0.1 Introduction

In this chapter we look at one of the canonical driving examples for multi-agent systems: average consensus. In this scenario, a group of agents seek to agree on the average of their initial states. Depending on the particular application, such states might correspond to sensor measurements, estimates about the position of a target, or some other data that needs to be fused. Due to its numerous applications in networked systems, many algorithmic solutions exist to the multi-agent average consensus problem; however, a majority of them rely on agents having continuous or periodic availability of information from other agents. Unfortunately, this assumption leads to inefficient implementations in terms of energy consumption, communication bandwidth, network congestion, and processor usage. Motivated by these observations, our main goal here is the design of provably correct distributed event-triggered strategies that autonomously decide when communication and control updates should occur so that the resulting asynchronous network executions still achieve average consensus.

The literature and motivation behind multi-agent average consensus is extensive, see e.g., [1, 2, 3, 4] and references therein. This chapter aims to provide a conceptual introduction to event-triggered control strategies applied to consensus problems. Triggered controllers seek to understand the trade-offs between computation, communication, sensing, and actuator effort in achieving a desired task with a guaranteed level of performance. Early works [5] only consider tuning controller executions to the state evolution of a given system, but these ideas have since been extended to consider other tasks such as when to take the sample of a state or when to broadcast information over a wireless network; see [6] and references therein for a recent overview. Among the many references in the context of multi-agent systems, [7] specifies the responsibility of each agent in updating the control signals, [8] considers network scenarios with disturbances, communication delays, and packet drops, and [9] studies decentralized event-based control that incorporates estimators of the interconnection signals among agents. These works are all concerned with designing event-triggers that ultimately determine when control signals should be updated in addition to how. Several works have explored the application of event-triggered ideas to the acquisition of information by the agents rather than only for actuation. To this end, [10, 11, 12] combine event-triggered controller updates with sampled data that allows for the periodic evaluation of the triggers. Instead, some works [13] drop the need for periodic access to information by considering event-based broadcasts, where agents decide with local information only when to obtain further information about neighbors. Self-triggered control [14, 15] 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. Team-triggered coordination [16] combines the strengths of event- and self-triggered control into a unified approach for networked systems.

	Triggered comm?	Trigger dependence	Memory structure	Graph structure	Trigger evaluation	Provably no Zeno?
[17]	no	state	centralized	undirected	continuous	yes
[17]	no	state	decentralized	undirected	continuous	no
[18]	no	state	centralized	directed	continuous	yes
[18]	no	state	decentralized	directed	continuous	no
[19]	yes	time	decentralized	undirected	continuous	no
[19]	yes	time	requires λ_2	undirected	continuous	yes
[20]	yes	state	requires N	undirected	continuous	no
[21]	yes	state	requires K	directed	periodic	yes
[12]	yes	state	decentralized	undirected	periodic	yes
[22, 23]	yes	state	decentralized	directed	periodic	yes

Table 1 Event-triggered multi-agent average consensus

Organization

Table 1 shows the progression of event-triggered consensus problems that are covered in this chapter. It should be noted that this is a very narrow scope on the field of event-triggered consensus problems intended to introduce the high-level ideas behind event-triggered communication and control laws and provide insight into how they are designed. In particular, this chapter only discusses works that consider single-integrator dynamics and no uncertainties (e.g., disturbances, noise, quantization, wireless communication issues). Given that this is currently an active area of research, it goes without saying that there are many important related works that are not highlighted here. Examples include scenarios with disturbances, sensor noise, delayed communication, quantized communication, packet drops, more general dynamics, dynamic topologies, and heterogeneous agents; to name a few. Lastly, it should also be noted that although the table references journal articles that first present these ideas going back to 2012, preliminary results from these works has been presented at various conferences as early as 2008. The contents of the chapter are summarized next.

The first application of event-triggered control to the multi-agent consensus problem was in [17], where the authors propose a Lyapunov-based event-triggered control strategy that dictates when agents should update their control signals. Unfortunately, its implementation relies on each agent having perfect information about their neighbors at all times. Identifying this limitation, the authors in [19] propose an event-triggered communication and control law that not only determines when agents should update their control signals, but also when they should broadcast state information to their neighbors. However, the drawback of the proposed algorithm is that it is a time-dependent triggering rule with design parameters that are difficult to choose to yield good performance. Instead, a state-dependent triggering rule is proposed in [20] which better aligns the events with the desired task; this is explained in more detail later. Lastly, all the above algorithms assume continuous evaluation of some function is possible to determine exactly when some event has occurred. Even in scenarios where Zeno behavior (an infinite number of events occurring in a finite period of time) can be provably avoided, the time between events may still be arbitrarily small which can be problematic for digital implementations. Consequently, the works [21, 12, 22, 23] propose algorithms that only require triggering functions to be evaluated periodically rather than continuously. Finally, we close the chapter by identifying some shortcomings of the current state of the art and ideas for future work.

0.2 Preliminaries

This section introduces some notational conventions and notions on graph theory. Let $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{Z}_{>0}$ denote the set of real, positive real, nonnegative real, and positive integer numbers, respectively. We denote by $\mathbf{1}_N$ and $\mathbf{0}_N \in \mathbb{R}^N$ the column vectors with entries all equal to one and zero, respectively. We let $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}^N$. We let $\operatorname{diag}(\mathbb{R}^N) = \{x \in \mathbb{R}^N \mid x_1 = \cdots = x_N\} \subset \mathbb{R}^N$ be the agreement subspace in $\mathbb{R}^N$. For a finite set S, we let $|S|$ denote its cardinality. Given $x, y \in \mathbb{R}$, Young's inequality states that, for any $\epsilon \in \mathbb{R}_{>0}$,

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

A weighted directed graph (or weighted digraph) $\mathcal{G} = (V, E, W)$ is comprised of a set of vertices $V = \{1, \ldots, N\}$, directed edges $E \subset V \times V$ and weighted adjacency matrix $W \in \mathbb{R}^{N \times N}_{\geq 0}$. Given an edge $(i, j) \in E$, 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 weighted adjacency matrix $W \in \mathbb{R}^{N \times N}$ satisfies $w_{ij} > 0$ if $(i, j) \in E$ and $w_{ij} = 0$ otherwise. The graph $\mathcal{G}$ is *undirected* if and only if $w_{ij} = w_{ji}$ for all $i, j \in V$. 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 $i \in V$ to all $j \in V$. The out- and in-degree matrices D^{out} and D^{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 $D^{\text{out}} = D^{\text{in}}$. The (weighted) Laplacian matrix is $L = D^{\text{out}} - W$. Based on the structure of L, at least one of its eigenvalues is zero and the rest of them have nonnegative real parts. If the digraph $\mathcal{G}$ is strongly connected, 0 is a simple eigenvalue with associated eigenvector $\mathbf{1}_N$. The digraph $\mathcal{G}$ is weight-balanced if and only if $\mathbf{1}_N^T L = \mathbf{0}_N$ if and only if $L_s = \frac{1}{2}(L + L^T)$ is positive semidefinite. For a strongly connected and weight-balanced digraph, zero is a simple eigenvalue of L_s. In this case, we order its eigenvalues as $\lambda_1 = 0 < \lambda_2 \leq \cdots \leq \lambda_N$, and note the inequality

$$x^T L x \geq \lambda_2(L_s) \|x - \frac{1}{N}(\mathbf{1}_N^T x)\mathbf{1}_N\|^2, \tag{2}$$

for all $x \in \mathbb{R}^N$. The following property will also be of use later,

$$\lambda_2(L_s) x^T L x \leq x^T L_s^2 x \leq \lambda_N(L_s) x^T L x. \tag{3}$$

This can be seen by noting that L_s is diagonalizable and rewriting $L_s = S^{-1}DS$, where D is a diagonal matrix containing the eigenvalues of L_s.

0.2.1 Event-triggered control of linear systems

Here we provide a very basic working introduction to the general idea of event-triggered control by working through a simple linear control problem. The exposition closely follows [24]. The remainder of this chapter then focuses on how these elementary ideas are extended to be applied to much more in the context of multi-agent consensus on networks. We refer the interested reader to [6] for further details on the subject of event-triggered control in general.

Consider a linear control system

$$\dot{x} = Ax + Bu, \tag{4}$$

with $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Our starting point is the availability of a linear feedback controller $u^* = Kx$ such that the closed-loop system

$$\dot{x} = (A + BK)x,$$

is asymptotically stable. Given a positive definite matrix $Q \in \mathbb{R}^{n \times n}$, let $P \in \mathbb{R}^{n \times n}$ be the unique solution to the Lyapunov equation $(A + BK)^T P + P(A + BK) = -Q$. Then, the evolution of the Lyapunov function $V_c(x) = x^T P x$ along the trajectories of the closed-loop system is

$$\dot{V}_c = x^T((A + BK)^T P + P(A + BK))x = -x^T Q x.$$

Consider now a sample-and-hold implementation of the controller, where the input is not updated continuously, but instead at a sequence of to-be-determined times $\{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}} \subset \mathbb{R}_{\geq 0}$,

$$u(t) = Kx(t_\ell), \quad t \in [t_\ell, t_{\ell+1}). \tag{5}$$

Such an implementation makes sense in practical scenarios given the inherent nature of digital systems. With this controller implementation, the closed-loop system can be written as

$$\dot{x} = (A + BK)x + BKe,$$

where $e(t) = x(t_\ell) - x(t)$, $t \in [t_\ell, t_{\ell+1})$, is the state error. Then, the objective is to determine the sequence of times $\{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}}$ to guarantee some desired level of performance for the resulting system. To make this concrete, define the function

$$V(t, x_0) = x(t)^T P x(t),$$

for a given initial condition $x(0) = x_0$ (here, $t \mapsto x(t)$ denotes the evolution of the closed-loop system using (5)). We define the performance of the system via a function $S : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ that upper bounds the evolution of V. Then, the sequence of times $\{t_\ell\}$ can be implicitly defined as the times at which

$$V(t, x_0) \leq S(t, x_0) \tag{6}$$

is not satisfied. More specifically, this is an *event-triggered* condition that updates the actuator signal whenever $V(t_\ell, x_0) = S(t_\ell, x_0)$. Assuming solutions are well defined, it is not difficult to see that if the performance function satisfies $S(t, x_0) \leq \beta(t, |x_0|)$, for some $\beta \in \mathcal{KL}$, then the closed-loop system is globally uniformly asymptotically stable. Moreover, if β is an exponential function, the system is globally uniformly exponentially stable.

Therefore, one only needs to guarantee the lack of Zeno behavior. We do this by choosing the performance function S so that the inter-event times $t_{\ell+1} - t_\ell$ are lower bounded by some constant positive quantity. This can be done in a number of ways. For the linear system (4), it turns out that it is sufficient to select S satisfying $\dot{V}(t_\ell) < \dot{S}(t_\ell)$ at the event times t_ℓ (this fact is formally stated below in Theorem 0.2.1). To do so, choose $R \in \mathbb{R}^{n \times n}$ positive definite such that $Q - R$ is also positive definite. Then, there exists a Hurwitz matrix $A_s \in \mathbb{R}^{n \times n}$ such that the Lyapunov equation

$$A_s^T P + P A_s = -R$$

holds. Consider the hybrid system,

$$\begin{aligned} \dot{x}_s &= A_s x_s, \quad t \in [t_\ell, t_{\ell+1}), \\ x_s(t_\ell) &= x(t_\ell), \end{aligned}$$

whose trajectories we denote by $t \mapsto x_s(t)$, and define the performance function S by

$$S(t) = x_s^T(t) P x_s(t).$$

Letting $y = [x^T, e^T]^T \in \mathbb{R}^n \times \mathbb{R}^n$, we write the continuous-time dynamics as

$$\dot{y} = F y, \quad t \in [t_\ell, t_{\ell+1}),$$

where

$$F = \begin{bmatrix} A + BK & BK \\ -A - BK & -BK \end{bmatrix}.$$

With a slight abuse of notation, we let $y_\ell = [x^T(t_\ell), 0^T]^T$ be the state y at time t_ℓ. Note that $e(t_\ell) = 0$, for all $\ell \in \mathbb{Z}_{\geq 0}$, by definition of the update times. With this notation, we can rewrite

$$S(t) = (Ce^{F_s(t-t_\ell)}y_\ell)^T P(Ce^{F_s(t-t_\ell)}y_\ell),$$
$$V(t) = (Ce^{F(t-t_\ell)}y_\ell)^T P(Ce^{F(t-t_\ell)}y_\ell),$$

where

$$F_s = \begin{bmatrix} A_s & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} I & 0 \end{bmatrix}.$$

The condition (6) can then be rewritten as

$$f(t, y_\ell) = y_\ell^T(e^{F^T(t-t_\ell)}C^T PCe^{F(t-t_\ell)} - e^{F_s^T(t-t_\ell)}C^T PCe^{F_s(t-t_\ell)})y_\ell \leq 0.$$

Note that because we consider a deterministic system here, with the information available at time t_ℓ, it is possible to determine the next time $t_{\ell+1}$ at which (6) is violated by computing $t_{\ell+1} = h(x(t_\ell))$ as the time for which

$$f(h(x(t_\ell)), y_\ell) = 0. \tag{7}$$

The following result from [24] provides a uniform lower bound t_{min} on the inter-event times $\{t_{\ell+1} - t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}}$.

Theorem 0.2.1 (Lower bound on inter-event times for event-triggered approach) *Given the system* (4) *with controller* (5) *and controller updates given by the event-triggered policy* (7)*, the inter-event times are lower bounded by*

$$t_{\text{min}} = \min\{t \in \mathbb{R}_{>0} \mid \det(M(t)) = 0\} > 0,$$

where

$$M(t) = \begin{bmatrix} I & 0 \end{bmatrix} \left(e^{Ft}C^T PCe^{Ft} - e^{F_s t}C^T PCe^{F_s t}\right) \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

Note that the above result can also be interpreted in the context of a periodic controller implementation: any period less than or equal to t_{min} results in a closed-loop system with asymptotic stability guarantees.

0.3 Problem statement

We let $\mathcal{G}$ denote the connected, undirected communication graph that describes the communication topology in a network of N agents. In other words, agent j can communicate with agent i if j is a neighbor of i in $\mathcal{G}$. We denote by $x_i \in \mathbb{R}$ the state of agent $i \in \{1, \ldots, N\}$ and consider single-integrator dynamics

$$\dot{x}_i(t) = u_i(t). \tag{8}$$

Then, the distributed controller

$$u_i^*(x) = -\sum_{j \in \mathcal{N}_i} (x_i - x_j) \tag{9}$$

is known to drive the states of all agents to the average of the initial conditions [25, 1]. This is formalized in Theorem 0.3.1.

Theorem 0.3.1 (Continuous controller) *Given the dynamics* (8)*, if all agents implement the control law* (9)*, then multi-agent average consensus is achieved; i.e.,*

$$\lim_{t \to \infty} x_i(t) = \frac{1}{N} \sum_{j=1}^{N} x_j(0) \tag{10}$$

for all $i \in \{1, \ldots, N\}$.

Unfortunately, implementing (9) in a digital setting is not possible since it requires all agents to have continuous access to the state of their neighbors and the control inputs $u_i(t)$ must also be updated continuously. This is especially troublesome in the context of wireless network systems since this means agents must communicate with each other continuously as well. Instead, this chapter is interested in event-triggered communication and control strategies to relax these requirements.

0.4 Centralized event-triggered control

Consider the dynamics (8) and the ideal control law (9). Letting $x = (x_1, \ldots, x_N)^T$ and $u = (u_1, \ldots, u_N)^T$, the closed-loop dynamics of the ideal system is given by

$$\dot{x}(t) = -Lx(t). \tag{11}$$

As stated before, implementing this requires all agents to continuously update their control signals which is not realistic for digital controllers. Instead, following the basic idea for event-triggered control presented in Section 0.2.1, let us consider a digital implementation of this ideal controller

$$u(t) = -Lx(t_\ell), \quad t \in [t_\ell, t_{\ell+1}), \tag{12}$$

where the event times $\{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}}$ are to be determined such that the system still converges to the desired state. Let $e(t) = x(t_\ell) - x(t)$ for $t \in [t_\ell, t_{\ell+1})$ be the state measurement error. For simplicity, we denote by $\hat{x}(t) = \hat{x}(t_\ell)$ for $t \in [t_\ell, t_{\ell+1})$ as the state that was used in the last computation of the control signal. The closed-loop dynamics of the controller (12) is then given by

$$\dot{x}(t) = -L\hat{x}(t) = -L(x(t) + e(t)). \tag{13}$$

The problem can now be formalized as follows.

Problem 0.4.1 (Centralized event-triggered control) *Given the closed-loop dynamics* (13)*, find an event-trigger such that the sequence of times* $\{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}}$ *ensures multi-agent average consensus* (10) *is achieved.*

Following [17], to solve this problem we consider the Lyapunov function

$$V(x) = x^T L x.$$

Given the closed-loop dynamics (13), we have

$$\dot{V} = x^T L \dot{x} = -x^T LL(x + e) = -\underbrace{\|Lx\|^2}_{\text{"good"}} - \underbrace{x^T LLe}_{\text{"bad"}}.$$

For simplicity, we are not interested in characterizing any specific performance as in Section 0.2.1. Instead, we are only interested in asymptotic stability. The main idea of event-triggered control is then to determine when the controller should be updated (i.e., when e should be set to 0) by balancing the "good" term against the "bad" term. More specifically, we are interested in finding conditions on the error e such that $\dot{V} < 0$ at all times. Using norms, we can bound

$$\dot{V} \leq -\|Lx\|^2 + \|Lx\|\|L\|\|e\|.$$

Then, if we enforce the error e to satisfy

$$\|e\| \leq \sigma \frac{\|Lx\|}{\|L\|},$$

with $\sigma \in (0, 1)$ for all times, we have

$$\dot{V} \leq (\sigma - 1)\|Lx\|^2,$$

which is strictly negative for all $Lx \neq 0$. The following centralized event-trigger ensures this is satisfied at all times.

Theorem 0.4.2 (Centralized event-triggered control) *Given the closed-loop dynamics* (13)*, if the update times are determined as the times when*

$$f(x, e) \triangleq \|e\| - \sigma \frac{\|Lx\|}{\|L\|} = 0, \tag{14}$$

then the system achieves multi-agent average consensus.

```
At times $t \in [t_\ell, t_{\ell+1})$, system (continuously) performs:
set $\hat{x}(t) = x(t_\ell)$
set $e(t) = \hat{x}(t) - x(t)$
if $\|e(t)\| = \sigma \frac{\|Lx(t)\|}{\|L\|}$ then
    set $t_{\ell+1} = t$
    set $\hat{x}(t) = x_i(t_{\ell+1})$
    set $\ell = \ell + 1$
end if
set $u(t) = -L\hat{x}(t)$
```

Table 2 Centralized event-triggered control.

In other words, given a control update at time t_ℓ, the next time $t_{\ell+1}$ is given by

$$t_{\ell+1} = \min\{t' > t_\ell \mid \|e(t')\| = \sigma \frac{\|Lx(t')\|}{\|L\|}\}.$$

The algorithm is formalized in Table 2.

The proof of convergence to the desired state then follows directly from the proof of Theorem 0.3.1 and the fact that the sum of all states is still an invariant quantity. Furthermore, the authors in [17] are able to rule out the existence of Zeno behavior (formally defined below) by showing there exists a positive time

$$\tau = \frac{\sigma}{\|L\|(1+\sigma)}$$

bounding the inter-event times, i.e.,

$$t_{\ell+1} - t_\ell \geq \tau$$

for all $\ell \in \mathbb{Z}_{\geq 0}$.

Definition 0.4.3 (Zeno behavior) *If there exists $T > 0$ such that $t_\ell \leq T$ for all $\ell \in \mathbb{Z}_{\geq 0}$, then the system is said to exhibit* Zeno behavior.

The centralized event-triggered controller (12) with triggering law (14) relaxes the requirement that agents need to continuously update their control signals, but it still has many issues. One of them is that the event-trigger $f(x, e)$ requires full state information to implement. Next, we provide a distributed solution instead of a centralized one.

0.5
Decentralized event-triggered control

In the previous section we presented a centralized event-triggered control law to solve the multi-agent average consensus problem. Unfortunately, implementing this requires a centralized decision maker and requires all agents in the network to update their control signals simultaneously. In this section we relax this requirement by following [17].

Let us now consider a distributed digital implementation of the ideal controller (9). In this case we assume each agent i has its own sequence of event times $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}}$. At any given time t, let $\hat{x}_i(t) = x_i(t_\ell^i)$ for $t \in [t_\ell^i, t_{\ell+1}^i)$ be the state of agent i at its last update time. The distributed event-triggered controller is then given by

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t)). \tag{15}$$

It is important to note here that the latest updated state $\hat{x}_j(t)$ of agent $j \in \mathcal{N}_i$ appears in the control signal for agent i. This means that when an event is triggered by a neighboring agent j, agent i also updates its control signal accordingly. As in the centralized case, let $e_i(t) = x_i(t_\ell^i) - x_i(t)$ be the state measurement error for agent i. Then, letting $\hat{x} = (\hat{x}_1, \dots, \hat{x}_N)^T$ and $e = (e_1, \dots, e_N)^T$, the closed-loop dynamics of the controller (15) is given by

$$\dot{x}(t) = -L\hat{x}(t) = -L(x(t) + e(t)). \tag{16}$$

The problem can now be formalized as follows.

Problem 0.5.1 (Decentralized event-triggered control) *Given the closed-loop dynamics* (16)*, find an event-trigger for each agent i such that the sequence of times $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}}$ ensures multi-agent average consensus* (10) *is achieved.*

Following [17], to solve this problem we again consider the Lyapunov function

$$V(x) = x^T L x.$$

Given the closed-loop dynamics (16), we have

$$\dot{V} = -\|Lx\|^2 - x^T L L e.$$

As before, we are interested in finding conditions on the error e such that $\dot{V} < 0$ at all times; however, we must now do this in a distributed way. For simplicity, let $Lx \triangleq z = (z_1, \dots, z_N)^T$. Then, expanding out $\dot{V}$ yields

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

Using Young's inequality (1) and the fact that $\mathcal{G}$ is symmetric, we can bound this by

$$\dot{V} \leq -\sum_{i=1}^N (1 - a|\mathcal{N}_i|) z_i^2 + \frac{1}{a} |\mathcal{N}_i| e_i^2$$

for all $a > 0$. Letting $a \in (0, 1/|\mathcal{N}_i|)$ for all i, if we can enforce the error of all agents to satisfy

$$e_i^2 \leq \frac{\sigma_i a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|} z_i^2$$

with $\sigma_i \in (0, 1)$ for all times, we have

$$\dot{V} \leq \sum_{i=1}^{N} (\sigma_i - 1)(1 - a|\mathcal{N}_i|) z_i^2,$$

which is strictly negative for all $Lx \neq 0$. The following decentralized event-trigger ensures this is satisfied at all times.

Theorem 0.5.2 (Decentralized event-triggered control) *Given the closed-loop dynamics* (16)*, if the updates times of each agent i are determined as the times when*

$$f_i(x_i, e_i, \{x_j\}_{j \in \mathcal{N}_i}) \triangleq e_i^2 - \frac{\sigma_i a(1 - a|\mathcal{N}_i|)}{|\mathcal{N}_i|} z_i^2 = 0, \tag{17}$$

with $0 < a < 1/|\mathcal{N}_i|$, then the system achieves multi-agent average consensus.

Note that the trigger (17) can be evaluated by agent i using only information about its own and neighbors' states. The algorithm is formalized in Table 3.

```
At times $t \in [t_\ell^i, t_{\ell+1}^i)$, agent $i$ (continuously) performs:
set $z_i(t) = \sum_{j \in \mathcal{N}_i} (x_i(t) - x_j(t))$
set $e_i(t) = \hat{x}_i(t) - x_i(t)$
if $e_i(t)^2 = \frac{\sigma_i a(1-a|\mathcal{N}_i|)}{|\mathcal{N}_i|} z_i(t)^2$ then
    set $t_{\ell+1}^i = t$
    broadcast $\hat{x}_i(t) = x_i(t_{\ell+1}^i)$ to neighbors $j \in \mathcal{N}_i$
    set $\ell = \ell + 1$
end if
set $u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$
```

Table 3 Decentralized event-triggered control.

The proof of convergence to the desired state then directly follows from the proof of Theorem 0.3.1 and the fact that the sum of all states is still an invariant quantity. Furthermore, the authors in [17] are able to show that at all times there exists one agent i for which the inter-event times are strictly positive. Unfortunately, this is not enough to rule out Zeno behavior which is quite problematic, both from a pragmatic and theoretical viewpoint, as the trajectories of the system are no longer well-defined beyond the accumulation point in time.

Remark 0.5.3 (Convergence and Zeno behavior) It should be noted here that when we refer to a "proof of convergence" for any closed-loop dynamics, it is only valid for trajectories do no exhibit Zeno behavior. Consequently, being able to guarantee Zeno behaviors do not occur is extremely important in validating the correctness of a given algorithm. We formalize the definition of Zeno behavior next. •

Remark 0.5.4 (Directed graphs) All the work from [17] has also been extended to consider weight-balanced directed graphs in [18]. For brevity, we defer the discussion on directed graphs to Section 0.6.1.

The decentralized event-triggered controller (15) with triggering law (17) relaxes the requirement that agents need to continuously update their control signals; however, there are still some severe issues. Although each agent now has a local event-triggering condition, it requires continuous information about all of its neighbors to implement it. This is still troublesome in a wireless network setting where this implies continuous communication among agents is still required. We address this next.

0.6 Decentralized event-triggered communication and control

In the previous sections we presented event-triggered *control* laws to determine when control signals should be updated; however, this relied on the continuous availability of some state information. In particular, each agent i requires exact state information about their neighbors $j \in \mathcal{N}_i$ to evaluate the trigger (17) and determine when its control signal u_i should be updated. Instead, we are now interested in developing event-triggered *communication and control* laws such that each agent i must not only determine when to update its control signal but also when to communicate with its neighbors. For simplicity, we refer to communication and control together as 'coordination.'

As in the previous section, we assume each agent i has its own sequence of event times $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}}$. However, these update times now correspond to when messages are broadcast in addition to when control signals are updated. At any given time t, let $\hat{x}_i(t) = x_i(t_\ell^i)$ for $t \in [t_\ell^i, t_{\ell+1}^i)$ be the last broadcast state of agent i. Then, at any given time t, agent i only has access to the last broadcast state $\hat{x}_j(t)$ of its neighbors $j \in \mathcal{N}_i$ rather than exact states $x_j(t)$.

The distributed event-triggered controller is then still given by

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t)). \tag{18}$$

It is important to note here that the latest broadcast state $\hat{x}_j(t)$ of agent $j \in \mathcal{N}_i$ appears in the control signal for agent i. This means that when an event is triggered by a neighboring agent j, agent i also updates its control signal accordingly. As before, let $e_i(t) = x_i(t_\ell^i) - x_i(t)$ be the state measurement error for agent i. Then, letting $\hat{x} = (\hat{x}_1, \dots, \hat{x}_N)^T$ and $e = (e_1, \dots, e_N)^T$, the closed-loop dynamics of the controller (15) is given by

$$\dot{x}(t) = -L\hat{x}(t) = -L(x(t) + e(t)). \tag{19}$$

The problem can now be formalized as follows. However, it should be noted that we are now looking for a strictly *local* event-trigger for each agent i that doesn't require exact information about its neighbors. More specifically, we recall the result of Theorem 0.5.2 and notice that the event-trigger for agent i depends on the exact state $x_j(t)$ of all its neighbors $j \in \mathcal{N}_i$. In this section we are interested in finding a trigger that only depends on the last broadcast information $\hat{x}_j(t)$ instead.

Problem 0.6.1 (Decentralized event-triggered coordination) *Given the closed-loop dynamics* (16)*, find a local event-trigger for each agent i such that the sequence of times* $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}}$ *ensures multi-agent average consensus* (10) *is achieved.*

Here we present two classes of event-triggered coordination solutions to the problem above: time-dependent and state-dependent triggers. The time-dependent event-trigger to solve this problem was first developed in [19] and is presented next. The algorithm is formalized in Table 4.

Theorem 0.6.2 (Decentralized event-triggered coordination (time-dependent)) *Given the closed-loop dynamics* (16)*, if the updates times of each agent i are determined as the times when*

$$f_i(e_i(t), t) \triangleq \|e_i(t)\| - (c_0 + c_1 e^{-\alpha t}) = 0, \tag{20}$$

with constants $c_0, c_1 \geq 0$ *and* $c_0 + c_1 > 0$*, then the system reaches a neighborhood of multi-agent average consensus upper-bounded by*

$$r = \|L\|\sqrt{N}c_0/\lambda_2(L).$$

Moreover, if $c_0 > 0$ *or* $0 < \alpha < \lambda_2(L)$*, then the closed-loop system does not exhibit Zeno behavior.*

The proof of convergence is shown in the appendix; however, we are now also interested in guaranteeing Zeno behavior does not occur to verify the correctness of the algorithm as mentioned earlier.

The main drawback of the event-triggered communication and control law proposed in Theorem 0.6.2 is that although the parameters c_0, c_1, and α play very important roles in the performance of the algorithm (e.g., convergence speed and amount of triggers), there is no good way of choosing these parameters a priori, without any global knowledge. Furthermore, the initial condition also plays an important role in the performance of the algorithm.

In particular we focus our discussion here on the parameters c_0 and α and their effects on convergence and possible Zeno behaviors. We begin with the more desirable $c_0 = 0$ case, as in this case the result of Theorem 0.6.2 states that the system

```
At times $t \in [t^i_\ell, t^i_{\ell+1})$, agent $i$ (continuously) performs:

set $\hat{x}_i(t) = x_i(t^i_\ell)$
set $e_i(t) = \hat{x}_i(t) - x_i(t)$
if $|e_i(t)| = c_0 + c_1 e^{-\alpha t}$ then
    set $t^i_{\ell+1} = t$
    broadcast $\hat{x}_i(t) = x_i(t^i_{\ell+1})$ to neighbors $j \in \mathcal{N}_i$
    set $\ell = \ell + 1$
end if
set $u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$
```

Table 4 Decentralized event-triggered coordination (time-dependent).

will asymptotically achieve exact multi-agent average consensus as defined in (10). However, in this case we require $\alpha < \lambda_2(L)$ to guarantee Zeno behaviors can be avoided and, unfortunately, $\lambda_2(L)$ is a global quantity that requires knowledge about the entire communication topology to compute. There are indeed methods for estimating this quantity in a distributed way (see e.g., [26, 27]), but we do not discuss this here. On the other hand, when $c_0 > 0$ we can guarantee that Zeno behaviors are avoided regardless of our choice of α; however, we lose the asymptotic convergence guarantee. That is, for $c_0 > 0$ we can only guarantee convergence to a neighborhood of the desired average consensus state.

As a result of the above discussion, we see that there is no way the agents can choose the parameters c_0, c_1, and α to ensure asymptotic convergence to the average consensus state while also guaranteeing Zeno executions are avoided. Consequently, more recent works have proposed a local Lyapunov-based event-triggering condition that only relies on currently available information and no exogenous signals (e.g., time). This also naturally aligns when events are triggered with the progression of the task as encoded in the Lyapunov function. The state-dependent event-trigger to solve this problem was first developed in [20] and improved upon in [23] (removed global parameter a requirement); we present this next.

Following [23], to solve this problem we consider a different Lyapunov function,

$$V(x) = \frac{1}{2}(x - \bar{x}\mathbf{1})^T (x - \bar{x}\mathbf{1}),$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^N x_i(0)$ is the average of all initial conditions. Then, given the closed-loop dynamics (16), we have

$$\dot{V} = -x^T \dot{x} - \bar{x}\mathbf{1}^T \dot{x} = -x^T L\hat{x} - \bar{x}\mathbf{1}^T L\hat{x} = -x^T L\hat{x},$$

where we have used the fact that the graph is weight-balanced in the last equality. As before, we are interested in finding conditions on the error e such that $\dot{V} < 0$ at all times; however, we must now do it without access to neighboring state information.

Recalling $e_i(t) = \hat{x}_i(t) - x_i(t)$, we can expand this out to

$$\begin{aligned}\dot{V} &= -\hat{x}^T L\hat{x} + e^T L\hat{x} \\ &= -\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \left(\frac{1}{2}(\hat{x}_i - \hat{x}_j)^2 - e_i(\hat{x}_i - \hat{x}_j)\right).\end{aligned}$$

Using Young's inequality for each product (see [23] for why this choice)

$$e_i(\hat{x}_i - \hat{x}_j) \le e_i^2 + \frac{1}{4}(\hat{x}_i - \hat{x}_j)^2$$

yields

$$\begin{aligned}\dot{V} &\le -\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \left(\frac{1}{2}(\hat{x}_i - \hat{x}_j)^2 - e_i^2 - \frac{1}{4}(\hat{x}_i - \hat{x}_j)^2\right) \\ &= -\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \left(\frac{1}{4}(\hat{x}_i - \hat{x}_j)^2 - e_i^2\right) \\ &= \sum_{i=1}^{N} e_i|\mathcal{N}_i| - \sum_{j\in\mathcal{N}_i} \left(\frac{1}{4}(\hat{x}_i - \hat{x}_j)^2\right).\end{aligned}$$

If we can enforce the error of all agents to satisfy

$$e_i^2 \le \sigma_i \frac{1}{4|\mathcal{N}_i|} \sum_{j\in\mathcal{N}_i} (\hat{x}_i - \hat{x}_j)^2$$

with $\sigma_i \in (0,1)$ for all times, we have

$$\dot{V} \le \sum_{i=1}^{N} \frac{\sigma_i - 1}{4} \sum_{j\in\mathcal{N}_i} (\hat{x}_i - \hat{x}_j)^2,$$

which is strictly negative for all $L\hat{x} \neq 0$. The following decentralized event-trigger ensures this is satisfied at all times.

Theorem 0.6.3 (Decentralized event-triggered coordination (state-dependent)) *Given the closed-loop dynamics* (16)*, if the updates times of each agent i are determined as the times when*

$$f_i(e_i) \triangleq e_i^2 - \sigma_i \frac{1}{4|\mathcal{N}_i|} \sum_{j\in\mathcal{N}_i} (\hat{x}_i - \hat{x}_j)^2 \ge 0, \tag{21}$$

then the system achieves multi-agent average consensus.

It should be noted here that unlike all the other triggers presented so far, this trigger is given by an inequality rather than an equality. This is a result of the state-dependent triggering function that agents use to determine when to communicate. Since agents are asynchronously sending each other messages, the information they have about one another is also changing discontinuously.

```
At times $t \in [t_\ell^i, t_{\ell+1}^i)$, agent $i$ (continuously) performs:

set $\hat{x}_i(t) = x_i(t_\ell^i)$
set $e_i(t) = \hat{x}_i(t) - x_i(t)$
if $e_i(t)^2 \geq \sigma_i \frac{1}{4|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))^2$ then
    set $t_{\ell+1}^i = t$
    broadcast $\hat{x}_i(t) = x_i(t_{\ell+1}^i)$ to neighbors $j \in \mathcal{N}_i$
    set $\ell = \ell + 1$
end if
set $u_i(t) = -\sum_{j \in \mathcal{N}_i} (\hat{x}_i(t) - \hat{x}_j(t))$
```

Table 5 Decentralized event-triggered coordination (state-dependent).

0.6.1 Directed graphs

Up until now we have assumed that the communication graph was always undirected. Here we extend the previous results to cases where the communication graph $\mathcal{G}$ are directed but strongly connected and weight-balanced.

More specifically, we say that agent i can only send messages to its out-neighbors $j \in \mathcal{N}_i^{\text{out}}$. Similarly, it can only receive messages broadcast by its in-neighbors $j \in \mathcal{N}_i^{\text{in}}$. Conveniently, the closed-loop system dynamics is still given by (16) where the only difference now is L is not symmetric. However, because it is weight-balanced we still have that the sum of all states is an invariant quantity,

$$\frac{d}{dt}\left(\mathbf{1}_N^T x(t)\right) = \mathbf{1}_N^T \dot{x}(t) = -\mathbf{1}_N^T L \hat{x}(t) = 0.$$

Remark 0.6.4 (Weight-balanced assumption) It should be noted that the weights of the directed graph for any digital implementations are design parameters that can be chosen to make a given directed communication topology weight-balanced. The works [28, 29] present provably correct distributed strategies that, given a directed communication topology, allow a network of agents to find such weight edge assignments. •

Remarkably, the same analysis from the previous section almost directly follows and admits a similar triggering law. More specifically, it can be shown that if we can enforce the error of all agents to satisfy

$$e_i^2 \leq \sigma_i \frac{1}{4 d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i^{\text{out}}} (\hat{x}_i - \hat{x}_j)^2,$$

with $\sigma_i \in (0,1)$ for all times, we have

$$\dot{V} \leq \sum_{i=1}^{N} \frac{\sigma_i - 1}{4} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\hat{x}_i - \hat{x}_j)^2, \tag{22}$$

which is strictly negative for all $L\hat{x} \neq 0$. The following decentralized event-trigger ensures this is satisfied at all times.

Theorem 0.6.5 (Decentralized event-triggered coordination on directed graphs) *Given the closed-loop dynamics* (16)*, if the communication graph $\mathcal{G}$ is weight-balanced and the updates times of each agent i are determined as the times when*

$$f_i(e_i) \triangleq e_i^2 - \sigma_i \frac{1}{4d_i^{out}} \sum_{j\in\mathcal{N}_i} w_{ij}(\hat{x}_i - \hat{x}_j)^2 \geq 0, \tag{23}$$

then the system achieves multi-agent average consensus.

```
At times $t \in [t_\ell^i, t_{\ell+1}^i)$, agent $i$ (continuously) performs:

set $\hat{x}_i(t) = x_i(t_\ell^i)$
set $e_i(t) = \hat{x}_i(t) - x_i(t)$
if $e_i(t)^2 \geq \sigma_i \frac{1}{4d_i^{\text{out}}} \sum_{j\in\mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t))^2$ then
    set $t_{\ell+1}^i = t$
    broadcast $\hat{x}_i(t) = x_i(t_{\ell+1}^i)$ to in-neighbors $j \in \mathcal{N}_i^{\text{in}}$
    set $\ell = \ell + 1$
end if
set $u_i(t) = -\sum_{j\in\mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t))$
```

Table 6 Decentralized event-triggered coordination on directed graphs.

Unfortunately, most of the algorithms presented here are not guaranteed to avoid Zeno behaviors making them risky to implement on real systems. Moreover, the one algorithm that can in some cases guarantee no Zeno behavior requires some global information. In some cases modifications can be made to theoretically ensure no Zeno behavior occurs; however, there may still be an arbitrarily small amount of time between any two events (see e.g., [23]) making it undesirable from an implementation viewpoint. This is addressed in Remark 0.6.6 below and the following section.

Remark 0.6.6 (Implementation) We note here an important issue regarding the connection between Zeno executions and implementation. In general, dedicated hardware can only operate at some maximum frequency (e.g., a physical device can only broadcast a message or evaluate a function a finite number of times in any finite period of time). This means that ensuring a system does not exhibit Zeno behavior may not be enough to guarantee the algorithm can be implemented on a physical system if the physical hardware cannot match the speed of actions required by the algorithm. More specifically, it is guaranteed that Zeno behavior does not exist if the sequence of times $t_\ell^i \to \infty$ as $\ell \to \infty$; however, this is not as strong as ensuring that there exists a minimum time in between triggers $t_{\ell+1}^i - t_\ell^i \geq \tau^{\text{min}} > 0$, which is a more pragmatic constraint when considering physical hardware. •

In light of Remark 0.6.6, we consider enforcing a minimum time between events in the following section.

0.7 Periodic event-triggered coordination

Throughout this chapter we have assumed that all event-triggers can be evaluated continuously. That is, the exact moment at which a triggering condition is met, an action (e.g., state broadcast and control signal update) is carried out. However, this may still be an unrealistic assumption when considering digital implementations. More specifically, a physical device cannot continuously evaluate whether a triggering condition has occurred or not. This observation motivates the need for studying sampled-data (or periodically checked) event-triggered coordination strategies.

Specifically, given a sampling period $h \in \mathbb{R}_{>0}$, we let $\{t_{\ell'}\}_{\ell' \in \mathbb{Z}_{\geq 0}}$, where $t_{\ell'+1} = t'_\ell + h$, denote the sequence of times at which agents evaluate the decision of whether to broadcast their state to their neighbors. This type of design is more in line with the constraints imposed by real-time implementations, where individual components work at some fixed frequency, rather than continuously. An inherent and convenient feature of this strategy is the lack of Zeno behavior (since inter-event times are naturally lower bounded by h).

Consequently, we begin by revisiting the result of Theorem 0.6.5. Intuitively, as long as the sampling period h is small enough, the closed-loop system with a periodically checked event-triggering condition will behave similarly to the system with triggers being evaluated continuously. The proof of convergence for the triggering law in Theorem 0.6.5 hinges on the fact that

$$e_i^2(t) \leq \sigma_i \frac{1}{4d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t))^2$$

for all times t. Instead, since we now assume the triggering function 17 can only be evaluated periodically, we have that

$$e_i^2(t_{\ell'}) \leq \sigma_i \frac{1}{4d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{x}_i(t_{\ell'}) - \hat{x}_j(t_{\ell'}))^2 \tag{24}$$

is only guaranteed at the specific times $\{t_{\ell'}\}_{\ell' \in \mathbb{Z}_{\geq 0}}$ at which the triggering function can be evaluated. The algorithm is formalized in Table 7.

It should be noted that this algorithm is identical to the one in Table 6 except it is only executed periodically now rather than continuously. The following result then provides a sufficient condition on how small the period h has to be to guarantee convergence. The result is obtained by analyzing what happens to the Lyapunov function V in between these times.

Theorem 0.7.1 (Periodic event-triggered coordination) *Given the closed-loop dynamics* (16)*, if the communication graph $\mathcal{G}$ is weight-balanced and the update times of each agent i are determined as the times $t' \in \{0, h, 2h, \dots\}$ when*

$$f_i(e_i) \triangleq e_i^2 - \sigma_i \frac{1}{4d_i^{out}} \sum_{j \in \mathcal{N}_i} w_{ij}(\hat{x}_i - \hat{x}_j)^2 \geq 0,$$

and $h \in \mathbb{R}_{>0}$ satisfies

$$\sigma_{\max} + 4hw_{\max}|\mathcal{N}^{\text{out}}_{\max}| < 1, \tag{25}$$

where $w_{\max} = \max_{i \in \{1,\dots,N\}, j \in \mathcal{N}_i^{out}} w_{ij}$ *and* $|\mathcal{N}^{\text{out}}_{\max}| = \max_{i \in \{1,\dots,N\}} |\mathcal{N}^{\text{out}}_i|$, *then the system achieves multi-agent average consensus.*

Note that checking the sufficient condition (25) requires knowledge of the global quantities $\sigma_{\max}$, $w_{\max}$, and $\mathcal{N}^{\text{out}}_{\max}$. Ensuring that this condition is met can either be enforced a priori by the designer or, alternatively, the network can execute a distributed initialization procedure, e.g., [30, 3], to compute these quantities in finite time. Once known, agents can compute h by instantiating a specific formula to select it that is guaranteed to satisfy (25).

```
At times $t \in \{0, h, 2h, \dots\}$, agent $i$ performs:
set $\hat{x}_i(t) = x_i(t^i_\ell)$
set $e_i(t) = \hat{x}_i(t) - x_i(t)$
if $e_i(t)^2 \geq \sigma_i \frac{1}{4d_i^{\text{out}}} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t))^2$ then
    set $t^i_{\ell+1} = t$
    broadcast $\hat{x}_i(t) = x_i(t^i_{\ell+1})$ to neighbors $j \in \mathcal{N}_i^{\text{in}}$
    set $\ell = \ell + 1$
end if
set $u_i(t) = -\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i(t) - \hat{x}_j(t))$
```

Table 7 Periodic event-triggered coordination on directed graphs.

A drawback of this algorithm is that the period h must be the same for all agents, requiring synchronous action. It is not difficult to envision asynchronous versions of this algorithm for which correctness guarantees have not currently been established.

0.8 Conclusions and future outlook

This chapter has presented a high-level overview of the ideas behind event-triggered communication and control applied to multi-agent average consensus problems. Although Table 1 makes it look like there is a complete story concerning event-triggered consensus problems, this is certainly not true as there still remain many issues to be addressed regarding asynchronism, guarantees on non-Zeno behavior, and practical considerations. There are indeed still exciting new directions being explored at the time of writing that would only serve to expand this table in the future. For instance, all event-triggered protocols discussed in this chapter assume that all agents are able to ‘listen’ for incoming messages at all times. In other words, when a message is broadcast by an agent i, this message is immediately received by a neighboring agent $j \in \mathcal{N}_i$ who immediately (or within some reasonable time due to delays, etc.) reacts to this event by updating its control signal. However, this may not be possible in all scenarios which presents a whole new set of technical challenges. For example, some recent preliminary results have been developed in this setup motivated by the need for coordinating submarines [31, 32], where agents can only communicate when they are at the surface of the water. While a submarine is submerged, any message broadcast by another submarine cannot be received until it resurfaces.

References

1 Olfati-Saber, R., Fax, J.A., and Murray, R.M. (2007) Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, **95** (1), 215–233.

2 Bullo, F., Cortés, J., and Martínez, S. (2009) *Distributed Control of Robotic Networks*, Applied Mathematics Series, Princeton University Press. Electronically available at `http://coordinationbook.info`.

3 Ren, W. and Beard, R.W. (2008) *Distributed Consensus in Multi-Vehicle Cooperative Control*, Communications and Control Engineering, Springer.

4 Mesbahi, M. and Egerstedt, M. (2010) *Graph Theoretic Methods in Multiagent Networks*, Applied Mathematics Series, Princeton University Press.

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

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

7 Mazo Jr., M. and Tabuada, P. (2011) Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, **56** (10), 2456–2461.

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

9 Stöker, C., Vey, D., and Lunze, J. (2013) Decentralized event-based control: Stability analysis and experimental evaluation. *Nonlinear Analysis: Hybrid Systems*, **10**, 141–155.

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

11 Heemels, W.P.M.H. and Donkers, M.C.F. (2013) Model-based periodic event-triggered control for linear systems. *Automatica*, **49** (3), 698–711.

12 Meng, X. and Chen, T. (2013) Event based agreement protocols for multi-agent networks. *Automatica*, **49** (7), 2125–2132.

13 Zhong, M. and Cassandras, C.G. (2010) Asynchronous distributed optimization with event-driven communication. *IEEE Transactions on Automatic Control*, **55** (12), 2735–2750.

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

15 Wang, X. and Lemmon, M.D. (2009) Self-triggered feedback control systems with finite-gain L_2 stability. *IEEE Transactions on Automatic Control*, **54** (3), 452–467.

16 Nowzari, C. and Cortés, J. (2016) Team-triggered coordination for real-time control of networked cyberphysical systems. *IEEE Transactions on Automatic Control*, **61** (1), 34–47.

17 Dimarogonas, D.V., Frazzoli, E., and Johansson, K.H. (2012) Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, **57** (5), 1291–1297.

18 Liu, Z., Chen, Z., and Yuan, Z. (2012) Event-triggered average-consensus of multi-agent systems with weighted and direct topology. *Journal of Systems Science and Complexity*, **25** (5), 845–855.

19 Seybotha, G.S., Dimarogonas, D.V., and Johansson, K.H. (2013) Event-based broadcasting for multi-agent average consensus. *Automatica*, **49** (1), 245–252.

20 Garcia, E., Cao, Y., Yu, H., Antsaklis, P., and Casbeer, D. (2013) Decentralised event-triggered cooperative control with limited communication. *International Journal of Control*, **86** (9), 1479–1488.

21 Guo, G., Ding, L., and Han, Q. (2014) A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems. *Automatica*, **50**, 1489–1496.

22 Meng, X., Xie, L., Soh, Y.C., Nowzari, C., and Pappas, G.J. (2015) Periodic event-triggered average consensus over directed graphs, in *IEEE Conf. on Decision and Control*, Osaka, Japan, pp. 4151–4156.

23 Nowzari, C. and Cortés, J. (2016) Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica*, **68**, 237–244.

24 Mazo Jr., M., Anta, A., and Tabuada, P. (2009) On self-triggered control for linear systems: Guarantees and complexity, in *European Control Conference*, Budapest, Hungary, pp. 3767–3772.

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

26 Aragues, R., Shi, G., Dimaragonas, D.V., Sagues, C., and Johansson, K.H. (2012) Distributed algebraic connectivity estimation for adaptive event-triggered consensus, in *American Control Conference*, Montreal, Canada, pp. 32–37.

27 Yang, P., Freeman, R.A., Gordon, G.J., Lynch, K.M., Srinivasa, S.S., and Sukthankar, R. (2010) Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, **46** (2), 390–396.

28 Gharesifard, B. and Cortés, J. (2012) Distributed strategies for generating weight-balanced and doubly stochastic digraphs. *European Journal of Control*, **18** (6), 539–557.

29 Rikos, A., Charalambous, T., and Hadjicostis, C.N. (2014) Distributed weight balancing over digraphs. *IEEE Transactions on Control of Network Systems*. To appear.

30 Lynch, N.A. (1997) *Distributed Algorithms*, Morgan Kaufmann.

31 Adaldo, A., Liuzza, D., Dimarogonas, D.V., and Johansson, K.H. (2015) Control of multi-agent systems with event-triggered cloud access, in *European Control Conference*, Linz, Austria, pp. 954–961.

32 Nowzari, C. and Pappas, G.J. (2016) Multi-agent coordination with asynchronous cloud access, in *American Control Conference*, Boston, MA. To appear.

33 Khalil, H.K. (2002) *Nonlinear Systems*, Prentice Hall, 3rd edn..

Appendix

Proof of Theorem 0.3.1

Consider the Lyapunov function

$$V(x) = \frac{1}{2}x^T Lx.$$

Then, given the dynamics (8) and the continuous control law (9),

$$\dot{V}(x) = x^T L\dot{x} = -x^T L^T Lx = -\|Lx\|^2,$$

where we have used the fact that L is symmetric. It is now clear that using the continuous control law (9) we have $\dot{V}(x) < 0$ for all $Lx \neq 0$. Using LaSalle's Invariance Principle [33], it can then be shown that

$$x(t) \to \{Lx = 0\} = \{x_i = x_j \forall i, j \in \{1, \dots, N\}\}$$

as $t \to \infty$. Combining this with the fact that the sum of all states is an invariant quantity concludes the proof,

$$\frac{d}{dt}\left(\mathbf{1}_N^T x(t)\right) = \mathbf{1}_N^T \dot{x}(t) = -\mathbf{1}_N^T Lx(t) = 0.$$

■

Proof of Theorem 0.6.2

Let $\delta(t) = x(t) - \bar{x}\mathbf{1}$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^N x_i(0)$ is the average of all initial conditions. Then, $\dot{\delta}(t) = -L\delta(t) - Le(t)$, yielding

$$\delta(t) = e^{-Lt}\delta(0) - \int_0^t e^{-L(t-s)}Le(s)ds.$$

Taking norms,

$$\begin{aligned}\|\delta(t)\| &\leq \|\delta(0)e^{-Lt}\| + \int_0^t \|e^{-L(t-s)}Le(s)\|ds \\ &\leq e^{-\lambda_2(L)t}\|\delta(0)\| + \int_0^t e^{-\lambda_2(L)(t-s)}\|Le(s)\|ds,\end{aligned}$$

where the second inequality follows from [19, Lemma 2.1].

Using the condition

$$|e_i(t)| \leq c_0 + c_1 e^{-\alpha t},$$

it follows that

$$\begin{aligned}\|\delta(t)\| &\leq e^{-\lambda_2 t}\|\delta(0)\| + \|L\|\sqrt{N}\int_0^t e^{-\lambda_2(t-s)}(c_0 + c_1 e^{-\alpha s})ds \\ &= e^{-\lambda_2 t}\left(\|\delta(0)\| - \|L\|\sqrt{N}\left(\frac{c_0}{\lambda_2} + \frac{c_1}{\lambda_2 - \alpha}\right)\right) + e^{-\alpha t}\frac{\|L\|\sqrt{N}c_1}{\lambda_2 - \alpha} + \frac{\|L\|\sqrt{N}c_0}{\lambda_2}.\end{aligned}$$

The convergence result then follows by taking $t \to \infty$. ■

Proof of Theorem 0.7.1

Since (24) is only guaranteed at the sampling times under the periodic event-triggered coordination algorithm presented in Table 7, we analyze what happens to the Lyapunov function V in between them. For $t \in [t_{\ell'}, t_{\ell'+1})$, note that

$$e(t) = e(t_{\ell'}) + (t - t_{\ell'})L\hat{x}(t_{\ell'}).$$

Substituting this expression into $\dot{V}(t) = -\hat{x}^T(t)L\hat{x}(t) + e^T(t)L\hat{x}(t)$, we obtain

$$\begin{aligned}\dot{V}(t) = -\hat{x}^T(t_{\ell'})L\hat{x}(t_{\ell'}) + e^T(t_{\ell'})L\hat{x}(t_{\ell'}) \\ + (t - t_{\ell'})\hat{x}^T(t_{\ell'})L^T L\hat{x}(t_{\ell'}),\end{aligned}$$

for all $t \in [t_{\ell'}, t_{\ell'+1})$. For a simpler exposition, we drop all arguments referring to time $t_{\ell'}$ in the sequel. Following a similar discussion to Section 0.6, it can be shown that

$$\dot{V}(t) \le \sum_{i=1}^{N} \frac{\sigma_i - 1}{4} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i - \hat{x}_j)^2 + (t - t_{\ell'})\hat{x}^T L^T L\hat{x}.$$

Note that the first term is exactly what we have when we are able to monitor the triggers continuously (22).

Using the fact that $\left(\sum_{k=1}^p y_k\right)^2 \le p\sum_{k=1}^p y_k^2$ (which follows directly from the Cauchy-Schwarz inequality), we bound

$$\begin{aligned}\hat{x}^T L^T L\hat{x} &= \sum_{i=1}^{N}\left(\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i - \hat{x}_j)\right)^2 \\ &\le \sum_{i=1}^{N} |\mathcal{N}_i^{\text{out}}| w_i^{\max} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i - \hat{x}_j)^2 \\ &= |\mathcal{N}_{\max}^{\text{out}}| w_{\max} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i - \hat{x}_j)^2, \qquad (26)\end{aligned}$$

where $w_i^{\max} = \max_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}$. Hence, for $t \in [t_{\ell'}, t_{\ell'+1})$,

$$\begin{aligned}\dot{V}(t) &\le \sum_{i=1}^{N}\Big(\frac{\sigma_i - 1}{4} + h w_{\max}|\mathcal{N}_{\max}^{\text{out}}|\Big) \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\hat{x}_i - \hat{x}_j)^2 \\ &\le \Big(\frac{\sigma_{\max} - 1}{2} + 2h w_{\max}|\mathcal{N}_{\max}^{\text{out}}|\Big)\hat{x}^T L\hat{x}.\end{aligned}$$

Then, by using (25), it can be shown that there exists $\mathcal{B} > 0$ such that

$$\dot{V}(t) \le \frac{1}{2\mathcal{B}}\Big(\sigma_{\max} + 4h w_{\max}|\mathcal{N}_{\max}^{\text{out}}| - 1\Big)V(x(t)),$$

which implies the result. See [23] for details. ■