Zeno-free, distributed event-triggered communication and control for multi-agent average consensus

Cameron Nowzari   Jorge Cortés

***Abstract*— This paper studies a distributed event-triggered communication and control strategy that solves the multi-agent average consensus problem. The proposed strategy does not rely on the continuous or periodic availability of information to an agent about the state of its neighbors, but instead prescribes isolated event times where both communication and controller updates occur. In addition, all parameters required for its implementation can be locally determined by the agents. We show that the resulting network executions are guaranteed to converge to the average of the initial agents' states, establish that events cannot be triggered an infinite number of times in any finite time period (i.e., no Zeno behavior), and characterize the exponential rate of convergence. We also provide sufficient conditions for convergence in scenarios with time-varying communication topologies. Simulations illustrate our results.**

I. Introduction

This paper considers the multi-agent average consensus problem, where a group of agents sets out to agree on the average of their initial states. Many solutions exist in the literature that rely on the continuous or periodic availability of information to the agents about the state of neighboring agents, and the synchronous execution of the strategies. Unfortunately, the continuous availability of information leads to inefficient implementations in terms of energy, communication bandwidth, congestion, and processor usage. Motivated by these observations, our main goal in this paper is the design of a real-time distributed coordination strategy that prescribes isolated events for when communication should occur that still ensures the resulting asynchronous network executions achieve average consensus.

Literature review: Event-triggered control is aimed at tuning controller executions to the state evolution of a given system, see e.g., [1], [2]. In line with this idea, in the context of multi-agent scenarios, an increasing body of work seeks to trade computation and decision making at the agent level for less communication, sensing, or actuator effort while still guaranteeing a desired level of performance. The work [3] specifies the responsibility of each agent in updating the control signals and [4] considers network scenarios with disturbances, communication delays, and packet drops. In addition to deciding when controllers should be updated, several works have also explored the application of event-triggered ideas to the acquisition of information, be it through either communication or sensing. To this end, [5], [6], [7] combine event-triggered controller updates with sampled data that allows for the periodic evaluation of the triggers. Other works like [8] even drop the need for periodic access to information by considering event-based broadcasts, where agents decide with local information only when to share information with neighbors. Self-triggered control [9], [10] relaxes the need for local information by deciding when a future sample of the state should be taken based on the available information from the last sampled state.

Regarding average consensus, the available literature is ample, see e.g., [11], [12], [13]. A continuous-time algorithm that achieves asymptotic convergence to average consensus for both undirected and weight-balanced directed graphs is introduced in [14]. The work [15] builds on this algorithm to propose a Lyapunov-based event-triggered strategy that dictates when agents should update their control signals. However, it requires each agent to have perfect information about their neighbors at all times. The work [16] uses event-triggered broadcasting with time-dependent triggering functions to provide an algorithm where each agent only requires exact information about itself, rather than its neighbors. However, its implementation requires knowledge of the algebraic connectivity of the network. In addition, the strictly time-dependent nature of the triggers decouples the network evolution from the actual state of the agents. Closest to the treatment of this paper, [17] proposes an event-triggered broadcasting law with state-dependent triggering functions where agents do not rely on the availability of continuous information about their neighbors – under the assumption that all agents have initial access to a common parameter. This algorithm guarantees that all inter-event times are strictly positive, but does not discard the possibility of an infinite number of events happening in a finite time period.

Statement of contributions: In this paper we propose a novel event-triggered broadcasting and controller update strategy that relies only on information available to the agents. This fully distributed communication and control strategy can be implemented without any a priori or online global knowledge about the network. This is in contrast to prior works where agents might need to know either a global property such as the algebraic connectivity of the entire network or set a parameter all to the same value. We show that our naturally asynchronous algorithm still ensures that all agent states converge to the initial average of all agents given a connected, undirected communication topology. We also show that there exists a finite number of broadcasts and

Cameron Nowzari and Jorge Cortés are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, {cnowzari,cortes}@ucsd.edu.

updates by each agent in any finite time period ensuring that Zeno behavior does not occur in the system. We are also able to characterize a lower bound on the exponential convergence rate of the algorithm. Lastly, we provide sufficient conditions for time-varying topologies such that convergence to the desired consensus state is still guaranteed. We demonstrate our results through simulations. For reasons of space, some proofs are omitted and will appear elsewhere.

Organization: Section II introduces basic notation and reviews some graph-theoretic concepts. Section III formulates the problem of interest. Section IV discusses the design of the event-triggered communication and control law and Section V characterizes its properties. We illustrate our results via simulations in Section VI. Finally, Section VII contains our concluding remarks and ideas for future work.

II. Preliminaries

We denote by $\mathbb{R}$ the set of real numbers. We let $\mathbf{1}_N \in \mathbb{R}^N$ denote the column vector with entries are all equal to one. The Euclidean norm on $\mathbb{R}^N$ is denoted by $\|\cdot\|$. Given a finite set S, we let $|S|$ denote its cardinality. Given $x \in \mathbb{R}$, we let sgn denote the sign function,

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{for} \quad x > 0 \\ -1 & \text{for} \quad x < 0 \\ 0 & \quad\text{otherwise.} \end{cases}$$

Given any $x, y \in \mathbb{R}$, Young's inequality [18] states that for any $\varepsilon > 0$,

$$xy \leq \frac{x^2}{2\varepsilon} + \frac{\varepsilon y^2}{2}. \tag{1}$$

A graph $\mathcal{G} = (V, E)$ is comprised of a set of vertices $V = \{1, \ldots, N\}$ and edges $E \subset V \times V$. The graph $\mathcal{G}$ is undirected if for any edge $(i, j) \in E$, the edge $(j, i) \in E$ also. An edge $(i, j) \in E$ means that vertex j is a neighbor of i. The set of neighbors of a given node i is given by $\mathcal{N}_i$. The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is defined by $a_{ij} = 1$ if $(i, j) \in E$ and $a_{ij} = 0$ otherwise. A path from vertex i to j is an ordered sequence of vertices such that each intermediate pair of vertices is an edge. An undirected graph $\mathcal{G}$ is connected if there exists a path from all $i \in V$ to all $j \in V$. The degree matrix D is a diagonal matrix where $d_{ii} = |\mathcal{N}_i|$. The Laplacian matrix is defined as $L = D - A$. For undirected graphs the Laplacian is symmetric $L = L^T$ and positive semidefinite. If the graph $\mathcal{G}$ is connected, the Laplacian has exactly one eigenvalue at 0 (with associated eigenvector $\mathbf{1}_N$) with the rest strictly positive, $0 = \lambda_1(L) < \lambda_2(L) \leq \cdots \leq \lambda_N(L)$. The following inequality will be useful later,

$$\lambda_2(L) x^T L x \leq x^T L^2 x \leq \lambda_N(L) x^T L x. \tag{2}$$

This property follows by noting that L is diagonalizable, and hence can be written as $L = S^{-1} D S$, where D is a diagonal matrix containing its eigenvalues. Then, using $\lambda_2(L) D \leq D^2 \leq \lambda_N(L) D$, equation (2) follows.

III. Problem statement

We consider the average consensus problem involving a network of N agents. We let $\mathcal{G}$ denote the connected, undirected graph in which neighbors of the graph are agents that are able to communicate with one another wirelessly. We denote by $x_i \in \mathbb{R}$ the state of agent $i \in \{1, \ldots, N\}$. We consider single-integrator dynamics

$$\dot{x}_i(t) = u_i(t), \tag{3}$$

for all $i \in \{1, \ldots, N\}$. It is well known [14] that the distributed continuous control law

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t)) \tag{4}$$

drives each agent of the system to asymptotically converge to the average of the agents' initial conditions. In compact form, this can be expressed by

$$\dot{x} = -Lx,$$

where $x = (x_1, \ldots, x_N)$ is the column vector of all agent states and L is the Laplacian of $\mathcal{G}$. However, in order to be implemented, this control law requires each agent to continuously have information about its neighbors and continuously update its control law. In this paper we are interested in implementations that relax both of these requirements.

According to the model, an agent $i \in \{1, \ldots, N\}$ is only able to communicate with its neighbors $\mathcal{N}_i$ in the graph $\mathcal{G}$. Neighbors of agent i only receive state information from it when agent i decides to broadcast its state to them. We denote by $\widehat{x}_i(t)$ the last broadcast state of agent i at any given time t. We assume that each agent i has continuous access to its own state. We then utilize an event-triggered implementation of the controller (4) given by

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\widehat{x}_i(t) - \widehat{x}_j(t)). \tag{5}$$

Note that although agent i has access to its own state $x_i(t)$, the controller (5) uses the last broadcast state $\widehat{x}_i(t)$. This is to ensure that the average of the agents' initial states is preserved throughout the evolution of the system. More specifically, utilizing this controller, one has

$$\frac{d}{dt}(\mathbf{1}_N^T x(t)) = \mathbf{1}_N^T \dot{x}(t) = \mathbf{1}_N^T L \widehat{x}(t) = 0, \tag{6}$$

where $\widehat{x} = (\widehat{x}_1, \ldots, \widehat{x}_N)$ and we have used the fact that L is symmetric and $L \mathbf{1}_N = 0$.

The purpose of this paper is to identify triggers or conditions to prescribe when each agent should broadcast its state to its neighbors with the ultimate objective of making the network converge to the average of the initial agents' states.

IV. Distributed trigger design

In this section we synthesize distributed triggers that prescribe when agents should broadcast state information and update their control signals. Section IV-A studies the evolution of the network disagreement to identify a triggering function and discusses the implementation problems of the associated agent executions. These observations are our starting point in Section IV-B, where we develop a refined trigger design that overcomes these implementation issues.

A. Rationale for primary triggering function

Our exposition here builds on the discussion in [15], [17]. With respect to [15], the ensuing design has the advantage of not requiring agents to have continuous information about their neighbors at all times. This advantage is shared by the design in [17], which still requires all network agents to have knowledge of an a priori chosen common parameter $a > 0$. The triggering function identified here drops this requirement allowing a fully distributed initialization and implementation.

Consider the candidate Lyapunov function

$$V(x) = \frac{1}{2} x^T L x. \tag{7}$$

The Lie derivative of V under the control law (5) is

$$\dot{V} = x^T L \dot{x} = -x^T L (L\widehat{x}).$$

For $i \in \{1, \dots, N\}$, let $e_i(t) = \widehat{x}_i(t) - x_i(t)$ be the error between agent i's last broadcast state and its true current state. Let $e = (e_1, \dots, e_N)$ be the vector of errors of all network agents. Using the fact that L is symmetric,

$$\begin{aligned} \dot{V} &= -(\widehat{x}^T - e^T) L L \widehat{x} \\ &= -\|L\widehat{x}\|^2 + (L\widehat{x})^T L e. \end{aligned} \tag{8}$$

Letting $\widehat{z} = L\widehat{x} = (\widehat{z}_1, \dots, \widehat{z}_N)$, we can expand (8) as

$$\begin{aligned} \dot{V} &= -\sum_{i=1}^N \widehat{z}_i^2 + \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \widehat{z}_i (e_i - e_j) \\ &= -\sum_{i=1}^N \widehat{z}_i^2 + \sum_{i=1}^N |\mathcal{N}_i| \widehat{z}_i e_i - \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \widehat{z}_i e_j. \end{aligned}$$

Next, we bound the cross-terms in the second and third summands. Given $a_1, \dots, a_N > 0$, we use Young's inequality (1) (with $\varepsilon = a_i$) to get

$$\sum_{i=1}^N |\mathcal{N}_i| \widehat{z}_i e_i \le \sum_{i=1}^N \left(\frac{1}{2} |\mathcal{N}_i| \widehat{z}_i^2 a_i + \frac{1}{2a_i} |\mathcal{N}_i| e_i^2 \right),$$

and (with $\varepsilon = a_j$)

$$-\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \widehat{z}_i e_j \le \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \left(\frac{1}{2} \widehat{z}_i^2 a_j + \frac{1}{2a_j} e_j^2 \right).$$

Using that $\mathcal{G}$ is undirected, we can rewrite the last term as

$$\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \frac{1}{2a_j} e_j^2 = \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \frac{1}{2a_i} e_i^2 = \sum_{i=1}^N \frac{1}{2a_i} |\mathcal{N}_i| e_i^2.$$

Substituting these inequalities in (8), we obtain

$$\dot{V} \le \sum_{i=1}^N \left(\Big(\frac{1}{2} a_i |\mathcal{N}_i| + \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j - 1 \Big) \widehat{z}_i^2 + \frac{|\mathcal{N}_i|}{a_i} e_i^2 \right).$$

Note that the coefficient of e_i^2 is always nonnegative. To ensure the coefficient $\widehat{z}_i^2$ is not positive, we choose

$$a_i < \frac{1}{\max_{j \in \mathcal{N}_i \cup \{i\}} |\mathcal{N}_j|}, \tag{9}$$

for all $i \in \{1, \dots, N\}$ so that

$$1 - \frac{1}{2} a_i |\mathcal{N}_i| - \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j > 0.$$

As a consequence of the above analysis, we define the triggering function for each $i \in \{1, \dots, N\}$ as

$$f_i(e_i) = e_i^2 - \sigma_i \frac{a_i}{|\mathcal{N}_i|} \Big(1 - \frac{1}{2} a_i |\mathcal{N}_i| - \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j \Big) \widehat{z}_i^2, \tag{10}$$

where $\sigma_i \in (0, 1)$. Note that if each agent i enforces

$$f_i(e_i) \le 0, \tag{11}$$

then

$$\dot{V} \le \sum_{i=1}^N (\sigma_i - 1) \Big(1 - \frac{1}{2} a_i |\mathcal{N}_i| - \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_j \Big) \widehat{z}_i^2 \tag{12}$$

is strictly negative for all $\widehat{z} \neq 0$. When the event $f_i(e_i) = 0$ occurs, it would seem natural to prescribe agent i to broadcast its current state x_i to its neighbors in order to avoid having the Lie derivative of V becoming positive. However, such a trigger would be subject to the following problems:

- (P1) The discontinuous nature of $\widehat{z}_i$ might make an agent completely "miss" this trigger when a jump in $\widehat{z}_i$ occurs. Such jumps are due to a neighboring agent broadcasting a new state to agent i. It could very well be the case that just before the update was received, $f_i(e_i) < 0$, and immediately after, $f_i(e_i) > 0$.
- (P2) The equality $f_i(e_i) = 0$ might still hold even after agent i broadcasts its new state to its neighbors. This would happen if agent i's last broadcast state is in agreement with the states received from its neighbors, making $\widehat{z}_i = 0$, hence causing the agent to broadcast its state continuously.
- (P3) Even if the trigger is never missed due to jumps in $\widehat{z}_i$, successive jumps (with a finite accumulation point in time) could cause Zeno behavior to occur.

These observations motivate our refinement of the trigger defined by the function (10) explained next.

B. Rationale for refined triggering functions

Rather than prescribing agent $i \in \{1, \dots, N\}$ to broadcast its state when $f_i(e_i) = 0$, we instead define an event by

$$f_i(e_i) > 0, \tag{13}$$

or

$$f_i(e_i) = 0, \quad \widehat{z}_i \neq 0 \quad \text{and} \quad \operatorname{sgn}(e_i) = \operatorname{sgn}(\widehat{z}_i). \tag{14}$$

The reasoning behind these triggers is the following. The inequality (13) makes sure that the discontinuities of $\widehat{z}_i$ do not make the agent miss an event (cf. problem P1 above). The trigger (14) makes sure that the agent is not required to continuously broadcast its state to neighbors when its last broadcast state is in agreement with the states received from them (cf. problem P2 above). Given that $\dot{e}_i = -\dot{x}_i = \widehat{z}_i$, if $f_i(e_i) = 0$ and $\widehat{z}_i = 0$ hold (and hence $e_i = 0$ holds too), then it is not necessary for agent i to broadcast its state, because (11) will hold until new information comes in from its neighbors. This argument explains the inequality $\widehat{z}_i \neq 0$ in (14). Finally, if $f_i(e_i) = 0$ and $\widehat{z}_i \neq 0$ hold, then one can see from the definition of the triggering function (10) that the time derivative of $f_i(e_i)$ is positive if and only if $\operatorname{sgn}(e_i) = \operatorname{sgn}(\widehat{z}_i)$, which explains the inclusion of the last equality in (14).

Finally, to address problem P3 above, we prescribe the following additional trigger. If at some time $t \geq t^i_{\text{last}}$ (here, t^i_{last} is the last time at which agent $i \in \{1, \dots, N\}$ broadcast information to its neighbor(s)), agent i receives new information from a neighbor $j \in \mathcal{N}_i$, i will immediately broadcast its state if

$$t < t^i_{\text{last}} + \varepsilon_i, \tag{15}$$

where

$$\varepsilon_i < \sqrt{\sigma_i \frac{a_i}{|\mathcal{N}_i|}\left(1 - \frac{1}{2}a_i|\mathcal{N}_i| - \frac{1}{2}\sum_{j \in \mathcal{N}_i} a_j\right)}. \tag{16}$$

Our analysis in Section V will expand on the reasons this trigger helps the network prevent the occurrence of Zeno behavior.

The triggers (13)-(15) form the basis of the EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW, which is formally presented in Table I. This law requires each agent $i \in \{1, \dots, N\}$ to initially choose a_i according to (9) and share this value with its neighbors, so that ε_i satisfying (16) can be selected too. Once this is done, agents do not communicate among them or update their control signals in between events. Each time an event is triggered by an agent, say $i \in \{1, \dots, N\}$, that agent broadcasts its current state to its neighbors and updates its control signal, while its neighbors $\mathcal{N}_i$ update their control signal. This is in contrast to other event-triggered designs that prescribe updates of the control signals but require continuous communication among the agents to provide them with the necessary information to check the triggers.

At all times t agent $i \in \{1, \dots, N\}$ performs:

if $f_i(e_i(t)) > 0$ **or** ($f_i(e_i(t)) = 0$, $\widehat{z}_i(t) \neq 0$, and $\operatorname{sgn}(e_i(t)) = \operatorname{sgn}(\widehat{z}_i(t))$) **then**
  broadcast state information $x_i(t)$ and update control signal
end if
if new information $x_j(t)$ is received from some neighbor(s) $j \in \mathcal{N}_i$ **then**
  if agent i has broadcast its state in the last ε_i seconds **then**
    broadcast state information $x_i(t)$
  end if
  update control signal
end if

TABLE I

EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW.

Remark IV.1 (Comparison with the event-triggered design in [17]) The trigger (13) is a generalization of the one in [17] to allow for possibly different parameters a_i that are locally determined by each agent. That work also implicitly considers the trigger (14), albeit the formulation proposed here is more in line with the requirements of real-time controller implementation. Finally, the trigger (15) is a novel addition with respect to [17] that plays a key role in guaranteeing that Zeno behavior does not arise in the executions of the proposed algorithm, as we show next. •

V. ANALYSIS OF THE EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW

Here we analyze the properties of the control law (5) in conjunction with the EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW proposed in Section IV. Specifically, we establish that Zeno behavior does not occur, prove convergence of the trajectories to the desired consensus state, and provide a lower bound on the convergence rate. First, we show that the network executions of EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW are guaranteed not to exhibit Zeno behavior. The proof of the following result strongly relies on the trigger (15) that is only checked at time instants when new information is received.

Proposition V.1 (No Zeno behavior) *Given the system* (3) *with control law* (5) *executing the* EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW *of Table I over a connected undirected graph, the agents will not be required to communicate an infinite number of times in any finite time period.*

Proof: We are interested in showing here that no agent will broadcast its state an infinite number of times in a finite time period. We begin by showing that if an agent i does not receive new information from neighbors, it will broadcast its state periodically with some period $\tau_i > 0$ as long as $\widehat{z}_i \neq 0$. Assume that agent i has just broadcast its state at time t_0, and thus $e_i(t_0) = 0$. If no new information is received for

$t \geq t_0$, the evolution of the error is simply

$$e_i(t) = \int_{t_0}^{t} \widehat{z}_i(s) ds = \widehat{z}_i(t_0)(t - t_0).$$

Note that if $\widehat{z}_i(t_0) = 0$, no broadcasts will ever happen because $e_i(t) = 0$ for all $t \geq t_0$. Also since we are now assuming no neighbors of i are broadcasting information, the trigger (15) is irrelevant. We are then interested in finding out the time t^* when (14) occurs, triggering a broadcast of agent i's state. Using the above description of the error, we rewrite the trigger (14) as

$$\widehat{z}_i(t_0)^2(t^* - t_0)^2 = \sigma_i \frac{a_i}{|\mathcal{N}_i|}\left(1 - \frac{1}{2}a_i|\mathcal{N}_i| - \frac{1}{2}\sum_{j\in\mathcal{N}_i} a_j\right)\widehat{z}_i(t_0)^2.$$

From this, it is clear to see that agent i will not broadcast its state for τ_i seconds where

$$\tau_i = t^* - t_0 = \sqrt{\sigma_i \frac{a_i}{|\mathcal{N}_i|}\left(1 - \frac{1}{2}a_i|\mathcal{N}_i| - \frac{1}{2}\sum_{j\in\mathcal{N}_i} a_j\right)} > 0.$$

We now show that messages cannot be sent an infinite number of times between agents in a finite time period. Again, let time t_0 be the time at which agent i has broadcast its information to neighbors and thus $e_i(t_0) = 0$. If no information is received by time $t_0 + \varepsilon_i < t_0 + \tau_i$ there is no problem, so we now consider the case that at least one neighbor of i broadcasts its information at some time $t_1 \in (t_0, t_0 + \varepsilon_i)$. In this case it means that at least one neighbor $j \in \mathcal{N}_i$ has broadcast new information, thus agent i would also rebroadcast its information at time t_1 due to trigger (15). Let I denote the set of all agents who have broadcast information at time t_1, we refer to these agents as synchronized. This means that as long as no agent $k \notin I$ sends new information to any agent in I, the agents in I will not broadcast new information for at least $\min_{j\in I} \tau_j$ seconds, which includes the original agent i. Similar to before, if no new information is received by any agent in I by time $t_1 + \min_{p\in I} \varepsilon_p$ there is no problem, so we now consider the case that at least one agent k sends new information to some agent $j \in I$ at time $t_2 \in (t_1, t_1 + \min_{p\in I} \varepsilon_p)$. By trigger (15), this would require all agents in I to also broadcast their state information at time t_2 and agent k will now be added to the set I. Reasoning repeatedly in this way, the only way for infinite communications to occur in a finite time period is for an infinite number of agents to be added to set I, which is clearly not possible. ■

Remark V.2 (Conditions for Zeno) The addition of (15) to the triggers (13)-(14) helps establish the lack of Zeno behavior of the resulting network executions. It is currently an open problem to show whether or not network executions with only the triggers (13)-(14) exhibit Zeno behavior. For such executions, the work [17, Corollary 2] guarantees that no agent would undergo an infinite number of updates at any given instant of time, but does not discard the possibility of an infinite number of updates in a finite time period, as Proposition V.1 does. •

Next, we state the main convergence result.

Theorem V.3 (Asymptotic convergence to initial average) *Given the system* (3) *with control law* (5) *executing the* EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW *of Table I over a connected undirected graph, all agents asymptotically converge to the average of the initial states,*

$$\lim_{t\to\infty} x_i(t) = \text{Ave}(x(0)) = \frac{1}{N}\sum_{j=1}^{N} x_j(0),$$

for each $i \in \{1, \dots, N\}$.

The next result provides a lower bound on the exponential convergence rate of the network.

Theorem V.4 (Convergence rate) *Given the system* (3) *with control law* (5) *executing the* EVENT-TRIGGERED COMMUNICATION AND CONTROL LAW *over an undirected graph, the system converges exponentially to the agreement space with at least a rate of*

$$(\sigma - 1)(1 - a\bar{N})\lambda_2(L),$$

where $\sigma = \max_{i\in\{1,\dots,N\}} \sigma_i$ *and* $\bar{N} = \max_{i\in\{1,\dots,N\}} |\mathcal{N}_i|$.

The following result gives a sufficient condition for convergence when the network topology is changing.

Proposition V.5 (Time-varying topologies) *In the case that the communication graph* $\mathcal{G}$ *is changing in time at discrete time instants, let* $\mathbf{L}$ *denote the set of all connected undirected graphs. If all agents are aware of who its neighbors are at each time, agents broadcast their state when their neighbors change, and*

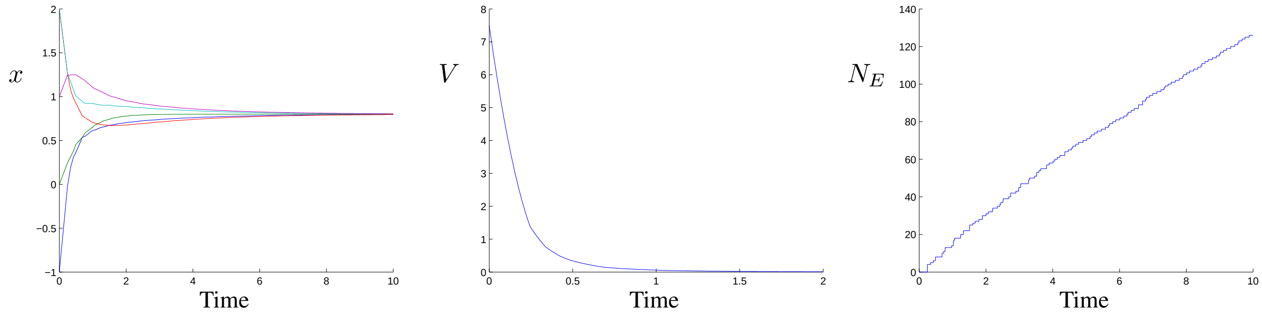
$$\max_{L\in\mathbf{L}} \lambda_N(L)\sigma D < 1,$$

then the agent states asymptotically converge to $\text{Ave}(x(0))$.

VI. Simulations

Here we demonstrate the effectiveness of the proposed algorithm with a simple simulation. We consider a system of $N = 5$ agents operating under the dynamics (3) with control law (5). The only difference is the definition of the event-triggers. In the simulation we use the fixed graph defined by the adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$



(a) (b) (c)

Fig. 1. Plots of (a) the trajectories of the agents' states, (b) the evolution of the Lyapunov function and (c) the total number of events triggered N_E throughout the execution of the proposed algorithm.

We set $a_1 = a_3 = a_5 = 0.3$, $a_2 = a_4 = 0.2$, and $\sigma_i = .999$ for all $i \in \{1, \dots, N\}$. The initial condition is set to $x(0) = [-1, 0, 2, 2, 1]$. Figure 1(a) shows the evolution of the trajectories of all agent states. Figure 1(b) shows the evolution of the Lyapunov function, demonstrating the exponential convergence of our algorithm. Figure 1(c) plots the total number of events triggered by the network agents.

VII. Conclusions

We have considered the multi-agent average consensus problem and studied in detail a distributed event-triggered strategy for communication and control. This strategy has several distinguishing features, including the fact that individual agents do not require continuous, or even periodic, information about the states of their neighbors, and the fact that all parameters required by its implementation can be locally determined by the agents. We have established several important properties of the network executions resulting from the implementation of the event-triggered law: asymptotic convergence to the initial average of the agents' states, absence of infinite updates in any finite time interval (lack of Zeno behavior), a lower bound on the exponential rate of convergence, and robustness to changes in the communication topology that maintain connectivity. Future work will be devoted to address the open question laid out in Remark V.2, tighter bounds on the exponential rate of convergence, as well as to study scenarios with directed communication topologies, more general agent dynamics, and physical sources of error, such as wireless communication delays and packet drops.

Acknowledgments

This research was supported by NSF award CNS-1329619.

References

[1] K. J. Åström and B. M. Bernhardsson., "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *IEEE Conf. on Decision and Control*, (Las Vegas, NV), pp. 2011–2016, Dec. 2002.

[2] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 3270–3285, 2012.

[3] M. Mazo Jr. and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[4] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586–601, 2011.

[5] G. Xie, H. Liu, L. Wang, and Y. Jia, "Consensus in networked multi-agent systems via sampled control: fixed topology case," in *American Control Conference*, (St. Louis, MO), pp. 3902–3907, 2009.

[6] W. P. M. H. Heemels and M. C. F. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.

[7] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125–2132, 2013.

[8] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, pp. 2735–2750, 2010.

[9] A. Anta and P. Tabuada, "To sample or not to sample: self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.

[10] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite-gain L_2 stability," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 452–467, 2009.

[11] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[12] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Communications and Control Engineering, Springer, 2008.

[13] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Applied Mathematics Series, Princeton University Press, 2010.

[14] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[15] 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.

[16] G. S. Seybotha, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

[17] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013.

[18] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*. Cambridge, UK: Cambridge University Press, 1952.