Distributed Event-Triggered Control for Multi-Agent Systems

Dimos V. Dimarogonas, Emilio Frazzoli and Karl H. Johansson

Abstract—Event-driven strategies for multi-agent systems are motivated by the future use of embedded microprocessors with limited resources that will gather information and actuate the individual agent controller updates. The controller updates considered here are event-driven, depending on the ratio of a certain measurement error with respect to the norm of a function of the state, and are applied to a first order agreement problem. A centralized formulation is considered first and then its distributed counterpart, in which agents require knowledge only of their neighbors' states for the controller implementation. The results are then extended to a self-triggered set-up, where each agent computes its next update time at the previous one, without having to keep track of the state error that triggers the actuation between two consecutive update instants. The results are illustrated through simulation examples.

I. INTRODUCTION

Distributed control of large scale multi-agent systems is currently facilitated by recent technological advances on computing and communication resources. Several results concerning multi-agent cooperative control have appeared in the recent literature involving agreement or consensus algorithms [20], [12],[23], formation control [5], [4], [8], [2] and distributed estimation [21],[25].

An important aspect in the implementation of distributed algorithms is the communication and controller actuation schemes. A future design may equip each agent with an embedded micro-processor, which will be responsible for collecting information from neighboring nodes and actuating the controller updates according to some rules. The goal of this paper is to provide rules in order to reduce the number of the actuator updates when this is preferable for the application in hand. This might be suitable, e.g., in the case of micro-processors with attached sensors. The scheduling of the actuator updates can be done in a time- or an event-driven fashion. The first case involves sampling at pre-specified time instances, usually separated by a fixed period. When the limited resources of embedded processors are considered, an event-triggered approach seems more favorable. In addition, a proper design should also preserve desired properties of the nominal system, such as stability and convergence. A comparison of time- and event-driven control for stochastic systems favoring the latter is found in [3]. Stochastic event-driven strategies have appeared in [22],[13]. In this paper, we use the deterministic event-triggered strategy introduced in [26]. Related results on deterministic event-triggered feedback control have appeared in [29],[28],[11],[18],[1],[15],[10].

In [26], the control actuation is triggered whenever a certain error becomes large enough with respect to the norm of the state. It is assumed that the nominal system is Input-to-State stable [24] with respect to measurement errors. We first show that this framework is suitable for a class of cooperative control algorithms, namely those that can be reduced to a first order agreement problem [20], which

Dimos V. Dimarogonas and Karl H. Johansson are with the KTH ACCESS Linnaeus Center, School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden {dimos,kallej@ee.kth.se}. The first author is also affiliated with the KTH Center of Autonomous Systems (CAS) and is supported by the Swedish Research Council (VR) through contract 2009-3948. Emilio Frazzoli is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, U.S.A. {frazzoli@mit.edu}. The work of the first and third authors was also supported by the Swedish Research Council, the Swedish Foundation for Strategic Research, and the EU FeedNetBack STREP FP7 project. The results of this paper were partially presented in [7],[6].

was shown to be ISS [14]. Both the centralized and distributed cases are considered. We then consider a self-triggered solution to the multi-agent agreement problem. In particular, each agent now computes its next update time at the previous one, without having to keep track of the error measurement that triggers the actuation between two consecutive updates. The approach is first presented in a centralized fashion, and then in a distributed one. Self-triggered control is a natural extension of the event-triggered approach and has been considered in [1],[29],[18]. In addition, the self-triggered analysis provides further results regarding the calculation of the inter-execution times in the distributed event-triggered case.

The resulting model of the system can be transformed to a time-delay system with varying delays which are different between the agent and its neighbors. This is in contrast to the first order agreement time-delayed models with constant delays [19],[16] and the first order agreement models with varying delays that do not consider self delay in the agents' state or consider equal delays between each agent and its neighbors [17]. Note that in the absence of self-delays, convergence is guaranteed even for the case of heterogeneous delays and asynchronous updates [27]. However, self delays are present in our model. In essence the delayed model resulting from the decentralized event triggered setup is more general than the first order agreement models with delays found in literature. On an equally important sidenote, it should be emphasized that a design that provides piecewise constant control laws with an event-triggered mechanism that is the decision maker of when the control law is updated, seems more applicable to networked multi-agent systems than an approach that assumes delayed information with continuously varying control laws and that provides no decision mechanism on when the sampling should take place.

The remainder is organized as follows: Section II presents some background and discusses the system model treated in the paper. The centralized event-triggered control design is discussed in Section III while Section IV presents the distributed counterpart. The self-triggered formulation of the frameworks of Sections III and IV is presented in Section V. Some examples are given in Section VI while Section VII includes a summary of the results of this paper and indicates further research directions.

II. BACKGROUND AND SYSTEM MODEL

A. Algebraic Graph Theory

For an undirected graph G with N vertices, the *adjacency matrix* $A = A(G) = (a_{ij})$ is the $N \times N$ matrix given by $a_{ij} = 1$, if $(i,j) \in E$, where E is the set of edges, and $a_{ij} = 0$, otherwise. If there is an edge $(i,j) \in E$, then i, j are called *adjacent*. A *path* of length r from a vertex i to a vertex j is a sequence of $r+1$ distinct vertices starting with i and ending with j such that consecutive vertices are adjacent. For $i = j$, this path is called a *cycle*. If there is a path between any two vertices of the graph G, then G is called *connected*. A connected graph is called a *tree* if it contains no cycles. The *degree* d_i of vertex i is defined as the number of its neighboring vertices, i.e. $d_i = \text{card}\{j : (i,j) \in E\}$. Let Δ be the $n \times n$ diagonal matrix of d_i's. Then Δ is called the *degree matrix* of G. The (combinatorial) *Laplacian* of G is the symmetric positive semidefinite matrix $L = \Delta - A$. For a connected graph, the Laplacian has a single zero eigenvalue and the corresponding eigenvector is the vector of ones, **1**. We denote by $0 = \lambda_1(G) \leq \lambda_2(G) \leq \ldots \leq \lambda_N(G)$ the eigenvalues of L. If G is connected, then $\lambda_2(G) > 0$.

B. System Model

The system considered consists of N agents, with $x_i \in \mathbb{R}$ denoting the state of agent i. Note that the results of the paper are extendable

to arbitrary dimensions. We assume that the agents' dynamics obey a single integrator model:

$$\dot{x}_i = u_i, i \in \mathcal{N} = \{1, \ldots, N\} \quad (1)$$

where u_i denotes the control input for each agent.

Each agent is assigned a subset $N_i \subset \{1, \ldots, N\}$ of the other agents, called agent i's *communication set*, that includes the agents with which it can communicate. The undirected *communication graph* $G = \{V, E\}$ of the multi-agent team consists of a set of vertices $V = \{1, \ldots, N\}$ indexed by the team members, and a set of edges, $E = \{(i, j) \in V \times V | i \in N_j\}$ containing pairs of vertices that correspond to communicating agents.

The agreement control laws in [9], [20] were given by

$$u_i = -\sum_{j \in N_i} (x_i - x_j) \quad (2)$$

and the closed-loop equations of the nominal system were $\dot{x}_i = -\sum_{j \in N_i} (x_i - x_j)$, $i \in \{1, \ldots, N\}$, so that $\dot{x} = -Lx$, where $x = [x_1, \ldots, x_N]^T$ is the stack vector of agents' states and L is the Laplacian of the communication graph. We also denote by $u = [u_1, \ldots, u_N]^T$ the stack vector of control inputs. For a connected graph, all agents' states converge to a common point, called the "agreement point", which coincides with the average $\frac{1}{N} \sum_i x_i(0)$ of the initial states.

Note that the model (1),(2) has been shown to capture the behavior of other multi-agent control problems as well. For example, it was shown in [8] that a class of formation control problems can be reduced to a first order agreement one with an appropriate transformation.

The above control formulation is redefined here to integrate event-triggered strategies. Considering the system (1), both centralized and distributed event-triggered cooperative control are treated. The control formulation for each case is described in the following sections.

III. CENTRALIZED APPROACH

We first consider the centralized event-triggered control scheme in the following paragraphs. The distributed case is treated in the next section.

For each agent i, and $t \geq 0$, introduce a time-varying error $e_i(t)$. Denote the vector $e(t) = [e_1(t), \ldots, e_N(t)]^T$. The sequence of event-triggered executions is denoted by: $t_0, t_1, \ldots$. To the sequence of events $t_0, t_1, \ldots$ corresponds a sequence of control updates $u(t_0), u(t_1), \ldots$. Between control updates the value of the input u is held constant in a zero-order hold fashion, and is equal to the last control update, i.e.,:

$$u(t) = u(t_i), \forall t \in [t_i, t_{i+1}) \quad (3)$$

and thus the control law is *piecewise constant* between the event times $t_0, t_1, \ldots$.

Following the above notation, the state measurement error is defined by

$$e(t) = x(t_i) - x(t), \quad i = 0, 1, \ldots \quad (4)$$

for $t \in [t_i, t_{i+1})$. The event-triggered design involves the choice of appropriate t_i. The proposed control law in the centralized case has the form (3) and is defined as the event-triggered analog of the ideal control law (2):

$$u(t) = -Lx(t_i), t \in [t_i, t_{i+1}) \quad (5)$$

The closed loop system is then given by $\dot{x}(t) = -Lx(t_i) = -L(x(t) + e(t))$. Denote by $\bar{x}(t) = \frac{1}{N} \sum_i x_i(t)$ the average of the agents' states. Given that the graph is undirected,

we have $\dot{\bar{x}} = \frac{1}{N} \sum_i \dot{x}_i = -\frac{1}{N} \sum_i \sum_{j \in N_i} (x_i(t) - x_j(t)) - \frac{1}{N} \sum_i \sum_{j \in N_i} (e_i(t) - e_j(t)) = 0$ so that $\bar{x}(t) = \bar{x}(0) = \frac{1}{N} \sum_i x_i(0) \equiv \bar{x}$, i.e., the initial average remains constant. A candidate ISS Lyapunov function for the closed-loop system is: $V = \frac{1}{2} x^T L x$. We have $\dot{V} = x^T L \dot{x} = -x^T LL(x + e) = -\|Lx\|^2 - x^T LLe$, so that $\dot{V} \leq -\|Lx\|^2 + \|Lx\| \|L\| \|e\|$. Enforcing e to satisfy

$$\|e\| \leq \sigma \frac{\|Lx\|}{\|L\|} \quad (6)$$

with $\sigma \in (0, 1)$, we get $\dot{V} \leq (\sigma - 1) \|Lx\|^2$ which is negative for $\sigma < 1$ and $\|Lx\| \neq 0$.

Thus, the events are triggered when:

$$\|e\| = \sigma \frac{\|Lx\|}{\|L\|} \quad (7)$$

The event times are thus defined by $\|e(t_i)\| = \sigma \frac{\|Lx(t_i)\|}{\|L\|} = 0$, for $i = 0, 1, \ldots$. At each t_i, the control law is updated according to (5): $u(t_i) = -Lx(t_i)$, and remains constant, i.e., $u(t) = -Lx(t_i)$ for all $t \in [t_i, t_{i+1})$. Once the control task is executed the error is reset to zero, since at that point we have $e(t_i) = x(t_i) - x(t_i) = 0$ for the specific event time so that (6) is enforced.

The following result regarding the convergence of the closed-loop system is now evident:

Theorem 1: Consider system $\dot{x} = u$ with the control law (5),(7) and assume that the communication graph G is connected. Suppose that $0 < \sigma < 1$. Then all agents are asymptotically stabilized to their initial average, i.e., $\lim_{t \to \infty} x_i(t) = \bar{x} = \frac{1}{N} \sum_i x_i(0)$ for all $i \in \mathcal{N}$.

Proof: Similarly to [26], since $\dot{V} \leq (\sigma - 1) \|Lx\|^2$, we have that $\lim_{t \to \infty} Lx(t) = 0$. Since G is connected, the latter corresponds to the fact that all elements of x are equal at steady state, i.e., $\lim_{t \to \infty} x_i(t) = x^*$. Since the initial average remains constant we have $x^* = \bar{x} = \frac{1}{N} \sum_i x_i(0)$ at steady state. $\diamond$

Under the proposed control policy, the inter-event times are lower bounded away from zero. This is proven in the following theorem:

Theorem 2: Consider system $\dot{x} = u$ with the control law (5),(7) and assume that the communication graph G is connected. Suppose that $0 < \sigma < 1$. Then for any initial condition in $\mathbb{R}^N$ the inter-event times $\{t_{i+1} - t_i\}$ implicitly defined by the rule (7) are lower bounded by a strictly positive time τ which is given by $\tau = \frac{\sigma}{\|L\|(1 + \sigma)}$.

Proof: Similarly to [26], we can show that the time derivative of $\frac{\|e\|}{\|Lx\|}$ satisfies $\frac{d}{dt} \frac{\|e\|}{\|Lx\|} \leq \left(1 + \frac{\|L\| \|e\|}{\|Lx\|}\right)^2$. Denoting $y = \frac{\|e\|}{\|Lx\|}$, we have $\dot{y} \leq (1 + \|L\| y)^2$, so that y satisfies the bound $y(t) \leq \phi(t, \phi_0)$ where $\phi(t, \phi_0)$ is the solution of $\dot{\phi} = (1 + \|L\| \phi)^2$, $\phi(0, \phi_0) = \phi_0$. Hence the inter-event times are bounded from below by the time τ that satisfies $\phi(\tau, 0) = \frac{\sigma}{\|L\|}$. The solution of the above differential equation is $\phi(\tau, 0) = \frac{\tau}{1 - \tau \|L\|}$, so that $\tau = \frac{\sigma}{\|L\|(1 + \sigma)}$, and the proof is complete. $\diamond$

IV. DISTRIBUTED APPROACH

In the centralized case, all agents have to be aware of the global measurement error e in order to enforce the condition (6). In this section, we consider the distributed counterpart. In particular, each agent now updates its own control input at event times it decides based on information from its neighboring agents. The event times for each agent $i \in \mathcal{N}$ are denoted by $t_0^i, t_1^i, \ldots$. The measurement error for agent i is defined as

$$e_i(t) = x_i(t_k^i) - x_i(t), t \in [t_k^i, t_{k+1}^i) \quad (8)$$

The distributed control strategy for agent i is now given by:

$$u_i(t) = -\sum_{j \in N_i} \left(x_i(t_k^i) - x_j(t_{k'(t)}^j) \right) \qquad (9)$$

where $k'(t) \triangleq \arg \min_{l \in \mathbb{N}: t \geq t_l^j} \{t - t_l^j\}$. Thus for each $t \in [t_k^i, t_{k+1}^i)$, $t_{k'(t)}^j$ is the last event time of agent j. Hence, each agent takes into account the last update value of each of its neighbors in its control law. The control law for i is updated both at its own event times $t_0^i, t_1^i, \ldots$, as well as at the event times of its neighbors $t_0^j, t_1^j, \ldots, j \in N_i$.

Note that this definition of k' implies $x_j(t_{k'(t)}^j) = x_j(t) + e_j(t)$. We thus have $\dot{x}_i(t) = -\sum_{j \in N_i} \left(x_i(t_k^i) - x_j(t_{k'(t)}^j) \right) = -\sum_{j \in N_i} (x_i(t) - x_j(t)) - \sum_{j \in N_i} (e_i(t) - e_j(t))$, so that $\dot{x}(t) = -L(x(t) + e(t))$ in stack vector form. Hence in this case we also have $\dot{\bar{x}} = 0$ for the agents' initial average.

Denote now $Lx \triangleq z = [z_1, \ldots, z_N]^T$. Note that each element of Lx contains exactly the relative state information that is available to each agent from its neighbors, that is

$$z_i(t) = \sum_{j \in N_i} (x_i(t) - x_j(t)), i = 1, \ldots, N$$

Consider again $V = \frac{1}{2} x^T L x$. Then

$$\dot{V} = x^T L \dot{x} = -x^T L (Lx + Le) = -z^T z - z^T L e$$

From the definition of the Laplacian matrix we get $\dot{V} = -\sum_i z_i^2 - \sum_i \sum_{j \in N_i} z_i (e_i - e_j) = -\sum_i z_i^2 - \sum_i |N_i| z_i e_i + \sum_i \sum_{j \in N_i} z_i e_j$. Using now the inequality $|xy| \leq \frac{a}{2} x^2 + \frac{1}{2a} y^2$, for $a > 0$, we can bound $\dot{V}$ as

$$\dot{V} \leq -\sum_i z_i^2 + \sum_i a|N_i|z_i^2 + \sum_i \frac{1}{2a}|N_i|e_i^2 + \sum_i \sum_{j \in N_i} \frac{1}{2a} e_j^2$$

where $a > 0$.

Since the graph is symmetric, by interchanging the indices of the last term we get $\sum_i \sum_{j \in N_i} \frac{1}{2a} e_j^2 = \sum_i \sum_{j \in N_i} \frac{1}{2a} e_i^2 = \sum_i \frac{1}{2a}|N_i|e_i^2$ so that $\dot{V} \leq -\sum_i (1 - a|N_i|)z_i^2 + \sum_i \frac{1}{a}|N_i|e_i^2$. Assume that a satisfies $0 < a < \frac{1}{|N_i|}$ for all $i \in \mathcal{N}$. Then, enforcing the condition

$$e_i^2 \leq \frac{\sigma_i a (1 - a|N_i|)}{|N_i|} z_i^2 \qquad (10)$$

for all $i \in \mathcal{N}$, we get

$$\dot{V} \leq \sum_i (\sigma_i - 1)(1 - a|N_i|)z_i^2$$

which is negative definite for $0 < \sigma_i < 1$.

Thus for each i, an event is triggered when

$$e_i^2 = \frac{\sigma_i a (1 - a|N_i|)}{|N_i|} z_i^2 \qquad (11)$$

where $z_i = \sum_{j \in N_i} (x_i - x_j)$. The update rule (11) holds at the event times t_k^i corresponding to agent i: $e_i^2(t_k^i) = \frac{\sigma_i a(1 - a|N_i|)}{|N_i|} z_i^2(t_k^i)$, with $k = 0, 1, \ldots$ and $i \in \mathcal{N}$. At an event time t_k^i, we have $e_i(t_k^i) = x_i(t_k^i) - x_i(t_k^i) = 0$ and thus, condition (10) is enforced.

It should be emphasized that the condition (11) is verified by agent i only based on information of each own and neighboring agents' information, which is encoded by the vector $z_i = \sum_{j \in N_i} (x_i - x_j)$, which includes only the relative state information of agent i's neighbors, as is the case in multi-agent control designs.

The following convergence result regarding the convergence of the agents thus holds:

Theorem 3: Consider system $\dot{x} = u$ with the control law (9),(11) and assume that the communication graph G is connected. Then all agents are asymptotically stabilized to their initial average, i.e., $\lim_{t \to \infty} x_i(t) = \bar{x} = \frac{1}{N} \sum_i x_i(0)$ for all $i \in \mathcal{N}$.

A related result regarding the inter-event times holds in the distributed case as well:

Theorem 4: Consider system $\dot{x}_i = u_i$, $i \in \mathcal{N} = \{1, \ldots, N\}$ with the control law (9) and update ruling (11), and assume that G is connected. Suppose that $0 < a < \frac{1}{|N_i|}$ and $0 < \sigma_i < 1$ for all $i \in \mathcal{N}$. Then for any initial condition in $\mathbb{R}^N$, and any time $t \geq 0$ there exists at least one agent $k \in \mathcal{N}$ for which the next inter-event interval is strictly positive.

Proof: Assume that (11) holds for all $i \in \mathcal{N}$ at time t. If it doesn't hold, then continuous evolution is possible since at least one agent can still let its absolute measurement error increase without resetting (8). Hence assume that at t all errors are reset to zero. We will show that there exists at least one $k \in \mathcal{N}$ such that its next inter-event interval is bounded from below by a certain time $\tau_D > 0$. Denoting $k = \arg \max_i |z_i|$, and considering that $|e_i| \leq \|e\|$ holds for all i, we have $\frac{|e_k|}{N|z_k|} \leq \frac{\|e\|}{\|z\|}$ so that $\frac{|e_k|}{|z_k|} \leq N \frac{\|e\|}{\|z\|} = N \frac{\|e\|}{\|Lx\|}$. From the proof of Theorem 2 and the control update rule (11), we deduce that the next inter-event interval of agent k is bounded from below by a time τ_D that satisfies $N \frac{\tau_D}{1 - \tau_D \|L\|} = \frac{\sigma_k a(1 - a|N_k|)}{|N_k|}$ so that $\tau_D = \frac{\sigma_k a(1 - a|N_k|)}{N|N_k| + \|L\|\sigma_k a(1 - a|N_k|)}$ and the proof is complete. $\diamond$

Theorem 4 provides a lower bound on the inter-execution times of at least one agent. An analysis of the inter-execution times for all agents is provided at the end of the next section.

V. SELF-TRIGGERED MULTI-AGENT CONTROL

A. Self-triggered Control-centralized formulation

We now present a self-triggered control design for the agreement problem. In the event-triggered formulation, it becomes apparent that continuous monitoring of the measurement error norm is required to check condition (7). In the context of self-triggered control, this requirement is relaxed. Specifically, the next time t_{i+1} at which control law is updated is predetermined at the previous event time t_i and no state or error measurement is required in between the control updates. Such a self-triggered control design is presented in the following.

For $t \in [t_i, t_{i+1})$, $\dot{x}(t) = -L(x(t) + e(t))$. yields $x(t) = -Lx(t_i)(t - t_i) + x(t_i)$. Thus (6) can be rewritten as $\|x(t) - x(t_i)\| \leq \sigma \frac{\|Lx(t)\|}{\|L\|}$, or $\| - Lx(t_i)(t - t_i)\| \leq \sigma \frac{\| - L^2 x(t_i)(t - t_i) + Lx(t_i)\|}{\|L\|}$ or, equivalently $\|Lx(t_i)\|(t - t_i) \leq \frac{\sigma}{\|L\|} \|(-(t - t_i)L + I)Lx(t_i)\|$. An upper bound on the next execution time t_{i+1} is thus given by the solution t of $\|Lx(t_i)\|(t - t_i) = \frac{\sigma}{\|L\|} \|(-(t - t_i)L + I)Lx(t_i)\|$. Using the notation $\xi = t - t_i$, the latter is rewritten as $\|Lx(t_i)\|^2 \|L\|^2 \xi^2 = \sigma^2 (\|L^2 x(t_i)\|^2 \xi^2 + \|L^2 x(t_i)\|^2 - 2(Lx(t_i))^T LLx(t_i)\xi)$, or equivalently,

$$(\|Lx(t_i)\|^2 \|L\|^2 - \sigma^2 \|L^2 x(t_i)\|^2)\xi^2 + 2\sigma^2 (Lx(t_i))^T LLx(t_i)\xi$$
$$- \sigma^2 \|L^2 x(t_i)\|^2 = 0.$$

Note that $(\|Lx(t_i)\|^2 \|L\|^2 - \sigma^2 \|L^2 x(t_i)\|^2) > (1 -$

$\sigma^2)\|Lx(t_i)\|^2\|L\|^2 > 0$ and

$$\Delta = 4\sigma^4\|(Lx(t_i))^T LLx(t_i)\|^2 \\ + 4\sigma^2\|L^2 x(t_i)\|^2 \cdot (\|Lx(t_i)\|^2\|L\|^2 - \sigma^2\|L^2 x(t_i)\|^2) > 0.$$

An upper bound is then given by

$$t = t_i + \frac{-2\sigma^2(Lx(t_i))^T LLx(t_i) + \sqrt{\Delta}}{2(\|Lx(t_i)\|^2\|L\|^2 - \sigma^2\|L^2 x(t_i)\|^2)}. \quad (12)$$

Note that as long as $Lx(t_i) \neq 0$, i.e., agreement has not been reached, $t - t_i$ is strictly positive, i.e., the inter-execution times are non-trivial. The preceding analysis, along with Theorem 1, yield the following result:

Theorem 5: Consider system $\dot{x} = u$ with the control law (5) and assume that the communication graph G is connected. Suppose that $0 < \sigma < 1$. Assume that for each $i = 1, 2, \dots$ the next update time is chosen such that the bound

$$t_{i+1} - t_i \leq \frac{-2\sigma^2(Lx(t_i))^T LLx(t_i) + \sqrt{\Delta}}{2(\|Lx(t_i)\|^2\|L\|^2 - \sigma^2\|L^2 x(t_i)\|^2)} \quad (13)$$

holds. Then for any initial condition in $\mathbb{R}^N$ all agents asymptotically converge to their initial average, i.e., $\lim_{t\to\infty} x_i(t) = \bar{x} = \frac{1}{N}\sum_i x_i(0), \quad \forall i \in \mathcal{N}$.

B. Distributed Self-triggered Control

Similarly to the centralized case, continuous monitoring of the measurement error norm is required to check condition (11) in the distributed case. In the self-triggered setup, the next time t_{k+1}^i at which control law is updated is predetermined at the previous event time t_k^i and no state or error measurement is required in between the control updates. Such a distributed self-triggered control design is presented below.

Define $\beta_i = \frac{\sigma_i a(1 - a|N_i|)}{|N_i|}$. Then, (10) is rewritten as $|x_i(t_k^i) - x_i(t)|^2 \leq \beta_i z_i^2(t)$. Since $\dot{x}_i(t) = -\sum_{j \in N_i} (x_i(t_k^i) - x_j(t_{k'}^j))$, we get $x_i(t) = -\sum_{j \in N_i} (x_i(t_k^i) - x_j(t_{k'}^j))(t - t_k^i) + x_i(t_k^i)$ for $t \in [t_k^i, \min\{t_{k+1}^i, \min_{j \in N_i} t_{k''}^j\})$, where $k'' \triangleq \arg\min_{l \in \mathbb{N}: t_k^i \leq t_l^j} \{t_l^j - t_k^i\}$, and hence $\min\{t_{k+1}^i, \min_{j \in N_i} t_{k''}^j\}$ is the next time when the control u_i is updated. Thus (10) is equivalent to

$$\left| \sum_{j \in N_i} (x_i(t_k^i) - x_j(t_{k'}^j))(t - t_k^i) \right|^2 \leq \beta_i z_i^2(t). \quad (14)$$

Recalling $z_i(t) = \sum_{j \in N_i} (x_i(t) - x_j(t))$, we also have $x_j(t) = -\sum_{l \in N_j} (x_j(t_{k'}^j) - x_l(t_{k'''}^l))(t - t_{k'}^j) + x_j(t_{k'}^j)$, where $k''' = k'''(t) \triangleq \arg\min_{m \in \mathbb{N}: t \geq t_m^l} \{t - t_m^l\}$. Denote now $\sum_{j \in N_i} (x_i(t_k^i) - x_j(t_{k'}^j)) = \rho_i$, $\sum_{l \in N_j} (x_j(t_{k'}^j) - x_l(t_{k'''}^l)) = \rho_j$, and $\xi_i = \xi_i(t) = t - t_k^i$, $t \geq t_k^i$. Thus ρ_i, ρ_j are constants whereas ξ_i is a function of time t. We can now compute

$$z_i(t) = \sum_{j \in N_i} (x_i(t) - x_j(t)) \\ = \sum_{j \in N_i} (-\rho_i \xi_i + x_i(t_k^i)) - \sum_{j \in N_i} (-\rho_j(t - t_{k'}^j) + x_j(t_{k'}^j)) \\ = -|N_i|\rho_i \xi_i + |N_i| x_i(t_k^i) \\ + \sum_{j \in N_i} (\rho_j(t - t_k^i + t_k^i - t_{k'}^j) - x_j(t_{k'}^j)),$$

or equivalently,

$$z_i(t) = (-|N_i|\rho_i + \sum_{j \in N_i} \rho_j)\xi_i + |N_i|\rho_i + \sum_{j \in N_i} (\rho_j(t_k^i - t_{k'}^j)).$$

where the dependence of z_i on t is encoded in the evolution of $\xi_i(t)$ with respect to time. Further denoting $P_i = -|N_i|\rho_i + \sum_{j \in N_i} \rho_j$ and $\Phi_i = |N_i|\rho_i + \sum_{j \in N_i} (\rho_j(t_k^i - t_{k'}^j))$, the condition (14) can be rewritten as $|\rho_i \xi_i| \leq \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$ and since $\xi_i \geq 0$, the latter is equivalent to

$$|\rho_i|\xi_i \leq \sqrt{\beta_i}|P_i \xi_i + \Phi_i|. \quad (15)$$

Note that this inequality always holds for $\xi_i = 0$. Also note that (14) may or may not hold for all $\xi_i \geq 0$, and this can be decided by agent i at time t_k^i. Based on this observation, the self-triggered policy for agent i at time t_k^i is defined as follows: if there is a $\xi_i \geq 0$ such that $|\rho_i|\xi_i = \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$ then the next update time t_{k+1}^i takes place at most ξ_i time units after t_k^i, i.e., $t_{k+1}^i \leq t_k^i + \xi_i$. Of course if there is an update in one of its neighbors, thus updating the control law (9), then agent i re-checks the condition. Otherwise, if the inequality $|\rho_i|\xi_i \leq \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$ holds for all $\xi_i \geq 0$, then agent i waits until the next update of the control law of one of its neighbors to re-check this condition.

The self-triggered ruling for each agent i is thus summarized as:

Definition 6: For each $i = 1, 2, \dots$ the self-triggered ruling defines the next update time as follows: if there is a $\xi_i \geq 0$ such that $|\rho_i|\xi_i = \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$ then the next update time t_{k+1}^i takes place at most ξ_i time units after t_k^i, i.e., $t_{k+1}^i \leq t_k^i + \xi_i$. Agent i also checks this condition whenever its control law is updated due an update of the error of one of its neighbors. Otherwise, if the inequality $|\rho_i|\xi_i \leq \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$ holds for all $\xi_i \geq 0$, then agent i waits until the next update of the control law of one of its neighbors to re-check this condition.

The preceding analysis, along with Theorem 3, yield the following result:

Theorem 7: Consider system $\dot{x} = u$ with the control law (9) and assume that the communication graph G is connected. Suppose that $0 < a < \frac{1}{|N_i|}$ and $0 < \sigma_i < 1$ for all $i \in \mathcal{N}$. Assume that for each $i = 1, 2, \dots$ the next update time is decided according to Definition 6. Then, for any initial condition in $\mathbb{R}^N$, the states of all agents converge to their initial average, i.e., $\lim_{t\to\infty} x_i(t) = \bar{x} = \frac{1}{N}\sum_i x_i(0)$ for all $i \in \mathcal{N}$.

Note that after simple calculations it is easily derived that $\Phi_i = z_i(t_k^i)$. From (15), we know that the next event for agent i occurs at a time t when the equation $|\rho_i|(t - t_k^i) = \sqrt{\beta_i}|P_i(t - t_k^i) + z_i(t_k^i)|$ holds. Thus a zero inter-execution time for agent i can only occur when $|z_i(t_k^i)| = 0$. By virtue of Theorem 7, the system is asymptotically stabilized to the initial average. By the Cauchy-Schwartz inequality, we have $\|z\|^2 = \|Lx\|^2 = |\sum_i \sum_{j \in N_i} (x_i - x_j)|^2 \leq \frac{1}{2} x^T Lx = V$, so that z asymptotically converges to zero. Unfortunately there is no guarantee that no element of z will reach zero in finite time (or be equal to zero initially), however, as shown above, the inter-execution time can only be zero when $z_i = 0$ for agent i, i.e., when agent i has already reached its control objective.

We can now make some further calculations regarding the inter-execution times for each agent, assuming that $\Phi_i = z_i(t_k^i) \neq 0$. By taking the squares of both sides of $|\rho_i|\xi_i = \sqrt{\beta_i}|P_i \xi_i + \Phi_i|$, we have the following calculation for the next inter-execution time: for $|\rho_i|^2 - \beta_i P_i^2 \neq 0$, we have $\xi_i = \frac{\sqrt{\beta_i}\Phi_i}{|\rho_i| \pm \sqrt{\beta_i}P_i}$. So ξ_i is the smallest positive solution, provided that at least one of the non-zero numbers $\frac{\sqrt{\beta_i}\Phi_i}{|\rho_i| + \sqrt{\beta_i}P_i}, \frac{\sqrt{\beta_i}\Phi_i}{|\rho_i| - \sqrt{\beta_i}P_i}$ is positive. Otherwise the strict inequality

$|\rho_i|\xi_i < \sqrt{\beta_i}|P_i\xi_i + \Phi_i|$ holds for all $\xi_i \geq 0$. For $|\rho_i|^2 - \beta_i P_i^2 = 0$, then no solution is feasible for $P_i = 0$ and $\Phi_i = z_i(t_k^i) \neq 0$. For $P_i \neq 0$ we have $\xi_i = \dfrac{-\Phi_i}{2P_i}$ which yields a feasible (non-negative)solution, if $\Phi_i P_i > 0$.

In all cases when $|\rho_i|\xi_i = \sqrt{\beta_i}|P_i\xi_i + \Phi_i|$ has a feasible solution, this is analogous to $\Phi_i = z_i(t_k^i)$, as expected. Thus the only occurence of zero inter-execution times can happen when $z_i(t_k^i) = 0$, i.e., when the control objective has been achieved by agent i.

VI. EXAMPLES

The results of the previous Sections are illustrated through computer simulations.

Consider a network of four agents whose Laplacian matrix is given by

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

The four agents start from random initial conditions and evolve under the control (5) in the centralized case, and the control (9) in the distributed case. We have set $\sigma = 0.65$, and $\sigma_1 = \sigma_2 = 0.55$, $\sigma_3 = \sigma_4 = 0.75$ and $a = 0.2$. In both cases, we consider two different cases of actuation updates: the event-triggered and the self-triggered one.

Figure 1 shows the evolution of the error norm in the centralized case. The top plot represents the event-triggered and the bottom the self-triggered formulation. In the event-triggered case, the control law is updated according to Theorem 1 and in the self-triggered according to Theorem 5. The solid line represents the evolution of $\|e(t)\|$. This stays in both plots below the specified state-dependent threshold $\|e\|_{max} = \sigma \dfrac{\|Lx\|}{\|L\|}$ which is represented by the dotted line in the Figure.
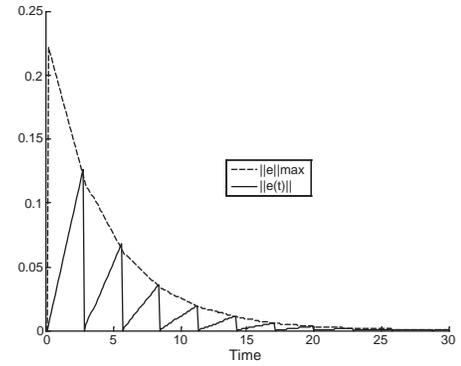
The next simulation depicts how the framework is realized in the distributed case for agent 1. In particular, the solid line in Figure 2 shows the evolution of $|e_1(t)|$. This stays below the specified state-dependent threshold given by (10) $|e_1|_{max} = \sqrt{\frac{\sigma_1 a(1-a|N_1|)}{|N_1|}} z_1$ which is represented by the dotted line in the Figure. Once again, the top plot shows the event-triggered case of Theorem 3 and the bottom plot the self-triggered case of Theorem 7.

In both cases, it can be seen that the event-triggered case requires fewer controller updates. On the other hand, the self triggered approach seems more robust, since the design provides an upper bound on the interval in which the update should be held.
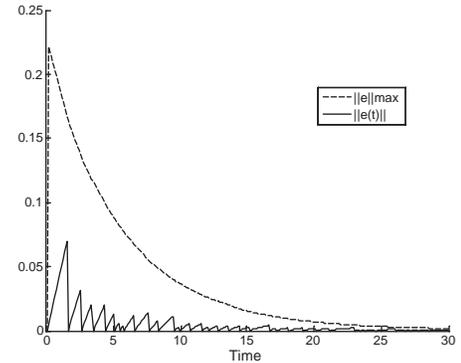
VII. CONCLUSIONS

We considered event-driven strategies for multi-agent systems. The actuation updates were event-driven, depending on the ratio of a certain measurement error with respect to the norm of a function of the state. A centralized formulation of the problem was considered first and then the results were extended to the distributed counterpart, in which agents required knowledge only of the states of their neighbors for the controller implementation. The results of the paper were supported through simulated examples.

Future work will focus on the performance analysis of the framework and its application to other cooperative multi-agent control tasks. Moreover, while the event-triggered formulation of the current paper focuses on the reduction of actuator updates, it is also interesting to consider sensing limitations in this case. Finally, current research also involves the case when it is the responsibility of each agent to broadcast information to its neighbors rather than requesting information from them, as in [30].
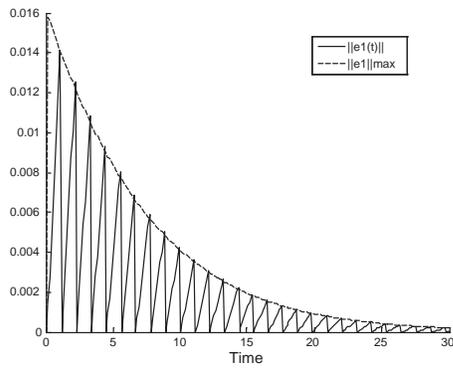

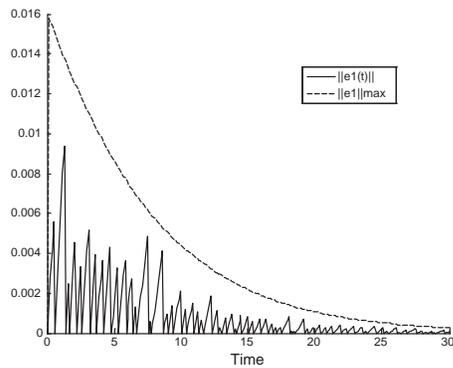(a) Event-triggered case

(b) Self-triggered case

Fig. 1. Four agents evolve under the centralized event-triggered (top plot) and self-triggered (bottom plot) proposed framework.

REFERENCES

[1] A.Anta and P.Tabuada. To sample or not to sample: self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030 –2042, 2010.

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

[3] K.J. Astrom and B. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. *41st IEEE Conference on Decision and Control*, pages 2011–2016, 2002.

[4] M. Cao, B.D.O. Anderson, A.S. Morse, and C. Yu. Control of acyclic formations of mobile autonomous agents. *47th IEEE Conference on Decision and Control*, pages 1187–1192, 2008.

[5] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5):1343–1349, 2008.

[6] D.V. Dimarogonas, E. Frazzoli, and K.H. Johansson. Distributed self-triggered control for multi-agent systems. *49th IEEE Conf. Decision and Control*, 2010. to appear.

[7] D.V. Dimarogonas and K.H. Johansson. Event-triggered control for multi-agent systems. *48th IEEE Conf. Decision and Control*, pages 7131–7136, 2009.

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

[9] J.A. Fax and R.M. Murray. Graph Laplacians and stabilization of vehicle formations. *15th IFAC World Congress*, 2002.

[10] L. Grüne and F. Müller. An algorithm for event-based optimal feedback control. *48th IEEE Conf. Decision and Control*, pages 5311–5316, 2009.

[11] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, 2007.

[12] M. Ji and M. Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

(a) Distributed event-triggered case

(b) Distributed self-triggered case

Fig. 2. Four agents evolve under the distributed event-triggered (top plot) and self-triggered (bottom plot) proposed framework.

[13] E. Johannesson, T. Henningsson, and A. Cervin. Sporadic control of first-order linear stochastic systems. *Hybrid Systems: Computation and Control*, pages 301–314, 2007.

[14] D.B. Kingston, W. Ren, and R. Beard. Consensus algorithms are input-to-state stable. *American Control Conference*, pages 1686–1690, 2005.

[15] D. Lehmann and J. Lunze. Event-based control: A state feedback approach. *European Control Conference*, pages 1716–1721, 2009.

[16] P. Lin and Y. Mia. Average consensus in networks of multi-agents with both switching topology and coupling time-delay. *Physica A*, 387:303–313, 2008.

[17] X. Liu, W. Lu, and T. Chen. Consensus of multi-agent systems with unbounded time-varying delays. *IEEE Transactions on Automatic Control*, 55(10):2396–2401, 2010.

[18] M. Mazo, A. Anta, and P. Tabuada. On self-triggered control for linear systems: Guarantees and complexity. *European Control Conference*, 2009.

[19] U. Munz, A. Papachristodoulou, and F. Allgower. Delay robustness in consensus problems. *Automatica*, 46(8):1252–1265, 2010.

[20] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[21] R. Olfati-Saber and J.S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *44th IEEE Conference on Decision and Control*, pages 6698–6703, 2005.

[22] M. Rabi, K.H. Johansson, and M. Johansson. Optimal stopping for event-triggered sensing and actuation. *47th IEEE Conference on Decision and Control*, pages 3607–3612, 2008.

[23] W. Ren and E.M. Atkins. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control*, 17(10-11):1002–1033, 2007.

[24] E.D. Sontag. On the input-to-state stability property. *European Journal of Control*, 1:24–36, 1995.

[25] A. Speranzon, C. Fischione, and K.H. Johansson. Distributed and collaborative estimation over wireless sensor networks. *45th IEEE Conference on Decision and Control*, pages 1025–1030, 2006.

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

[27] J.N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Inst. of Technology, 1984.

[28] X. Wang and M.D. Lemmon. Event-triggered broadcasting across distributed networked control systems. *American Control Conference*, 2008.

[29] X. Wang and M.D. Lemmon. Self-triggered feedback control systems with finite-gain L2 stability. *IEEE Transactions on Automatic Control*, 45(3):452–467, 2009.

[30] X. Wang and M.D. Lemmon. Event-triggering in distributed networked control systems. *IEEE Transactions on Automatic Control*, 56(3):586–601, 2011.