^i)\| \leq \|\delta(t_{k+1}^i)\|$, then

$$t_{k+1}^i = t_k^i + \frac{1}{a_i}\log\frac{\|\delta_i(t_k^i)\|}{d_{max}(t_k^i)} \tag{18}$$

is an optimistic estimation of the time required by the leader to command the team so that the distance d_{max} is reached by the group.

Algorithm 1 Computation of a_i

Input: κ, R, $\lambda_2(\mathcal{G})$, N, T_p.
Output: a_i.
1: Compute $\hat{d}_i$ according to (13) at time $t = t_p^0$
2: Compute a_i according to (16) as $a_i(t_p^0) = \frac{R\lambda_2(\mathcal{G})\kappa}{\sqrt{N}\hat{d}_i}$
3: Compute d_{max} according to (17)
4: Set $\tau_i = T_p$ and $t_k^i = t_p^0$
5: **while** $d_{max} < \hat{d}_i$ & $\hat{d}_i > r_p$ & $\tau_i > 0$ **do**
6: Define next update time $t_{k+1}^i = t_k^i + \frac{1}{a_i}\log\frac{\|\delta_i(t_k^i)\|}{d_{max}}$
7: Wait until $t = t_{k+1}^i$
8: Update $\tau_i = T_p - (t_{k+1}^i - t_k^i)$
9: Measure distance to the objective $\|\delta_i\|$
10: Update $\hat{d}_i$ according to (13) at t_{k+1}^i
11: Update a_i according to (16) as $a_i(t_{k+1}^i) = \frac{R\lambda_2(\mathcal{G})\kappa}{\sqrt{N}\hat{d}_i}$
12: Compute d_{max} according to (17) but replacing T_p by τ_i
13: $k = k + 1$
14: **end while**
15: **return** a_i

Proposition 16. Consider the multi-agent system (2) with control law (5) and adaptive feedback gains a_i (6) updated according to (16) at instances of time (18). If for each leader $i \in \mathcal{L}$ the initial measured distance to the center c_p is such that $\|\delta_i(t_p^0)\| \leq d_{max}$, where $d_{max} > 0$ is

$$d_{max} = -\alpha_d + \sqrt{\alpha_d^2 + 2\beta_d}, \tag{19}$$

$\alpha_d = (N-1)R$, $\beta_d = \frac{R\lambda_2\kappa\bar{r}_pT_p}{\sqrt{N}} + \frac{1}{2}\bar{r}_p^2 + (N-1)R\bar{r}_p$, and $\bar{r}_p = \frac{r_p-(N-1)R}{\sqrt{N}}$, then the team completes the task ϕ_p before the deadline T_p.

Proof. According to (18), the sum of the inter-updating times has to satisfy $\sum_k(t_{k+1}^i - t_k^i) \leq T_p$, taking only the subset of updating times t_k^i such that $t_p^0 \leq \{t_k^i\} \leq t_p^0 + T_p$, with the constraint that $\|\delta(t_p^0 + T_p)\| \leq r_p$, in order to ensure that the task ϕ_p is completed. Then

$$\sum_{\substack{k\in\mathbb{N} \\ t_p^0 \leq t_k^i \leq t_p^0 + T_p}} \frac{1}{a_i(t_k^i)}\log\frac{\|\delta_i(t_k^i)\|}{d_{max}(t_k^i)} \leq T_p. \tag{20}$$

Note that $d_{max}(t_k^i) \geq \|\delta_i(t_{k+1}^i)\|$.Then, and according to (16), the left hand side of (20) can be upper bounded by $\sum_k \frac{\hat{d}_i(t_k^i)}{R\lambda_2\kappa}\log\frac{\|\delta_i(t_k^i)\|}{\|\delta_i(t_{k+1}^i)\|}$. Using $\log x \leq (x-1)$ if $x \geq 1$, replacing $\hat{d}_i(t_k^i)$ by its upper bound $\|\delta_i(t_k^i)\| + (N-1)R$ (see Proposition 7), and noting that $\|\delta_i(t_{k+1})\| \geq \bar{r}_p$ (otherwise the algorithm stops), it follows

$$\sum_k(t_{k+1}^i - t_k^i) \leq \sum_k \frac{\|\delta_i(t_k^i)\| + (N-1)R}{R\lambda_2\kappa\bar{r}_p}\left(\|\delta_i(t_k^i)\| - \|\delta_i(t_{k+1}^i)\|\right).$$

The above sum can be computed as $\Delta(t_0^p + T_p) - \Delta(t_0^p)$, where the function $\Delta(t)$ is $\Delta(t) = \frac{-\sqrt{N}}{R\lambda_2\kappa\bar{r}_p}\left(\frac{1}{2}\|\delta_i(t)\|^2 + (N-\right.$

$1)R\|\delta_i(t)\|\big)$. Imposing the constraint for the satisfaction of the task $\|\delta_i(t_0^p + T_p)\| \leq \bar{r}_p$, yields

$$\tfrac{1}{2}\|\delta_i(t_0^p)\|^2 + (N-1)R\|\delta_i(t_0^p)\| \leq \tfrac{R\lambda_2\kappa\bar{r}_pT_p}{\sqrt{N}} + \tfrac{\bar{r}_p^2}{2} + (N-1)R\bar{r}_p,$$

whose feasible solution yields the bound d_{max} in (19). $\square$

Example 17. Let us take the same setting as in the Example 14. The solution of d_{max} given by (19) is $d_{max} = 1.26$ m. Note that is a bound for the initial conditions of $\|\delta_i\|$ not for $\|\delta\|$. Hence, $\|\delta_i(t_0^p)\|$ can be increased more than 5 times with the self-triggered adaptive scheme.

4.6 Accomplishment of the high level plan

The following theorem, based on the previous results, summarizes the main results of this work.

Theorem 18. Consider the multi-agent system (2) with control law (5) and adaptive feedback gains a_i (6) updated according to the Algorithm 1. If for each leader $i \in \mathcal{L}$ the initial conditions satisfy $\|\delta_i(0)\| \leq d_{max}$ and the distance between the centers of the two regions for consecutive tasks p and $p+1$ satisfy $\|c_{p+1} - c_p\| \leq d_{max} - \bar{r}_p$, for all $p = 1, \dots, N_{task} - 1$, where $d_{max} > 0$ is given by (19), then the high level plan ϕ is successfully completed.

Proof. The proof immediately follows applying Proposition 16. If $\|\delta_i(0)\| \leq d_{max}$ $\forall i \in \mathcal{L}$, the task $p = 1$ is completed on time since it is guaranteed that the measured distance satisfies $\|\delta_i(t)\| \leq \bar{r}_1$ for some time t such that $t \leq T_1$. For the rest of the tasks $p = 2, \dots, N-1$, $\|c_{p+1} - c_p\| \leq d_{max} - \bar{r}_p$ ensures that $\|\delta_i(t_p^0)\| \leq d_{max}$, which according to Proposition 16 guarantees that the team reaches each region $\mathcal{B}_p$ before the deadline T_p. Hence, the high level plan is completed. $\square$

5. SIMULATION EXAMPLE

Let us consider a team of $N = 5$ agents evolving in $\mathbb{R}^2$ whose topology is defined by

$$L(\mathcal{G}) = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}, \quad \mathcal{L} = \{v_2, v_4\}.$$

The planning consists of $N_{task} = 4$ tasks, and the objective regions are defined as follows: $\mathcal{B}_1 = (c_1 = (1,2)^\top, r_1 = 0.25)$, $\mathcal{B}_2 = (c_2 = (3,0)^\top, r_2 = 0.25)$, $\mathcal{B}_3 = (c_3 = (0,0)^\top, r_3 = 0.25)$, and $\mathcal{B}_4 = (c_4 = (-1,2)^\top, r_4 = 0.25)$. The respective deadlines are $T_1 = 15$ s, $T_2 = 13$ s, $T_3 = 12$ s, and $T_4 = 15$ s. Furthermore, the connectivity radius is $R = 0.05$ m and the feedback gain $\kappa = 100$. The leaders of the team update the feedback gains a_i according to Algorithm 1. The state evolution is depicted in Fig. 2. The time deadlines T_p are depicted with vertical lines in red. However, the team reaches the objective before T_p, for all p, and switches to the next task. These instances of time are depicted in green vertical lines. On the top right corner of the figure, a zoom for the interval of time [0,0.1] s is shown to illustrate how the agents reach consensus.

The agents' trajectories in $\mathbb{R}^2$ are depicted in Fig. 3. The objective regions, whose radius is $r_p = 0.25$ m are depicted in dashed line. However, to ensure that all agents are inside $\mathcal{B}_p$, leader agents do not switch of task until the the distance measured to c_p reaches the value of $\bar{r}_p$ (0.0224 m in this example for all tasks), according to Proposition 7. This define smaller regions, which are depicted in black solid lines. Finally, Fig. 4 illustrates the evolution of a_i for the leaders (agent 2 and agent 4) when they execute Algorithm 1. Note how the feedback gains take conservative values at the beginning of each task, and when they are closer to the objective regions, they increase the value of a_i.

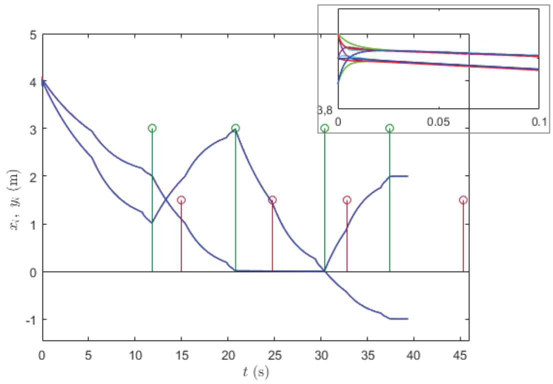

Fig. 2. State evolution of the system.

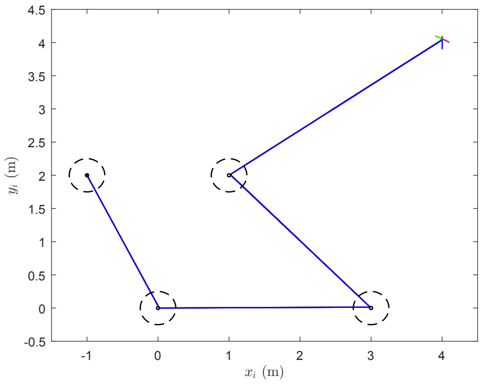

Fig. 3. Agents' trajectories in $\mathbb{R}^2$.

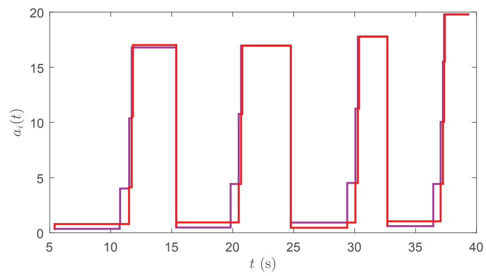

Fig. 4. Evolution of $a_i(t)$ for agent 2 (violet line) and agent 4 (red line).

6. CONCLUSIONS

We have presented distributed control laws for the coordination of a multi-agent system that is requested to visit sequentially a finite number of regions of the state-space with some timed constraints while maintaining connectivity. The existing coupling between satisfying the given deadlines and keeping the agents close to each other is solved by the distributed implementation of a self-triggered algorithm to update the controller gains. Simulation results have illustrated the validity of the proposed approach. Future work will address time-varying topologies and the effect of disturbances.

APPENDIX

Proof of Lemma 3: The Laplacian matrix L has eigenvalues $0 = \lambda_1(L) < \lambda_2(L) \leq ... \leq \lambda_N(L)$. Furthermore, A has eigenvalues equal to a_i, $i = 1, \dots, N$, where $a_i > 0$ if $i \in \mathcal{L}$ (i.e., some of the eigenvalues can be 0). Note that $M = \kappa L + A$ is positive definite by construction ($\kappa, a_i > 0$), and thus all its eigenvalues are real and positive. Then, the results of Lemma 1 in Merikoski and Virtanen (1997) apply, i.e., for any matrix $C \in \mathbb{C}^{n\times n}$ with real and positive eigenvalues it holds that

$$\lambda_1(C) \geq \left(\tfrac{n-1}{\mathrm{tr}(C)}\right)^{n-1} \det(C), \tag{21}$$

where $\lambda_1(C)$ is the smallest eigenvalue of C.

The trace of the matrix M is upper bounded by

$$\mathrm{tr}(M) \leq \kappa N(N-1) + \sum_{i=1}^{N} a_i > 0, \tag{22}$$

since the diagonal elements of L are the degree of each vertex which is upper bounded by $N-1$.

We next find a lower bound for the determinant of M. The determinant of M is the product of its eigenvalues and, thus, is positive. From Weyl's theorem Horn and Johnson (2012), it holds that $0 < \lambda_1(M) \leq \kappa\lambda_2(L)$, $\kappa\lambda_N(L) \leq \lambda_N(M)$, and $\kappa\lambda_i(L) \leq \lambda_i(M) \leq \kappa\lambda_{i+1}(L)$ $\forall i = 2, \dots, N-1$ (the eigenvalues interlace). Connected graphs contain a spanning tree, and the edges that are not in the given spanning tree must complete the cycles of the graph. Furthermore, according to Theorem 4.1 in Zelazo et al. (2007), the eigenvalues of a connected graph Laplacian L are lower bounded by those of the spanning tree. Hence, we next analyze the case of M when the graph is a tree and, for simplicity, a path graph, and we show that adding an edge in the graph can only increase the determinant. The extension to a general tree is then straightforward. We denote by P_N the path graph with N vertices.

By mathematical induction in the total number of agents N and in the number of leaders $|\mathcal{L}|$, it can be proven (the steps are omitted due to space constraints) that the $\det(M)$ for the path graph P_N is lower bounded by

$$\det(M) > \kappa^{N-1} \sum_{i=1}^{N} a_i. \tag{23}$$

Next lets prove that adding new edges in the graph can only increase the determinant so that the results hold for a general connected graph. Assume that an edge is added to P_N and denote this graph by P_N^{+e}. Then the incidence matrix of P_N^{+e} is $E(P_N^{+e}) = [E(P_N)\ e]$ and the Laplacian matrix $L(P_N^{+e}) = L(P_N) + ee^\top$, where $e^\top = (e_1 \dots e_N)$ denotes a column corresponding to the added edge and $e_i = 1$ if i is the initial node of the edge, $e_i = -1$ if i is the terminal node of the edge, otherwise $e_i = 0$. Thus, in this case, the matrix M can be written as $M = \kappa(L(P_N) + ee^\top) + A$. Using the fact that $\det(A + B) \geq \det(A) + \det(B)$, it follows that adding new edges in the graph can only increase the determinant and (23) is a lower bound: $(\det(\kappa(L(P_N) + ee^\top) + A) \geq \det(\kappa(L(P_N) + A)) + \det(ee^\top) = \kappa^{N-1}\sum_{i=1}^{N} a_i)$. Combining this with (22) and (21), it yields $\left(\frac{N-1}{N(N-1)\kappa + \sum_{i=1}^{N} a_i}\right)^{N-1} \kappa^{N-1}\sum_{i=1}^{N} a_i \leq \lambda_1(M) \leq \dots \leq \lambda_N(M)$, which completes the proof. □

Proof of Proposition 7: The proof intermediately follows from Proposition 6. If $\|\delta_i\| \leq \bar{r}_p$, then the distance to the objective region of the team $\|\delta\|$ satisfies $\|\delta\| \leq (N-1)R + \sqrt{N}\bar{r}_p = r_p$. Then, all agents lie inside $\mathcal{B}_p$. □

REFERENCES

Godsil, C. and Royle, G. (2001). *Algebraic Graph Theory*. Springer.

Gombolay, M.C., Wilcox, R.J., and Shah, J.A. (2018). Fast scheduling of robot teams performing tasks with temporospatial constraints. *IEEE Trans. on Robot.*, 34(1), 220–239.

Guinaldo, M. and Dimarogonas, D. (2017). A hybrid systems framework for multi agent task planning and control. In *American Control Conference*, 1181–1186.

Guo, M. and Zavlanos, M.M. (2017). Distributed data gathering with buffer constraints and intermittent communication. In *IEEE International Conference on Robotics and Automation*, 279–284.

Heemels, W., Johansson, K.H., and Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *51st IEEE Conference on Decision and Control*, 3270–3285.

Hong, Y., Hu, J., and Gao, L. (2006). Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7), 1177–1182.

Horn, R. and Johnson, C. (2012). *Matrix analysis*. Cambridge university press.

Ji, M. and Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Trans. on Robot.*, 23(4), 693–703.

Lafferriere, G., Williams, A., Caughman, J., and Veerman, J. (2005). Decentralized control of vehicle formations. *Syst. Control Lett.*, 54(9), 899 – 910.

Merikoski, J. and Virtanen, A. (1997). Bounds for eigenvalues using the trace and determinant. *Linear Algebra and its Applications*, 264(Supplement C), 101 – 108.

Nikou, A., Tumova, J., and Dimarogonas, D.V. (2016). Cooperative task planning of multi-agent systems under timed temporal specifications. In *2016 American Control Conference*, 7104–7109.

Pazy, A. (2012). *Semigroups of linear operators and applications to partial differential equations*, volume 44. Springer Science & Business Media.

Ren, W. (2007). Multi-vehicle consensus with a time-varying reference state. *Syst. Control Lett.*, 56(7), 474–483.

Zelazo, D., Rahmani, A., and Mesbahi, M. (2007). Agreement via the edge laplacian. In *46th IEEE Conference on Decision and Control*, 2309–2314.