

# A class of event-triggered coordination algorithms for multi-agent systems on weight-balanced digraphs

Ping Xu      Cameron Nowzari      Zhi Tian

**Abstract**—This paper revisits the multi-agent average consensus problem on weight-balanced directed graphs. In order to reduce communication among the agents, many recent works have considered event-triggered communication and control as a method to reduce communication while still ensuring that the entire network converges to the desired state. One common way to do this is to design events such that a specifically chosen Lyapunov function is monotonically decreasing; however, depending on the chosen Lyapunov function the transient behaviors can be very different. Consequently, we are instead interested in considering a class of Lyapunov functions such that each Lyapunov function produces a different event-triggered coordination algorithm to solve the multi-agent average consensus problem. The proposed class of algorithms all guarantee exponential convergence of the resulting network and exclusion of Zeno behavior. This allows us to easily consider the implementation of different algorithms that all guarantee correctness to be able to meet varying performance needs. Simulations are provided to illustrate our findings.

## I. INTRODUCTION

The distributed coordination problem of dynamic multi-agent systems has been widely studied due to their broad applications in areas such as unmanned vehicles, mobile robots, and wireless communication networks [1]–[3]. In many of these applications, groups of agents are required to agree upon certain quantities of interest, or in other words, to achieve a consensus state; typical results can be found in [4]–[9]. When considering implementation of these ideas, some of these algorithms require agents to communicate and update control signals continuously or with a fixed sampling period [4]–[6], which are inefficient and generally result in excessive consumption of on-board energy resources. To reduce the amount of communications and controller updates while maintaining the desired performance of the network, event-triggered algorithms have recently been gaining popularity [7]–[9].

The main idea behind event-triggered algorithms is to take actions only when necessary so that some desired property of the system can still be maintained efficiently. There are many recent works on distributed event-triggered control over multi-agent systems for both undirected and directed graphs [9]–[20]. Among them, [9] proposes an algorithm using a triggering function whose threshold is time-dependent with predefined constant parameters. In general, these time-dependent thresholds are easy to design to guarantee deadlocks (or Zeno behaviors, meaning an infinite number of events triggered in a finite number of time period) do not occur, but require global information to guarantee

convergence to exactly a consensus state. Instead, some event-triggered algorithms use state-dependent thresholds to determine when actions should be taken [10], [11]; however, these triggers might be risky to implement as Zeno behaviors are harder to exclude. A combination of time-dependent and state-dependent algorithms are given in [12], [13], either by introducing a time-dependent internal dynamic variable or a bounded convergent function to the state-dependent threshold to rule out Zeno behaviors. As Zeno behaviors are impossible in a given physical implementation, it is necessary and essential to exclude it in the event-triggered algorithm design to guarantee its correctness.

The event-triggered algorithms we propose in this paper are state-dependent and Lyapunov function-based. More specifically, given a Lyapunov function for a certain system, an event-triggered controller can generally be developed to maintain stability of the system while reducing sampling or communication, using the given Lyapunov function as a certificate of correctness. In other words, all events are triggered based on how we want the given Lyapunov function to evolve in time. However, it is known that a Lyapunov function is not unique for a given system, and each individual function may result in a totally different, but equally valid/correct triggering law. Consequently, there are many works that propose one such algorithm based on one function that all have the same guarantee: asymptotic convergence to a consensus state. Simulations then show that these ideas are promising when compared against periodic implementations in reducing communication while maintaining stability, but there are no formal guarantees on the gained efficiency. Moreover, this means there is no established way to compare the performance of two different event-triggered algorithms that solve the same problem. In particular, given two different event-triggered algorithms that both guarantee convergence, their trajectories and communication schedules may be wildly different before ultimately converging to the desired set of states. There are some new works that are addressing exactly this topic [21]–[23], which set the basis for this paper. More specifically, once established methods of comparing the performance of event-triggered algorithms against one another are developed, currently available algorithms will likely be revisited to optimize different types of performance metrics. In particular, we notice that different algorithms are better than others in different scenarios when considering metrics such as convergence speed or total energy consumption. Therefore, instead of trying to design only one event-triggered algorithm that simply guarantees convergence, we design an entire class of event-triggered algorithms that can be easily tuned to meet varying performance needs.

The authors are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030, USA, {pxu3@masonlive.gmu.edu}

*Statement of contributions:* The main contribution of this paper is that we propose an entire class of event-triggered coordination algorithms that all guarantee exponential stability while excluding Zeno behaviors. One such algorithm that solves the exact problem we consider here is given in [11], which is named as **Algorithm 1** for simplicity. For our work, we first design a distributed event-triggered algorithm based on an alternative Lyapunov function and name it as **Algorithm 2**. Using these two algorithms, we then parameterize an entire class of Lyapunov functions and show how each individual function can be used to develop a **Combined Algorithm**. More specifically, choosing any parameter  $\lambda \in [0, 1]$  yields an event-triggered algorithm that guarantees convergence while using a different Lyapunov function as a certificate for correctness. Changing  $\lambda$  can then help achieve varying performance goals while always guaranteeing stability. With the asymptotic convergence and exclusion of Zeno behavior for both **Algorithm 1** and **Algorithm 2**, we establish that the **Combined Algorithm** also excludes Zeno behavior and guarantees convergence of the system. Various simulations illustrate the correctness and performance of different algorithms we propose.

*Organization:* The rest of the paper is organized as follows. Section II introduces the preliminaries and Section III formulates the problem of interest. Section IV first summarizes the related work in [11] and then proposes a novel strategy based on a new Lyapunov function. The combined algorithm that based on the combined Lyapunov function is proposed in Section V. Section VI presents the simulation results, followed by the conclusions given in Section VII.

*Notations:*  $\mathbb{R}$  denotes the set of real numbers.  $\mathbf{1}_N \in \mathbb{R}^N$  denotes the column vector with each components being one and dimension  $N$ .  $\|\cdot\|$  denotes the Euclidean norm for vectors or induced 2-norm for matrices.

## II. PRELIMINARIES

Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$  denote a weighted digraph of  $N$  agents with a vertices set  $\mathcal{V} = \{1, \dots, N\}$ , directed edges  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  and a weighted adjacency matrix  $\mathcal{W} \in \mathbb{R}_{\geq 0}^{N \times N}$ . Given an edge  $(i, j) \in \mathcal{E}$ , we refer to  $j$  as an out-neighbor of  $i$  and  $i$  as an in-neighbor of  $j$ . The sets of out- and in-neighbors of a given node  $i$  are  $\mathcal{N}_i^{\text{out}}$  and  $\mathcal{N}_i^{\text{in}}$ , respectively. The elements in  $\mathcal{W}$  satisfies  $w_{ij} > 0$  if  $(i, j) \in \mathcal{E}$  and  $w_{ij} = 0$  otherwise. A path from vertex  $i$  to  $j$  is an ordered sequence of vertices such that each intermediate pair of vertices is an edge. A digraph  $\mathcal{G}$  is strongly connected if there exists a path from all  $i \in \mathcal{V}$  to all  $j \in \mathcal{V}$ . The out- and in-degree matrices  $D^{\text{out}}$  and  $D^{\text{in}}$  are diagonal matrices where

$$d_i^{\text{out}} = \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}, \quad d_i^{\text{in}} = \sum_{j \in \mathcal{N}_i^{\text{in}}} w_{ji},$$

respectively. A digraph is weight-balanced if  $D^{\text{out}} = D^{\text{in}}$  and the weighed Laplacian matrix is  $L = D^{\text{out}} - \mathcal{W}$ .

Young's inequality [24] states that given  $x, y \in \mathbb{R}$ , for any  $\varepsilon \in \mathbb{R}_{>0}$ ,

$$xy \leq \frac{x^2}{2\varepsilon} + \frac{\varepsilon y^2}{2}, \quad (1)$$

which shall be used in the theoretical analysis of this paper.

For a strongly connected and weight-balanced digraph, zero is a simple eigenvalue of  $L$ , therefore, we order its eigenvalues as  $\lambda_1 = 0 < \lambda_2 \leq \dots \lambda_N$ . The following property will also be used:

$$\lambda_2(L)x^T L^T x \leq x^T L^T L x \leq \lambda_N(L)x^T L^T x. \quad (2)$$

## III. PROBLEM STATEMENT

Consider the multi-agent average consensus problem for a network of  $N$  agents over a weight-balanced and strongly connected digraph. Let  $\mathcal{G}$  denote the communication topology of this network. Without loss of generality, we say that an agent  $i$  is able to receive information from neighbors in  $\mathcal{N}_i^{\text{out}}$  and send information to neighbors in  $\mathcal{N}_i^{\text{in}}$ . Assume that all inter-agent communications are instantaneous and of infinite precision. Let  $x_i$  denote the state of agent  $i \in \{1, 2, \dots, N\}$  and consider the single-integrator dynamics

$$\dot{x}_i(t) = u_i(t). \quad (3)$$

As is well known, the distributed continuous control law

$$u_i(t) = - \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(x_i(t) - x_j(t)) \quad (4)$$

drives the states of all agents in the system asymptotically converge to the average of the initial conditions [25].

However, implementing this protocol requires all agents to continuously access their neighbors' state information and keep updating their own control signals, which is unrealistic in practice in terms of both communication and control. Therefore, here we consider the situation where neighbors of a given agent receive information from it only when this agent decides to broadcast, and with the information received, neighbors can update their states accordingly. Let  $\hat{x}_i(t)$  denote the last broadcast state of agent  $i \in \{1, \dots, N\}$  at time  $t \in \mathbb{R}_{\geq 0}$  and assume that all agents have continuous access to their own states, then the distributed event-triggered control law (4) is modified into [7]

$$u_i(t) = - \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t)). \quad (5)$$

With the above controller (5), each agent  $i$  is equipped with a triggering function  $f_i(\cdot)$  that takes values in  $\mathbb{R}$  and depends on local information only, i.e., on the true state  $x_i(t)$  and the broadcast state  $\hat{x}_i(t)$ . An event for agent  $i$  is triggered as soon as the triggering condition

$$f_i(t, x_i(t), \hat{x}_i(t)) > 0 \quad (6)$$

is fulfilled. The triggered event drives agent  $i$  to broadcast its state so that the neighbors of agent  $i$  can update their states. Therefore our purpose of this paper is to identify event-based triggers that work efficiently under the Lyapunov function-based triggering law with state-dependent thresholds.

## IV. DISTRIBUTED TRIGGER DESIGN

### A. Related work

The distributed event-triggered coordination problem for multi-agent systems over weight-balanced digraphs has been studied in [11]. As we study the same problem and their

results are essential for us to develop our algorithms, we summarize their results first and name their algorithm as **Algorithm 1**.

The event-triggered law proposed in [11] is Lyapunov function-based, with candidate Lyapunov function be

$$V_1(x(t)) = \frac{1}{2}(x(t) - \bar{x})^T(x(t) - \bar{x}), \quad (7)$$

where  $x(t) = (x_1(t), \dots, x_N(t))^T$ ,  $\bar{x} = \frac{1}{N}(\mathbf{1}_N^T x(0))\mathbf{1}_N$  is the agreement at the average of the states of all agents.

The derivative of  $V_1(x(t))$  is upper bounded by

$$\dot{V}_1(x(t)) \leq -\frac{1}{2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} \left[ (1-a_i)(\hat{x}_i(t) - \hat{x}_j(t))^2 - \frac{e_i^2(t)}{a_i} \right], \quad (8)$$

where  $a_i$  are arbitrary positive constants and  $e_i(t) = \hat{x}_i(t) - x_i(t)$  is the measurement error between agent  $i$ 's last broadcast state and its current state at time  $t$ .

The condition to ensure that the candidate Lyapunov function  $V_1(x(t))$  is monotonically decreasing is to maintain

$$\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} \left[ (1-a_i)(\hat{x}_i(t) - \hat{x}_j(t))^2 - \frac{e_i^2(t)}{a_i} \right] \geq 0,$$

for all agents  $i \in \{1, \dots, N\}$  at all times, which can be accomplished by ensuring

$$e_i^2(t) \leq \frac{a_i(1-a_i)}{d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2. \quad (9)$$

As the trigger design is optimal when  $a_i = 0.5$  for all agents  $i \in \{1, \dots, N\}$  [11], their triggering function is defined as

$$f_i(e_i(t)) = e_i^2(t) - \frac{\sigma_i}{4d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2, \quad (10)$$

where  $\sigma_i \in (0, 1)$  is a design parameter that affects how flexible the trigger is. According to the triggering function, the event is triggered when  $f_i(e_i(t)) > 0$  or when  $f_i(e_i(t)) = 0$  and  $\phi_i = \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 \neq 0$ .

Basically, the trigger above makes sure that  $\dot{V}_1(x(t))$  is always negative as long as the system has not converged, therefore, **Algorithm 1** guarantees all agents to converge to the average of the initial states, i.e.  $\lim_{t \rightarrow \infty} x(t) = \bar{x} = \frac{1}{N}(\mathbf{1}_N^T x(0))\mathbf{1}_N$ , interested readers can refer to [11, Theorem 5.3] for more details.

### B. Proposed new algorithm

As we know, the Lyapunov function is not unique for the stability studying of the same system and each individual function may result a totally different triggering law. Therefore, we propose a novel triggering strategy based on an alternative Lyapunov function

$$V_2(x(t)) = \frac{1}{2}x(t)^T L^T x(t), \quad (11)$$

and name our algorithm as **Algorithm 2**.

**Proposition 4.1:** For  $i \in \{1, \dots, N\}$ , let  $b_i, c_i, c_j > 0$  for all  $i, j \in \{1, \dots, N\}$  and  $e_i(t) = \hat{x}_i(t) - x_i(t)$ , then the

derivative of  $V_2(x(t))$  is upper bounded by

$$\dot{V}_2(x(t)) \leq -\sum_{i=1}^N \left( \delta_i u_i^2(t) - \left( \frac{d_i^{\text{out}}}{2b_i} + \frac{d_i^{\text{out}}}{2c_i} \right) e_i^2(t) \right), \quad (12)$$

where

$$\delta_i \triangleq 1 - \frac{d_i^{\text{out}} b_i}{2} - \sum_{j \in \mathcal{N}_i^{\text{out}}} \frac{w_{ij} c_j}{2}, \quad (13)$$

and  $u_i(t)$  is what defined in (5).

The proof to Proposition 4.1 is omitted due to space limit. Note that the coefficient of  $e_i^2$  is always positive. To ensure the coefficient of  $u_i^2$  is also positive, we require  $b_i, c_j < \frac{1}{d_i^{\text{out}}}$ .

From Proposition 4.1, a sufficient condition to guarantee that the proposed candidate Lyapunov function  $V_2(x(t))$  is monotonically decreasing is to ensure that

$$\delta_i \left( \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 \right) - \left( \frac{d_i^{\text{out}}}{2b_i} + \frac{d_i^{\text{out}}}{2c_i} \right) e_i^2(t) \geq 0$$

for all agents  $i \in \{1, \dots, N\}$  at all times, meaning that

$$e_i^2(t) \leq \frac{2\delta_i b_i c_i}{(b_i + c_i) d_i^{\text{out}}} \left( \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 \right). \quad (14)$$

The triggering function for **Algorithm 2** is therefore defined as

$$f_i(e_i(t)) = e_i^2(t) - \frac{2\sigma_i \delta_i b_i c_i}{(b_i + c_i) d_i^{\text{out}}} \left( \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 \right), \quad (15)$$

where  $\sigma_i$  as before is a design parameter that affects how flexible the trigger is and controls the trade-off between communication and performance. Setting  $\sigma_i$  close to 0 is generally greedy, meaning that the trigger is enabled more frequently and the network requires more communications, which makes agent  $i$  contribute more to the decrease of the Lyapunov function  $V_2(x(t))$  and therefore the network converges faster while setting the value of  $\sigma_i$  close to 1 achieves the opposite results.

**Corollary 4.2:** For each agent  $i \in \{1, \dots, N\}$  with the triggering function defined in (15), if each agent  $i$  enforces the condition  $f_i(e_i) \leq 0$  at all times, then

$$\dot{V}_2(x(t)) \leq -\sum_{i=1}^N (1 - \sigma_i) \delta_i \left( \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 \right).$$

Similar as [11], to avoid the possibility that agent  $i$  may miss any triggers, we define an event either by

$$f_i(e_i(t)) > 0 \quad \text{or} \quad (16)$$

$$f_i(e_i(t)) = 0 \quad \text{and} \quad \phi_i \neq 0 \quad (17)$$

where  $\phi_i = (\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2)$ .

We also prescribe the following additional trigger as in [11] to address the non-Zeno behavior. Let  $t_{last}^i$  be the last time at which agent  $i$  broadcasts its information to its neighbors. If at some time  $t \geq t_{last}^i$ , agent  $i$  receives information from a neighbor  $j \in \mathcal{N}_i^{\text{out}}$ , then agent  $i$  immediately

<p><b>Algorithm 1</b></p> $f_i(e_i) \triangleq e_i^2 - \frac{\sigma_i}{4d_i^{out}} \sum_{j \in \mathcal{N}_i^{out}} w_{ij} (\hat{x}_i - \hat{x}_j)^2,$ $\varepsilon_i < \sqrt{\frac{\sigma_i}{4d_i^{out} w_i^{\max}  \mathcal{N}_i^{out} }}$
<p><b>Algorithm 2</b></p> $f_i(e_i) \triangleq e_i^2 - \frac{2\sigma_i \delta_i b_i c_i}{(b_i + c_i) d_i^{out}} \left( \sum_{j \in \mathcal{N}_i^{out}} w_{ij} (\hat{x}_i - \hat{x}_j) \right)^2$ $\varepsilon_i < \sqrt{\frac{2\sigma_i \delta_i b_i c_i}{(b_i + c_i) d_i^{out}}}$

TABLE I  
DIFFERENCE BETWEEN **ALGORITHM 1** AND **ALGORITHM 2**

broadcasts its state if

$$t \in (t_{last}^i, t_{last}^i + \varepsilon_i), \quad (18)$$

where

$$\varepsilon_i < \sqrt{\frac{2\sigma_i \delta_i b_i c_i}{(b_i + c_i) d_i^{out}}} \quad (19)$$

is a design parameter selected to ensure the exclusion of Zeno. The reasoning is similar as that in [11].

We summarize the differences between **Algorithm 1** proposed in [11] and **Algorithm 2** proposed here in Table I. Once the triggering function and parameters  $\varepsilon_i$  are chosen for each agent, either algorithm can be implemented using the coordination algorithm provided in Table II.

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

- 1: **if**  $f_i(e_i(t)) > 0$  or  $(f_i(e_i(t)) = 0$  and  $\phi_i \neq 0)$  **then**
- 2: broadcast state information  $x_i(t)$  and update control signal  $u_i(t)$
- 3: **end if**
- 4: **if** new information  $x_j(t)$  is received from some neighbor(s)  $j \in \mathcal{N}_i^{out}$  **then**
- 5: **if** agent  $i$  has broadcast its state at any time  $t' \in [t - \varepsilon_i, t)$  **then**
- 6: broadcast state information  $x_i(t)$
- 7: **end if**
- 8: update control signal  $u_i(t)$
- 9: **end if**

TABLE II  
EVENT-TRIGGERED COORDINATION ALGORITHM.

For **Algorithm 2**, we have the following proposition and theorem specify its non-zeno behavior and convergence. Proof is omitted due to space limit.

**Proposition 4.3:** (No Zeno Behavior) Consider the system (3) executing control law (5) with the triggering function given by (15). For the weight-balanced, strongly connected digraph with any initial conditions, when executing Table II, the system will not exhibit Zeno behavior.

**Theorem 4.4:** (Asymptotic Convergence to Average Consensus). Given the system (3) executing Table II over a weight-balanced, strongly connected digraph, all agents

asymptotically converge to the average of the initial states, i.e.  $\lim_{t \rightarrow \infty} x(t) = \bar{x}$ , where  $\bar{x} = \frac{1}{N}(\mathbf{1}_N^T x(0))\mathbf{1}_N$ .

## V. A CLASS OF EVENT-TRIGGERED ALGORITHMS

As stated in Section I, given a system and a Lyapunov function, there are many works studying event-triggered control to reach the goal of maintaining the stability of the system while increasing the efficiency of the system. However, there is very little work currently available that mathematically quantifies these benefits. Recently, some works began establishing results along this line [21]–[23], however, this area is still in its infancy. In particular, there are not yet established ways to compare the performance of an event-triggered algorithm with another. Consequently, many different algorithms can be proposed to ultimately solve the same problem, while each algorithm is slightly different and can produce different trajectories. Specifically in our case, **Algorithm 1** and **Algorithm 2** solve the same problem, but what we can say about the two algorithms is only that they both exclude Zeno behavior and ensure asymptotical convergence of the network. However, we have found that depending on the initial condition and network topology, each algorithm may out-perform the other in terms of different evaluation metrics. In any case, once these performance metrics become better researched, there will likely be more standard ways to mathematically compare the two different algorithms. Therefore, for now, instead of designing only one event-triggered algorithm for the system that only works better in one situation, we aim to design an entire class of algorithms that can easily be tuned to meet varying performance needs.

We do this by parameterizing a set of Lyapunov functions rather than studying only a specific one. To the best of our knowledge, this paper is then a first study of how to design an entire class of algorithms that use different Lyapunov functions to guarantee correctness, with the intention of being able to use the best one at all times.

More specifically, given any  $\lambda \in [0, 1]$ , we define a combined candidate Lyapunov function as

$$V_\lambda(x(t)) = \lambda V_1(x(t)) + (1 - \lambda) V_2(x(t)). \quad (20)$$

Accordingly, the derivative of  $V_\lambda(x(t))$  takes the form

$$\dot{V}_\lambda(x(t)) = \lambda \dot{V}_1(x(t)) + (1 - \lambda) \dot{V}_2(x(t)). \quad (21)$$

Following the steps of deriving the triggering functions in Section IV, the triggering function developed based on (20) is given by

$$f_i(e_i(t)) = e_i^2(t) - \sigma_i \left[ \frac{\lambda}{4d_i^{out}} \sum_{j \in \mathcal{N}_i^{out}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t))^2 + \frac{(1 - \lambda) 2\delta_i b_i c_i}{(b_i + c_i) d_i^{out}} \left( \sum_{j \in \mathcal{N}_i^{out}} w_{ij} (\hat{x}_i(t) - \hat{x}_j(t)) \right)^2 \right] \quad (22)$$

We refer to this as the **Combined Algorithm** that is parameterized by  $\lambda \in [0, 1]$ . Similarly, we set a parameter  $\varepsilon_i$

such that

$$\varepsilon_i < \sqrt{\frac{\lambda \sigma_i}{4d_i^{\text{out}} w_i^{\text{max}} |\mathcal{N}_i^{\text{out}}|} + \frac{2(1-\lambda)\sigma_i \delta_i b_i c_i}{(b_i + c_i) d_i^{\text{out}}}.$$

Then, with this triggering function (22) and  $\varepsilon_i$ , the **Combined Algorithm** can also be implemented using Table I. Note that  $\lambda = 0$  in the **Combined Algorithm** recovers **Algorithm 2** and  $\lambda = 1$  recovers **Algorithm 1**.

*Corollary 5.1:* **Algorithm 1** and **Algorithm 2** both ensure all agents to asymptotically converge to the average of their initial states through proving that their Lyapunov functions converge asymptotically. Therefore, as a linear combination of  $V_1(x(t))$  and  $V_2(x(t))$ ,  $V_\lambda(x(t))$  also converges exponentially, which means that a network executing the **Combined Algorithm** shall converge asymptotically to the average of its initial state.

## VI. SIMULATIONS

We demonstrate the performance of the proposed algorithms through several simulations. In particular, we show how either **Algorithm 1** or **Algorithm 2** could be argued to be ‘better’ given different initial conditions and network topologies, which has set the basis for our introduction of the **Combined Algorithm** to easily go between the two.

In all simulations we consider a system of  $N = 5$  agents with dynamics (3) and control law (5). The triggering functions for **Algorithm 1**, **Algorithm 2** and **Combined Algorithm** are defined in (10), (15) and (22), respectively. Throughout the simulations, we set  $b_i = c_i = 0.5$  for all agents  $i \in \{1, \dots, N\}$ . We implement the  $\lambda = 0.5$  version of the **Combined Algorithm**.

We adopt two different networks with different initial conditions for comparison. The initial state of **Network 1** is  $x_1(0) = [1, 1, 0, 2, 0]^T$  and its weighted adjacency matrix is

$$\mathcal{W}_1 = \begin{bmatrix} 1/4 & 1/4 & 0 & 1/3 & 1/6 \\ 0 & 0 & 1/2 & 1/6 & 1/3 \\ 1/2 & 1/3 & 1/6 & 0 & 0 \\ 1/4 & 1/6 & 0 & 1/3 & 1/4 \\ 0 & 1/4 & 1/3 & 1/6 & 1/4 \end{bmatrix}.$$

The initial state of **Network 2** is  $x_2(0) = [0, 1, 1, 1, 1]^T$ , with an weighted adjacency matrix  $\mathcal{W}_2$  whose diagonal elements are 0 and the rest of the entries are  $1/4$ .

Figure 1 shows the evolutions of the three Lyapunov functions for the two networks with  $\sigma_i = 0.9$  for all agents, which corroborates our analysis that the proposed **Algorithm 2** and **Combined Algorithm** ensure convergence for the resulting systems. In addition, Figure 1(a) shows that **Network 1** converges fastest when executing **Algorithm 2** while Figure 1(b) shows that **Network 2** converges fastest when executing **Algorithm 1**. A more direct comparison is given in Figure 2, on which we have  $T_{\text{con}}$  denote the time needed for the two networks to reach a 99% convergence of the Lyapunov function when executing all algorithms with respect to varying  $\sigma$ . It is clear that for **Algorithm 1** and **Algorithm 2**, there exist situations when one outperforms the other in terms of convergence time.

In addition to convergence time, other important metrics may include power consumption and total energy expenditure. To compare these we use a simulation step size of  $h = 0.001$  second and we adopt the following power calculation model in units of dBmW [26] :

$$\mathcal{P} = 10 \log_{10} \sum_{i=1}^N \left[ \sum_{j \in \{1, \dots, N\}, j \neq i} \beta 10^{0.1 P_{i \rightarrow j} + \alpha \|x_i - x_j\|} \right],$$

where  $\alpha > 0$  and  $\beta > 0$  depend on the characteristics of the wireless medium and  $P_{i \rightarrow j}$  is the power of the signal transmitted from agent  $i$  to agent  $j$  in units of dBmW. Similar as [27], we set  $\alpha$ ,  $\beta$  and  $P_{i \rightarrow j}$  to be 1. The total energy needed can be calculated by multiplying the power in units of milliwatt (mW) with the number of steps for convergence, which is  $T_{\text{con}}/h$ , which in decibels (dB) is

$$E = \mathcal{P} + 10 \log_{10} \frac{T_{\text{cov}}}{h}.$$

Figures 3(a) and 3(b) compare the average power consumption for each algorithm and Figures 4(a) and 4(b) show the total communication energy required to reach a 99% consensus state. These figures show that in **Network 1**, **Algorithm 2** can always reach consensus using less total communication energy for varying  $\sigma_i$ . On the other hand, in **Network 2**, **Algorithm 1** can complete the same task using less total communication energy. Therefore, depending on different network topologies and initial conditions and depending on what performance metrics are most important for the application at hand, it may be desirable to implement different types of event-triggered algorithms. Note that the **Combined Algorithm** can easily be tuned to approach either **Algorithm 1** or **Algorithm 2** or anything in between to meet varying system needs by setting values for  $\lambda$ . This also motivates our future work of adapting  $\lambda$  online to further improve performance.

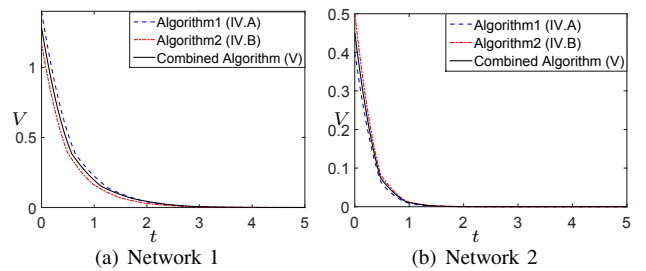


Fig. 1. The evolution of the Lyapunov functions.

## VII. CONCLUSION

This paper first proposes a novel distributed event-triggered communication and control law based on a new Lyapunov function that achieves consensus and excludes the possibility of Zeno behavior for multi-agent systems on weight-balanced digraphs. We then show how the algorithm design can easily be extended by considering a class of Lyapunov functions parameterized by  $\lambda \in [0, 1]$  such that each  $\lambda$  defines a new Lyapunov function coupled with a new event-triggered coordination algorithm which uses that particular

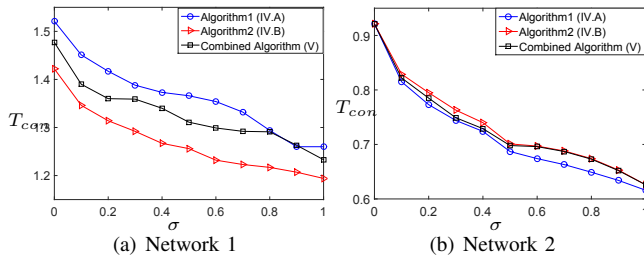


Fig. 2. Time needed to reach 99% convergence.

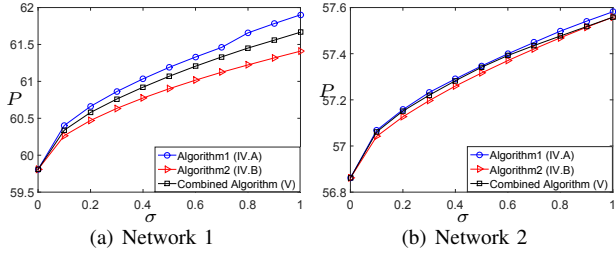


Fig. 3. Average communication power consumption.

function to guarantee correctness. Although any  $\lambda \in [0, 1]$  produces an algorithm that guarantees Zeno-free asymptotic convergence to the desired state, the trajectories (or performance) can be very different. Consequently, this gives us an easy way to consider many event-triggered algorithms that all have the minimum requirement of guaranteed asymptotic stability. Future work will be devoted to studying how to adapt  $\lambda$  online to take full advantage of this class of algorithms to meet varying performance needs.

## REFERENCES

- [1] J. N. Callen, "Distributed control for unmanned vehicles," *IEEE Concurrency*, vol. 6, no. 2, pp. 16–20, 1998.
- [2] R. Vidal, O. Shakhmurov, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *Robotics and Automation, 2003*, vol. 1. IEEE, 2003, pp. 584–589.
- [3] J. Liang, Z. Wang, B. Shen, and X. Liu, "Distributed state estimation in sensor networks with randomly occurring nonlinearities subject to time delays," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 1, p. 4, 2012.
- [4] B. Liu, W. Lu, and T. Chen, "Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes," *SIAM Journal on Control and Optimization*, vol. 49, no. 1, pp. 227–253, 2011.
- [5] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.
- [6] K. You and L. Xie, "Network topology and communication data rate for consensusability of discrete-time multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2262–2275, 2011.
- [7] 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.
- [8] E. García, 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.
- [9] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.

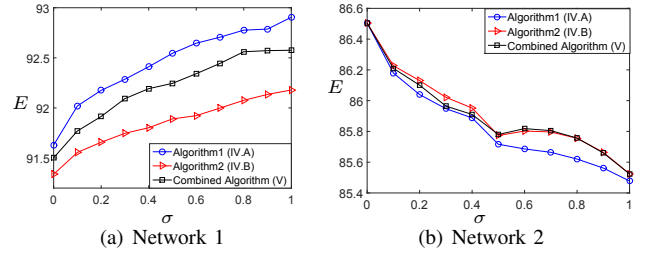


Fig. 4. Total energy expenditure.

- [10] C. Nowzari and J. Cortés, "Zeno-free, distributed event-triggered communication and control for multi-agent average consensus," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 2148–2153.
- [11] —, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.
- [12] A. Girard, "Dynamic triggering mechanisms for event-triggered control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1992–1997, 2015.
- [13] Z. Sun, N. Huang, B. D. Anderson, and Z. Duan, "A new distributed zeno-free event-triggered algorithm for multi-agent consensus," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 3444–3449.
- [14] X. Yi, K. Liu, D. V. Dimarogonas, and K. H. Johansson, "Distributed dynamic event-triggered control for multi-agent systems," 2017. [Online]. Available: arXiv:1704.05434
- [15] V. Dolk and W. Heemels, "Dynamic event-triggered control under packet losses: The case with acknowledgements," in *Event-based Control, Communication, and Signal Processing (ECCSP), 2015 International Conference on*. IEEE, 2015, pp. 1–7.
- [16] X. Meng, L. Xie, Y. C. Soh, C. Nowzari, and G. J. Pappas, "Periodic event-triggered average consensus over directed graphs," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 4151–4156.
- [17] X. Yi, W. Lu, and T. Chen, "Distributed event-triggered consensus for multi-agent systems with directed topologies," in *Control and Decision Conference (CCDC), 2016 Chinese*. IEEE, 2016, pp. 807–813.
- [18] W. Hu, L. Liu, and G. Feng, "Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy," *IEEE transactions on cybernetics*, 2017.
- [19] D. Liuzza, D. V. Dimarogonas, M. di Bernardo, and K. H. Johansson, "Distributed model based event-triggered control for synchronization of multi-agent systems," *Automatica*, vol. 73, pp. 1–7, 2016.
- [20] X.-M. Zhang, Q.-L. Han, and B.-L. Zhang, "An overview and deep investigation on sampled-data-based event-triggered control and filtering for networked systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 4–16, 2017.
- [21] D. Antunes and W. Heemels, "Rollout event-triggered control: Beyond periodic control performance," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3296–3311, 2014.
- [22] C. Ramesh, H. Sandberg, and K. H. Johansson, "Performance analysis of a network of event-based systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3568–3573, 2016.
- [23] B. A. Khashoei, D. Antunes, and W. Heemels, "Output-based event-triggered control with performance guarantees," *IEEE Transactions on Automatic Control*, 2017.
- [24] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge university press, 1952.
- [25] 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.
- [26] N. C. Martins *et al.*, "Jointly optimal placement and power allocation of wireless networks," Ph.D. dissertation, 2008.
- [27] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.