SCIENCE CHINA
Information Sciences

May 2020, Vol. 63 150201:1–150201:18
https://doi.org/10.1007/s11432-019-2637-9

• REVIEW •

Special Focus on Advanced Techniques for Event-Triggered Control and Estimation

How often should one update control and estimation: review of networked triggering techniques

Zhiyong CHEN[1], Qing-Long HAN[2*], Yamin YAN[1] & Zheng-Guang WU[3]

[1]*School of Electrical Engineering and Computing, University of Newcastle, Callaghan* NSW 2308*, Australia;*
[2]*School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne* VIC 3122*, Australia;*
[3]*Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou* 310027*, China*

Received 13 June 2019/Accepted 12 August 2019/Published online 26 February 2020

Abstract Management of resources is a constant topic in industrial systems. How to use minimum communication resources is of particular interest for control and estimation of networked systems. It raises the question for researchers in the field: how often should one update control and estimation? One of the most intelligent approaches is to trigger updates by events. In the literature, event-triggered control and estimation have been widely studied in the last decade. On one hand, events should be triggered sufficiently frequent to maintain system performance; on the other hand, the possibility of Zeno behavior caused by infinite frequency should be avoided. This review aims at revisiting some existing triggering techniques in a unified formulation, separated from system dynamics and control and estimation strategies. It brings readers better understanding of triggering mechanisms, the underlying technical challenges, and some promising future research topics.

Keywords event-triggered control, event-triggered estimation, networked systems, Zeno behavior

Citation Chen Z Y, Han Q-L, Yan Y M, et al. How often should one update control and estimation: review of networked triggering techniques. Sci China Inf Sci, 2020, 63(5): 150201, https://doi.org/10.1007/s11432-019-2637-9

1 Introduction

Greek philosopher Zeno of Elea (490–430 BC) devised a set of philosophical problems, called Zeno's paradoxes, contrary to the evidence of one's senses. One of the paradoxes is "Achilles and the Tortoise". Achilles allows the tortoise a head start of 100 meters in a footrace. Suppose that Achilles and the tortoise run at the speeds of 10 m and 1 m per second, respectively. After 10 s, Achilles will have run 100 m to the tortoise's starting point. Meanwhile, the tortoise has run 10 m. It will then take Achilles further 1 s to run that distance, by which time the tortoise will have advanced farther. Whenever Achilles arrives somewhere the tortoise has been, he still has some distance to go before he can even reach the tortoise. It concludes that Achilles can never overtake the tortoise [1].

Let us put the story in modern control language with $y(t)$ and $u(t)$ being the position and velocity of Achilles, at time t, respectively, which satisfy

$$\dot{y}(t) = u(t), \quad y(0) = -100, \tag{1}$$

and

$$u(t) = 10. \tag{2}$$

* Corresponding author (email: qhan@swin.edu.au)

So, Achilles' trajectory is simply given by $y(t) = -100 + 10t$. Let the tortoise's trajectory be $\mathcal{Y}(t) = t$. On one hand, at the finite time $T = 100/9$ s, one has $y(T) - \mathcal{Y}(T) = 0$ which means that Achilles catches the tortoise. On the other hand, we define the time sequence t_k, $k = 0, 1, 2, \ldots$, with $t_0 = 0$ and

$$t_{k+1} = t_k + \frac{\mathcal{Y}(t_k) - y(t_k)}{10}. \tag{3}$$

It means that at t_{k+1} Achilles arrives at the point where the tortoise has been at t_k, which is called an event. Simple calculation gives $t_k = (100/9)(1 - 0.1^k) < T$. Therefore, within the finite time T, one can observe infinite events with $k \to \infty$. In the paradox, Zeno ambiguously used infinite events to define 'never' that is contrary to one's sense of finite time. Today, for a triggered control system, an event is not just for observation, but to trigger a control action $u(t)$. In a triggering mechanism, without a strictly positive lower bound for inter-event intervals, an infinite number of events may occur in a finite time period, called Zeno behavior.

The concept of event has been widely applied in modern industries, especially, in networked control systems. When events occur, it is unavoidable to analyze their frequency, especially, to exclude Zeno behavior. The introduction of event into control systems can be traced back to the relevant early research using sampled-data control. In practical applications such as multi-robot systems, electrical power systems, natural gas systems, chemical processes, and manufacturing processes, a control task is usually executed in a remote platform via a network. Sampled-data control, or called time-triggered control, is capable of handling this scenario thanks to the well established sampled-data theory (e.g., [2]). However, the sampled-data control approach may sample redundant data owing to the fixed sampling period, which results in waste of network bandwidth and even congestion of communication channels. As a consequence, stability, reliability, and other desired performance of a control system will be degraded owing to the effects such as network-induced delays and packet loss [3, 4]. Readers can refer to more discussion about sampled-data control (e.g., [5]). How to design a suitable controller that not only guarantees desired system performance but also makes use of limited network resources is of significant interest to control researchers and engineers.

The disadvantage of sampled-data control motivates the concept of event-triggered control and estimation that acts "when necessary" rather than at prescribed periods. A triggering mechanism is firstly introduced to schedule the network resources in [6]. The main idea is that whether the current state is transmitted to update the control input depends on whether the designed measurement error exceeds the predefined triggering condition associated with the system state or output or the triggering time. That is to say, if the triggering condition is activated, called occurrence of an event, the state is transferred to update the control action.

State estimation is regarded as the fundamental issue in the study of control systems because it allows a system to acquire as accurate as possible the knowledge of the system's full state and then make informed control decisions. However, estimation is challenging because the system state is often partially measurable and accessible in practice. This is particularly true when some unwanted signals, e.g., unknown disturbance and noise, enter into the system and/or system measurements. These unwanted signals, however, have no significant meaning to the understanding of the system. Furthermore, they may even result in inaccuracies to the outcome of control tasks when a controller is designed based on state estimates.

Over the past decade, research on event-triggered control and estimation has attracted increasing attention and many results have been developed. In particular, overview of event-triggered control and estimation in networked systems using different triggering mechanisms can be found in some recent survey papers [7–10]. Not surprisingly, Zeno behavior is a critical issue in event-triggered control and estimation research and it is also the main theme of this survey paper. In this study, we aim to give a unified problem formulation and revisit the existing triggering techniques within the formulation. It will bring readers better understanding of triggering mechanisms that determine how often one should update control and estimation. On one hand, events should be triggered sufficiently frequent to maintain system performance; on the other hand, the possibility of Zeno behavior caused by infinite frequency should be avoided.

2 Triggering techniques for control of linear networked systems

2.1 A unified problem formulation

We first give a unified problem formulation for event-triggered control systems and then specialize it to linear networked systems in the subsequent subsections. For this purpose, we consider a general control system described by the following equations:

$$\dot{x}(t) = f(x(t), u(t), w(t)), \tag{4}$$
$$y(t) = h(x(t)), \tag{5}$$

where $x(t)$ is the state, $u(t)$ the input, $y(t)$ the measurement output, and $w(t)$ the system uncertainty. It is assumed that the control objective can be achieved by a continuous-time controller:

$$u(t) = k(y(t), \mathcal{Y}(t)), \tag{6}$$

where $\mathcal{Y}(t)$ is the measurement from the external, e.g., network. A typical control objective is to drive the system trajectory to asymptotically approach a steady-state target $x_d(t)$, i.e.,

$$\lim_{t\to\infty} \|x(t) - x_d(t)\| = 0. \tag{7}$$

It includes a traditional stabilization/regulation problem if $x_d(t)$ is pre-specified, as well as a collaborative control problem (e.g., consensus, synchronization, formation) in a networked scenario if $x_d(t)$ is collaboratively determined by the entities of the network.

An event-triggered controller aims to replace the controller (6) by

$$u(t) = k(y(t_k), \mathcal{Y}(t_k)), \quad t \in [t_k, t_{k+1}) \tag{8}$$

for a time sequence $t_k,\ k \in \mathbb{S}$, corresponding to occurrence of specified events while maintaining the control objective. More specifically, denote the set $\mathbb{S} = \{0, 1, \dots, k^*\}$ for a finite integer k^* or $\mathbb{S} = \mathbb{Z}_+$ where $\mathbb{Z}_+$ is the set of non-negative integers. The time sequence satisfies $t_0 = 0$ and $t_k < t_{k+1}$. If $\mathbb{S}$ is a finite set, we let $t_{k^*+1} = \infty$ for complement of notation.

On one hand, the more frequent the events are, the closer is the event-triggered controller (8) to its continuous version (6). So, it is more likely that the control objective can be maintained. On the other hand, over frequent events bring high cost. In particular, Zeno behavior of infinite frequency of events must be avoided. The main effort in the filed is devoted to answer the question: how often should one update control? It hence pushes the development of triggering techniques.

Let us define a so-called event signal that is used to trigger an event:

$$z(t) = \zeta(y(t), \mathcal{Y}(t)). \tag{9}$$

An event-triggering mechanism is formulated in determining the t_k sequence, that is,

$$t_{k+1} = \mathcal{E}(t_k, z(t : t_k, t_{k+1})), \tag{10}$$

where $z(t : t_k, t_{k+1})$ is defined as the time function $z(t), t \in [t_k, t_{k+1})$ and $\mathcal{E}$ is an operator of its arguments. Such an event-triggering mechanism requires continuously monitoring the signal $z(t)$, which may further require continuous communication. The disadvantage of continuous communication is obvious, which in general contradicts the philosophy of event-triggered control. Then, a special case is called a self-triggering mechanism where

$$t_{k+1} = \mathcal{E}(t_k, z(t_k)) \tag{11}$$

and thus the event instant t_{k+1} is precomputed at t_k.

In many situations, it is impractical to continuously monitor the signal $z(t)$ in applying the event-triggering mechanism (10) or to design a self-triggering mechanism (11). A more typical approach is to introduce a broadcast action. A conceivable scenario is that two actions are executed at the triggering instant t_k: (i) the controller (8) is implemented; and (ii) the event signal:

$$z^{\mathrm{b}}(t) = z(t_k), \quad t \in [t_k, t_{k+1}) \tag{12}$$

at the instant is broadcasted. Vice verse, the system also receives the event signal broadcasted by the other entities in the network, denoted as $\mathcal{Z}^{\mathrm{b}}(t)$. It is worth mentioning that the triggering instant for $\mathcal{Z}^{\mathrm{b}}(t)$ is determined by other entities, not necessarily t_k. Therefore, an event-triggering mechanism is given by

$$t_{k+1} = \mathcal{E}(t_k, y(t : t_k, t_{k+1}), z(t_k), \mathcal{Z}^{\mathrm{b}}(t : t_k, t_{k+1})), \tag{13}$$

or especially,

$$t_{k+1} = \mathcal{E}(t_k, z(t_k), \mathcal{Z}^{\mathrm{b}}(t : t_k, t_{k+1})). \tag{14}$$

This mechanism does not continuously monitor the signal $z(t)$, so it is sometimes called self-triggering in literature. However, we prefer not to call so as the event instant t_{k+1} cannot be precomputed at t_k, but depends on receiving the external broadcast $\mathcal{Z}^{\mathrm{b}}(t : t_k, t_{k+1})$. Also, Eq. (13) requires continuously monitoring its own output $y(t)$ although not relying on communication.

In the subsequent sections, we will survey the existing development of triggering techniques in the aforementioned framework. For this purpose, let us first summarize a fundamental criterion in designing these triggering techniques using a simple example. Let

$$p = \phi(k, \mathcal{Y}), \quad \kappa(p) = k(y, \mathcal{Y}), \tag{15}$$

and

$$e(t) = p(t_k) - p(t), \quad t \in [t_k, t_{k+1}) \tag{16}$$

with $e(t_k) = 0$. Then, the controller (8) can be rewritten as

$$u(t) = \kappa(p(t_k)) = \kappa(p(t) + e(t)). \tag{17}$$

With $e(t) = 0$, the controller reduces to the continuous-time one. A typical control design approach is to make the closed-loop system with (8) have the desired control objective, but with perturbation by $e(t)$ in a certain sense. Then, using the small gain philosophy, the closed-loop system can eventually maintain the desired objective if the triggering mechanism guarantees the criterion $\|e(t)\| \leqslant \beta(\|p(t)\|)$ for a specified gain function β. The criterion is assured by the event-triggering (10) of the particular form, with $z(t) = p(t)$,

$$t_{k+1} = \inf_{t > t_k} \{\|e(t)\| = \beta(\|p(t)\|)\}. \tag{18}$$

This small gain philosophy is used in triggering mechanism design and different variations will be discussed in the subsequent subsections, with emphasis on exclusion of Zeno behavior.

2.2 A fundamental tradeoff

Even though the development of triggering techniques for consensus of linear multi-agent systems (MASs) is pretty mature, a complete solution is still inadequate for some fundamental technical bottlenecks. First of all, an MAS can be modeled as

$$\dot{\xi}_i(t) = A\xi_i(t) + B\mu_i(t), \quad i = 1, \ldots, N. \tag{19}$$

Let us consider the control of agent i in the aforementioned framework with $y = x = \xi_i$, $u = \mu_i$, and $\mathcal{Y} = \text{col}(\xi_{s_1}, \xi_{s_2}, \dots, \xi_{s_\ell})$, where $1 \leqslant s_1 < s_2 < \cdots < s_\ell \leqslant N$ and $\{s_1, s_2, \dots, s_N\} = \mathcal{N}_i$. Here, $\mathcal{N}_i$ represents the set of neighbors of agent i.

Let us take the early study [11] as an example to explain how a triggering mechanism is designed for a linear MAS. Define the composite measurement of agent i as

$$p(t) = \phi(y(t), \mathcal{Y}(t)) = \sum_{j \in \mathcal{N}_i} a_{ij}(\xi_i(t) - \xi_j(t)), \tag{20}$$

for some weights a_{ij}. Then, one has

$$e(t) = \sum_{j \in \mathcal{N}_i} a_{ij} \left\{ [\xi_i(t_k) - \xi_i(t)] - [\xi_j(t_k) - \xi_j(t)] \right\}, \quad t \in [t_k, t_{k+1}). \tag{21}$$

A component of $e(t)$ is also defined as follows:

$$\epsilon(t) = \xi_i(t_k) - \xi_i(t), \quad t \in [t_k, t_{k+1}). \tag{22}$$

A variation of (18) is directly constructed as follows:

$$t_{k+1} = \inf_{t > t_k} \{\|\epsilon(t)\| = \sigma \|p(t)\|\}, \tag{23}$$

that is of the event-triggering form (10) with the event signal $z(t) = \text{col}(y(t), p(t))$. The constant σ represents a linear gain function. This mechanism requires continuous communication between neighboring agents for continuously monitoring/measuring its neighbors' states to detect the event using $p(t)$. This issue can be resolved by a self-triggered mechanism in this study. However, another issue is the possible occurrence of Zeno behavior. It is noted that $\epsilon(t_k) = 0$ and it takes $t_{k+1} - t_k$ for $\epsilon(t)$ to change from 0 to $\sigma\|p(t)\|$. So the inter-event interval can be roughly characterized by the lower bound of

$$\frac{\|p(t)\|}{\mathrm{d}\|\epsilon(t)\|/\mathrm{d}t} \tag{24}$$

with the denominator being the change rate of $\|\epsilon(t)\|$. Obviously, this inter-event interval may be zero when the agent i has already reached the control objective with $p(t) = 0$ while $\mathrm{d}\|\epsilon(t)\|/\mathrm{d}t = \mathrm{d}\|\xi_i(t)\|/\mathrm{d}t \neq 0$, resulting Zeno behavior. Similar studies can be found in [12,13]. Alternatively, when $\|\epsilon(t_k)/p(t_k)\| = 0$, it takes $t_{k+1} - t_k$ for $\|\epsilon(t_k)/p(t_k)\|$ to change from 0 to σ. So the inter-event interval can also be roughly characterized by the reciprocal of the upper bound of

$$\frac{\mathrm{d}\|\epsilon(t)/p(t)\|}{\mathrm{d}t}. \tag{25}$$

In the previous example, Zeno behavior occurs when $p(t) = 0$. So, to avoid Zeno behavior, one simple idea is to trigger an event whenever $\|e(t)\|$ in the criterion (18) deviates by a specified constant threshold $\delta > 0$, that is,

$$t_{k+1} = \inf_{t > t_k} \{\|e(t)\| = \delta\}. \tag{26}$$

The idea of using a constant threshold to trigger an event can be found in the early study, e.g., [14]. Zeno behavior can be avoided by noting that a positive lower boundary of $t_{k+1} - t_k$, characterized by

$$\frac{\delta}{\mathrm{d}\|e(t)\|/\mathrm{d}t}, \tag{27}$$

exists if $\mathrm{d}\|e(t)\|/\mathrm{d}t$ is bounded by proper control design. However, the controller (17) with the triggering mechanism (26) instead of (18) cannot achieve the asymptotic control objective (7) but the following practical version:

$$\limsup_{t \to \infty} \|x(t) - x_d(t)\| \leqslant \Delta, \tag{28}$$

for a constant $\Delta > 0$.

In other words, the idea of using a constant threshold is able to exclude Zeno behavior, but makes the system state converge to a ball centered at the target, not asymptotically to the target. This observation can be summarized as a fundamental tradeoff in the development of event-triggered control. There is tradeoff between the lower bound of the inter-event intervals (reciprocal of event frequency) and the steady-state control error. A smaller control error Δ may require a smaller threshold δ that results in smaller inter-event intervals owing to (27).

While a pure constant threshold is used in [14] in determining an event, a similar idea of inclusion of a constant threshold in a state dependent event function can be found in [15]. The adopted criterion is the mix of (18) and (26), that is,

$$t_{k+1} = \inf_{t>t_k} \{\|e(t)\| = \beta(\|p(t)\|) + \delta\}. \tag{29}$$

It is not surprising to see that the control objective is obtained in the practical sense of (28), while Zeno behavior is excluded.

The tradeoff has also been demonstrated in other studies on event-triggered control of MASs, using different strategies of adding a threshold in an event function for excluding Zeno behavior. For instance, consensus of first-order integrators is studied in [16] while that of general linear MASs in [17]. In particular, for the MAS (19) studied in [17], $p(t)$ is defined by (20) and $e(t)$ by (21). Let $\varepsilon(t)$ be the composite vector of the element of $e(t)$ as follows:

$$\begin{aligned} &\varepsilon(t) = \mathrm{col}(\epsilon_1(t), \epsilon_2(t)),\\ &\varepsilon_1(t) = y(t_k) - y(t),\\ &\varepsilon_2(t) = \mathcal{Y}(t_k) - \mathcal{Y}(t), \quad t \in [t_k, t_{k+1}). \end{aligned} \tag{30}$$

Then, an event-triggering mechanism of the form (10) with $z(t) = \mathrm{col}(y(t), \mathcal{Y}(t))$ can be directly constructed as follows:

$$t_{k+1} = \inf_{t>t_k} \{\|\varepsilon(t)\| = \sigma\|p(t)\| + \delta\}, \tag{31}$$

for a constant $\delta > 0$. As a result, practical consensus is achieved in the sense of (28) for $x_d(t)$ being the agreed trajectory for all the agents, while Zeno behavior is avoided.

The implementation of the triggering mechanism (31) requires continuous communication for monitoring $\mathcal{Y}(t)$. A more conservative triggering function is also constructed in the same paper, relying on broadcast of neighbors' inputs. Specifically, the event-triggering mechanism takes the form (13) with $z(t) = \mathrm{col}(y(t), p(t))$ and $\mathcal{Z}^{\mathrm{b}} = \mathrm{col}(\mu^{\mathrm{b}}_{s_1}, \mu^{\mathrm{b}}_{s_2}, \ldots, \mu^{\mathrm{b}}_{s_\ell})$, that is,

$$t_{k+1} = \inf_{t>t_k} \{\|\varepsilon_1(t)\| + \rho(t, \mathcal{Z}^{\mathrm{b}}) = \sigma\|p(t_k)\| + \delta\}, \tag{32}$$

where ρ is an explicitly precomputed function and $\mu^{\mathrm{b}}_j(t)$ is the input of the neighbor j, broadcasted at its own triggering instant, not necessarily t_k. In this mechanism, the agent needs to continuously monitor its own state $y(t)$, not relying on communication though, and receive the broadcast $\mathcal{Z}^{\mathrm{b}}$ from the neighbors.

With either (31) or (32), Zeno behavior is excluded but the tradeoff implies the existence of consensus error Δ in (28). The size of consensus error is adjustable by choosing suitable design parameters according to some explicit formula developed in a recent study [18].

2.3 Is there a free lunch?

Both exclusion of Zeno behavior and asymptotic control accuracy are desirable in an event-triggered control scenario, albeit restricted by the aforementioned tradeoff. Researchers have put great efforts to seek a method to achieve both without paying additional cost. Such a solution that solves the tradeoff without any extra cost is like a free lunch that has yet to be cooked in the existing literature, to the best of our knowledge.

Some results are claimed in literature, but there may exist flaws in the technical development. For instance, a technical flaw in proving exclusion of Zeno behavior in [19] is pointed out in [20]. We also believe that an incorrect statement may exist in the proof of [21], where a clear-cut result on consensus of a general linear MAS is developed. In particular, a triggering mechanism like (18) is used in the study, with the notation introduced in this study. Then, it is clear that the inter-event interval can be roughly characterized by

$$\frac{\|p(t)\|}{\mathrm{d}\|e(t)\|/\mathrm{d}t}. \tag{33}$$

It is noted that $\mathrm{d}\|e(t)\|/\mathrm{d}t$ depends on $\dot{\xi}_i(t)$ and $\dot{\xi}_j(t)$, or alternatively, on $p(t)$ and $\mu_j(t)$. So, Eq. (33) can be rewritten as

$$\frac{\|p(t)\|}{\|p(t)\| + \sum_{j\in\mathcal{N}_i}\|\mu_j(t)\|}, \tag{34}$$

whose lower bound is unlikely to determine as $\mu_j(t)$, the input to its neighbors, is independent of $p(t)$. It seems that the proof given in [21] confuses $\lim_{k\to\infty} p(t_k) = 0$ and $p(t_k) = 0$ and makes the wrong assumption that

$$p(t) = 0, \quad \dot{p}(t) = 0, \tag{35}$$

which relates the upper bound of $\sum_{j\in\mathcal{N}_i}\|\mu_j(t)\|$ to $p(t_k)$; e.g., Eq. (40) of [21]. As a result, the lower bound of (34) is explicitly calculated.

The development in [22] follows the similar paradigm where an error may also exist in the proof. A triggering mechanism similar to (23) is used in the study and the inter-event interval can be roughly characterized by (25). Again, it is unlikely to determine the upper bound of (25), that is the change rate of $\|e_i(t)/z_i(t)\|$ using the notation given in [22], because $\dot{z}_i(t)$ depends on the states and inputs to the neighbors. However, the change rate of $\|e(t)/z(t)\|$ of the whole network is used instead in the proof. A gap exists between the change rate of $\|e_i(t)/z_i(t)\|$ and that of $\|e(t)/z(t)\|$.

2.4 Costs of different approaches

Next, let us revisit various approaches in achieving the two targets of exclusion of Zeno behavior and asymptotic control performance, at different additional costs.

(1) The cost of the first approach is using a synchronous clock mechanism. For example, in [23], the control input for an agent is updated both at its own event time as well as that of its neighbors, which results in a synchronous triggering clock for all the agents. In [24], during a certain time interval, an agent receives new information from the broadcast of a neighbor and then immediately broadcasts its own state, assuming no time-delay in information broadcasting. It also results in a synchronous broadcasting clock for a set of connected agents. Specifically, the following triggering mechanism:

$$t_{k+1} = \inf_{t>t_k}\{\|\epsilon(t)\| = \sigma\|p^{\mathrm{b}}(t)\|\} \tag{36}$$

is used in [24] where

$$p^{\mathrm{b}}(t) = \sum_{j\in\mathcal{N}_i} a_{ij}(\xi_i^{\mathrm{b}}(t) - \xi_j^{\mathrm{b}}(t)). \tag{37}$$

It is of the form (13) with $z(t) = y(t)$ and $\mathcal{Z}^{\mathrm{b}} = \mathrm{col}(\xi_{s_1}^{\mathrm{b}}, \xi_{s_2}^{\mathrm{b}}, \ldots, \xi_{s_\ell}^{\mathrm{b}})$ where ξ_j^{b} is the state of the neighbor j, broadcasted at its own triggering instant, not necessarily t_k. Three cases are discussed in the setting. (i) If an agent does not receive any broadcast after t_k, it means that $\xi_j^{\mathrm{b}}(t) = \xi_j(t_k)$ is a constant. It together with the fact that $\xi_i^{\mathrm{b}}(t) = \xi_i(t_k)$ is a constant, implies that $\|p^{\mathrm{b}}(t)\|$ and hence $\mathrm{d}\|\epsilon(t)\|/\mathrm{d}t$ are constants. Then, the inter-event interval can be lower bounded by a finite value:

$$\frac{\|p^{\mathrm{b}}(t)\|}{\mathrm{d}\|\epsilon(t)\|/\mathrm{d}t}. \tag{38}$$

(ii) If an agent does not receive any broadcast until $t_k + \delta$ for a fixed constant δ, then the inter-event interval is lower bounded by either (38) or δ, excluding Zeno behavior. (iii) If an agent receives a broadcast within $(t_k, t_k + \delta]$, it immediately broadcasts its own state. This synchronous broadcasting operation guarantees that no agent will broadcast its state by infinite times in a finite interval, excluding Zeno behavior.

(2) The idea of inclusion of a constant threshold in an event function has been discussed in [14, 15]. From (26) and (29), the non-zero constant threshold sets the lower bound of inter-event intervals, thus excluding Zeno behavior, but results in a certain steady-state control error. An extension of this idea is to replace the constant threshold by a time-dependent function illustrated as follows:

$$t_{k+1} = \inf_{t>t_k} \{\|e(t)\| = g(t)\}, \tag{39}$$

where $g(t)$ is a positive time-dependent function and it decreases to zero with time. On one hand, $g(t) > 0$ for all t sets the lower bound of inter-event intervals; on the other hand, $\lim_{t\to\infty} g(t) = 0$, drives the steady-state control error to zero. This approach is used in [25] for the state independent triggering case where $g(t) = c\exp(-\alpha t)$ with $c, \alpha > 0$, under the condition that the system states should converge faster than the time-dependent function decreases. Thus, a positive lower bound of the inter-event interval is guaranteed depending on the agent's upper bounded control input which relies on time and the initial states. Another variation can be found in [26] where $g(t) = \alpha\beta^l,\ t \in [lh, (l+1)h)$, for a specified time period $h > 0$ and two parameters $\alpha > 0$ and $0 < \beta < 1$.

One cost of this approach is that the time-dependent function makes the closed-loop system non-autonomous and the lower bound of the inter-event intervals relies on time or the initial states. The application of this approach can also be found in [27] where a time-dependent function is generated by an auxiliary dynamical system and in [28] where a general framework is provided with great flexibility in choosing a time dependent function. It is also noted that a tradeoff exists between the inter-event interval and the convergence rate of consensus process.

(3) The third approach is to combine event-triggered control and sampled-data control. For example, in [29–31], the sampling period for all agents is specified and the sampled data are used for event detection. In this way, the continuous event detection in numerous previous studies ([11, 12, 25]) is avoided.

Suppose the sampling period is $h > 0$. The triggering mechanism takes the form:

$$t_{k+1} = \inf_{t=t_k+lh, l>0} \{\|\epsilon(t)\| = \sigma\|p(t)\|\}. \tag{40}$$

Whether sampled-data of an agent should be used for actuation and broadcasted at the sampling instant depends on whenever an event is detected, which makes the protocol different from sampled-data control. In general, the information broadcasting times will be fewer than those of traditional sampled-data control. In this setting, it is easy to prove that the inter-event interval is at least one sampling period h, excluding Zeno behavior. The cost is again the existence of a synchronous clock where the utilization of a common sampling period for all agents might be restricted, especially in distributed networks.

(4) The fourth approach also takes the advantage of sampled-data control in a natural setting that the lower bound of inter-event intervals is at least one sampling period [31]. But this approach does not require a synchronous clock. The idea is to trigger an event when a normal triggering condition is satisfied (which by itself may result in Zeno behavior) and the inter-event interval is larger than a specified sampling period (called a fixed timer). For recent advances on this approach, the readers can refer to the survey papers [32, 33].

This approach is first used in [31, 34, 35] for event-triggered stabilization problems. It is also used for sampled-data consensus of multi-agent systems in [30], and consensus of first-order integrators in [36] for a network of an undirected and connected graph. The result is further generalized to general linear MASs in [37], also for an undirected and connected network; and in [38] for a leader-following network containing an undirected subgraph of the follower network. Both control targets are guaranteed in these scenarios. It is noted that, in [38], an agent updates the control protocol at its own triggering instants and its neighbors' triggering instants as well. The approach is also studied for general linear MASs with a directed network in [39] where a clear-cut condition for triggering event is proposed in terms of an explicit mathematical expression. The specific triggering mechanism developed in [39] is revisited below to illustrate the idea of this approach, that is,

$$t_{k+1} = \max\{\bar{t}_{k+1}, t_k + \tau\}, \tag{41}$$

where $\tau > 0$ is the fixed timer and

$$\bar{t}_{k+1} = \mathcal{E}(t_k, z(t_k), \mathcal{Z}^{\rm b}(t : t_k, t_{k+1})) \tag{42}$$

is of the form (14) with $z = p$ and $\mathcal{Z}^{\rm b} = {\rm col}(p^{\rm b}_{s_1}, p^{\rm b}_{s_2}, \ldots, p^{\rm b}_{s_\ell})$ where $p^{\rm b}_j$ is the specified state p_j of the neighbor j, broadcasted at its own triggering instant, not necessarily t_k.

This approach combines time-triggering and event-triggering strategies. The minimum inter-event interval is lower bounded by the positive fixed timer τ, excluding Zeno behavior. The cost is that the global information about the network, such as the network size and the dimension of the agent state, is required in determining the fixed timer. In our opinion, this cost is minimal in the existing approaches.

3 Triggering techniques for estimation in linear networked systems

Triggering techniques have also paid much attention to state estimation during the past decade. State estimation aims to estimate the inaccessible state of a dynamical system. Depending on the assumed properties of disturbance and noise, the existing state estimation methods can be roughly categorized into Kalman state estimation, H_∞ state estimation and set-membership state estimation, as surveyed in [9] in a distributed state estimation context. Introducing triggering techniques to state estimation is promising for reducing superfluous resource consumption such as sensor power and network bandwidth, whereas it is essentially challenging because of an inaccessible full state signal of the system and the presence of disturbance and noise. Different triggering mechanisms have been developed to answer a major question: how often should one update estimation? In the following, we present a brief survey on event-based state estimation for linear networked systems.

3.1 Event-triggered Kalman state estimation

When system disturbance and noise are known to be Gaussian, Kalman state estimation (or called Kalman filtering) represents an effective and efficient method for estimating system state. Boosted by advanced communication technologies, Kalman filtering for networked systems has become a practical and interesting topic. However, exchanging measurements from sensor to estimator consumes a lot of energy. An efficient way to reduce the energy consuming is to minimize the number of data transmission and computational complexity. In doing so, an event sampling strategy is proposed in [40], where it is pointed out that if event sampling is not accounted for in the design of an estimator, it can cause undesirable behaviors, e.g., a diverging error-covariance matrix. Let $y(t) \in \mathbb{R}^m$ be the system measurement vector. Define a time-measurement-space $\mathcal{H} \in \mathbb{R}^{m+1}$ as

$$\mathcal{H} := \left\{ \begin{pmatrix} y(t) \\ t \end{pmatrix} \middle| y(t) \in \mathbb{R}^m,\ t \in \mathbb{R} \right\}. \tag{43}$$

Suppose that the sequence of event instants is given as $\{t_0, t_1, t_2, \ldots, t_k, \ldots,\}$. Let $\mathcal{H}_k$ denote a subset of $\mathcal{H}$ with $t \in [t_{k-1}, t_k)$. Then the next event instant t_k for given t_{k-1} can be calculated by

$$t_k := \inf \left\{ t \in \mathbb{R}_+ \middle| t > t_{k-1},\ \begin{pmatrix} y(t) \\ t \end{pmatrix} \notin \mathcal{H}_k \right\} \tag{44}$$
$$\text{s.t.} \begin{pmatrix} y(t_{k-1}) \\ t_{k-1} \end{pmatrix} \in \text{int}(\mathcal{H}_k),$$

where int($\mathcal{H}_k$) means the interior of $\mathcal{H}_k$. The scheme (44) presents a general framework for event sampling strategies because it is not restricted to a specific event criterion. Thus, it is suitable for any type of event sampling strategies. However, it just makes sense from the theoretical perspective. In fact, how to know $\binom{y(t)}{t} \notin \mathcal{H}_k$ is quite difficult in practice. That is, when to sample system measurement signals according to (44) is challenging, which is not analyzed in [40]. Instead, the authors in [40] applied (44) to discrete-time networked systems, where system measurements are in the form of discrete-time signals, with Zeno behavior being excluded naturally. For such systems, an event-based state estimator, namely Gaussian sum filter, with a hybrid update scheme is proposed to avoid a diverging error-covariance, where at an event instant, the estimated state is updated using the measurement, while at a synchronous time instant, the update is based on $\mathcal{H}_{k|t}$ [40]. This idea has been further developed in the recent years [41,42].

It should be mentioned that even though a number of results on event-based Kalman state estimation are derived for discrete-time networked systems, to the best of the authors' knowledge, no results on this issue have been reported for networked systems with continuous-time plants.

3.2 Event-triggered H_∞ state estimation

H_∞ state estimation allows disturbance and noise to be unknown but energy bounded. Unlike Kalman state estimation, another significant feature is that the system model is not necessarily exactly known for H_∞ state estimation. Thus, H_∞ state estimation has gained applications in a wider range of fields. Its objective is to design a suitable H_∞ filter to achieve a prescribed H_∞ performance level γ (> 0). On one hand, the resultant filtering error system is asymptotically stable, and on the other hand, the H_∞ norm of the filtering error system should be minimized to ensure an L_2-induced gain from the noise signals to the estimation error less than γ.

Intuitively, it is not difficult to extend those triggering techniques presented in Section 2 to deal with H_∞ state estimation of networked systems. First, one can design a desired H_∞ filter without introducing event-triggering techniques. Then, an emulation method can be used to devise an event-sampling scheme such that (1) the filtering error system is asymptotically stable, and (2) the L_2-induced gain is less than γ. Nevertheless, such an event-sampling scheme is difficult to devise because Zeno behavior is difficult to exclude. In fact, the minimum inter-event interval for event-triggered control systems is deeply analyzed in [15], where it is revealed that the minimum inter-event interval is possibly zero even though small external disturbances are imposed on the system. Consequently, an emphasis of event-based H_∞ state estimation has been placed on the use of sampled-data-based event-triggered transmission schemes [31, 43–45] rather than event-sampling schemes.

A sampled-data-based event-triggered transmission scheme can be devised through two steps. First, let the system measurements be sampled with a period $h > 0$. Then, whether the sampled measurement will be released to a communication network is determined by an event-triggering condition. Hence, one does not need to concern about the minimum inter-event interval because all events are triggered at the sampling instants. The key issue on this scheme is how to define an event-triggering condition. Inspired by [31], a relative-error-based event-triggering scheme is presented for H_∞ state estimation [43, 44, 46]. Let t_kh, $\varepsilon \in [0, 1)$ and $W > 0$ be the current event-triggering instant, a threshold and a weighting matrix, respectively. An event-triggering condition reads

$$\|W^{\frac{1}{2}}[y(t_kh + jh) - y(t_kh)]\|_2 < \varepsilon\|W^{\frac{1}{2}}y(t_kh)\|_2, \tag{45}$$

where $j = 1, 2, \ldots$. Then the next event-triggering time $t_{k+1}h$ can be calculated by

$$\begin{cases} t_{k+1}h = t_kh + \min\{jh | \text{ (45) is violated}\}, \\ t_0 = 0. \end{cases} \tag{46}$$

This event-triggering scheme can be interpreted as: As long as the relative error between $W^{\frac{1}{2}}y(t_kh)$ and the current sampled data $W^{\frac{1}{2}}y(t_kh + jh)$ exceeds the threshold ε, an event is generated, which makes sense. One of the significant characteristics of this scheme is that the resultant filtering error system can be modeled as a time-delay system with a quadratic constraint. As a result, an event-based H_∞ filter can be designed such that the expected objective aforementioned can be fulfilled. The key idea is to define two piecewise continuous functions $d(t)$ and $\delta(t)$ on the interval $\mathcal{I}_{k,j} \triangleq [(t_k + (j-1))h, (t_k + j)h)$ with $j \geqslant 1$:

$$\begin{cases} d(t) = t-(t_k+(j-1))h, \\ \delta(t) = y(t_k h)-y((t_k+(j-1))h), \end{cases} t \in \mathcal{I}_{k,j}, \tag{47}$$

which leads to $y(t_k h) = \delta(t) + y(t - d(t))$. With these two functions, for linear physical plants, the filtering error system is readily modeled as a linear system with the time delay $d(t) \in [0, h]$. Moreover, the event-triggering condition (45) can be described by

$$\|W^{\frac{1}{2}}\delta(t)\|_2 < \varepsilon\|W^{\frac{1}{2}}[\delta(t)+y(t-d(t))]\|_2, \quad t \in \mathcal{I}_{k,j}.$$

In this way, using the S-procedure, the above quadratic constraint can be exploited in the stability analysis and filter design of time-delay systems. Notice that the time delay $d(t)$ is upper-bounded by a sampling period h. Thus, a small sampling period can easily ensure the feasibility of a stability criterion for the filtering error system. Boosted by the advantages, the above sampled-data-based event-triggered transmission scheme has been employed to further deal with dissipativity estimation [47, 48].

Recent development on the event-triggering condition (45) is to introduce

$$\left\|W_1^{\frac{1}{2}}[y(s_i) - y(\tilde{t}_k)]\right\|_2 < \varepsilon \left\|W_2^{\frac{1}{2}} y(\tilde{t}_k)\right\|_2 \tag{48}$$

as a modified event-triggering condition, where $\tilde{t}_k$ is the latest event-triggering instant, $s_i \geqslant \tilde{t}_k$ with $\{s_1, s_2, \ldots, s_j, \ldots\}$ being the sequence of sampling instants, and W_1 and W_2 are weighting matrices. Clearly, the difference between (45) and (48) lies in two aspects: On one hand, signal sampling may be not periodic, and on the other hand, the weight matrix on the two sides of (45) may be different. Now, we make some comments on the event-triggering condition (48). First, it seems that the condition (48) is more general than (45). However, it is possible that the event condition (48) acts as a time-triggering scheme because there may exist $W_1 > 0$ and $W_2 > 0$ such that Eq. (48) is violated at all sampling instants. Moreover, insightful analysis reveals that the condition (48) does not make practical sense. In fact, if $W_1 \neq W_2$,

$$\frac{\|W_1^{\frac{1}{2}}[y(s_i) - y(\tilde{t}_k)]\|_2}{\|W_2^{\frac{1}{2}} y(\tilde{t}_k)\|_2}$$

is neither a relative error nor an absolute error (in the sense of vector norms). Second, it is reasonable to change periodic sampling to aperiodic sampling. Nevertheless, how to perform aperiodic sampling for event-based state estimation raises a challenging problem. What is more, how to derive a time-delay system model for the filtering error system needs to be reconsidered.

3.3 Event-triggered set-membership state estimation

The celebrated Kalman state estimation method necessitates a priori statistical properties of the disturbance and noise as well as an accurate system model without uncertainties. If these requirements are not met, Kalman filters may lead to poor performance [49]. The H_∞ state estimation method, on the other hand, provides a bound for the worst-case estimation error of the system. Additionally, both the methods can only obtain a pointwise estimation, namely a single vector, at each instant of time. In other words, the true system state cannot be guaranteed to within some reliable confidence region, namely a set in state space, at each instant of time because the estimation has no hard bounds [50]. In many real-world applications such as target tracking and system guidance and navigation, however, the system state requires a confidence level of 100% to be estimated. This stimulates the development of set-membership state estimation (also known as ellipsoidal or set-valued estimation). The principle behind a set-membership state estimation method is that given an assumption of unknown-but-bounded disturbance and noise signals, a bounding ellipsoidal set in state space is computed which guarantees the enclosing of the true state of the system at each instant of time.

In the context of event-triggered state estimation, it is also true that there is a tradeoff between enlarging the inter-event intervals and achieving accurate estimation performance. As discussed in Section 2, to increase inter-event intervals, some existing triggering mechanisms can be improved by introducing a small positive constant threshold, a time-dependent function and some auxiliary dynamic variables (or called offset variables) into the event triggering functions. However, the steady-state estimation errors of some existing state estimators in this case need to be evaluated in a practical sense, especially for H_∞ state estimators. In contrast, the boundedness of the estimation errors can be readily verified within an event-triggered set-membership state estimation framework as the calculated state estimation is a bounding set in state-space rather than a single vector, as recently shown in [51]. To elaborate such an event-triggered set-membership state estimation strategy, we consider a discrete-time linear time-varying networked system described by the following state-space equations:

$$x(t+1) = A(t)x(t) + B(t)w(t), \tag{49}$$

$$y(t) = C(t)x(t) + D(t)v(t), \tag{50}$$

where $x(t)$ is the system state at time step t, $y(t)$ is the measurement output to be transmitted over a communication network, $w(t)$ is the system disturbance, $v(t)$ is the measurement noise, $x(0) = x_0$ is the initial state, and $A(t), B(t), C(t)$ and $D(t)$ are real-valued time-varying matrices with appropriate dimensions. Within a set-membership state estimation framework, without requiring accurate statistical knowledge, the disturbance and noise signals are confined to some specified ellipsoids of the following form:

$$\mathcal{W}_t \triangleq \{w(t) : w^{\mathrm{T}}(t)Q^{-1}(t)w(t) \leqslant 1\}, \tag{51}$$

$$\mathcal{V}_t \triangleq \{v(t) : v^{\mathrm{T}}(t)R^{-1}(t)v(t) \leqslant 1\}, \tag{52}$$

where $Q(t) = Q^{\mathrm{T}}(t) > 0$ and $R(t) = R^{\mathrm{T}}(t) > 0$ are real-valued time-varying matrices with appropriate dimensions.

The state estimator has the following form:

$$\hat{x}(t+1) = G(t)\hat{x}(t) + L(t)y(t_k), \tag{53}$$

where $\hat{x}(t)$ is the state estimate of the true system state at time k, $G(t)$ and $L(t)$ are the estimator gain matrix sequences to be designed, $\hat{x}(0) = \hat{x}_0$ is the initial state estimate, and $y(t_k)$ is the received system measurement output at time step t_k calculated recursively by the following dynamic triggering mechanism:

$$t_{k+1} = \inf_{t>t_k} \{\theta l(t) > \delta(t),\ t \in \mathbb{N}\}, \tag{54}$$

where $l(t) = (y(t) - y(t_k))^{\mathrm{T}} W(t)(y(t) - y(t_k)) - \varepsilon y^{\mathrm{T}}(t_k)W(t)y(t_k)$ and $\delta(t)$ is an auxiliary offset variable satisfying

$$\delta(t+1) = \rho\delta(t) - l(t), \quad \delta(0) = \delta_0 \geqslant 0 \tag{55}$$

with $0 < \rho < 1$ and $\theta \geqslant 1/\rho$.

To ensure that the true system state can be included in a confidence region, the following performance requirement is pursued at each time step:

$$\mathcal{X}_{t+1} \triangleq \{x(t+1) : e^{\mathrm{T}}(t+1)P^{-1}(t+1)e(t+1) \leqslant 1\}, \tag{56}$$

where $e(t+1) = x(t+1) - \hat{x}(t+1)$ is the estimation error, $P(t+1) = P^{\mathrm{T}}(t+1) > 0$ is a real-valued time-varying matrix to be determined, and $e^{\mathrm{T}}(0)P^{-1}(0)e(0) \leqslant 1$ with $P(0) = P^{\mathrm{T}}(0) > 0$ is a known real-valued matrix. Therefore, it is clear that the system's one-step ahead state $x(t+1)$ will always reside in the confidence state estimation ellipsoid $\mathcal{X}_{t+1}$ regardless of unknown-but-bounded disturbance $\mathcal{W}_t$ and noise $\mathcal{V}_t$.

Owing to the non-negativity of the auxiliary offset variable $\delta(t)$, it is analytically and numerically proved in [51] that the dynamic triggering mechanism (54) can result in larger average inter-event interval and thus less totally release data packets in comparison with some existing static triggering mechanisms of the following form:

$$t_{k+1} = \inf_{t>t_k} \{l(t) > 0,\ t \in \mathbb{N}\}. \tag{57}$$

However, it is noteworthy that the tradeoff between the inter-event intervals and tightness of the ellipsoidal estimation set is not examined. Indeed, how to find tight upper and lower bounds for the ellipsoidal estimation sets remains a challenging issue, which is of particular significance within an event-triggered state estimation context.

4 Some extensions

In this section, some extensions on switching topologies, quantized control, input-based triggering mechanism, denial-of-service (DoS) attacks, and nonlinear systems will be discussed.

4.1 Switching topologies

Most results on MASs usually consider the fixed topology case with the event-triggered control. However, in practical communication environment, the data exchange among agents may not be always node-to-node communication owing to failure or repair of agent itself so that the fixed topology becomes time-varying topologies. An event-triggered controller based on the conventional results on consensus of MASs with switching topologies requires the absolute states that are difficult to obtain in practice [52]. So, there arises an issue that continuous communication is needed to check the triggering function. To deal with this issue, in [53], a mode-based control approach is employed to estimate the state values instead of measuring the absolute states. In particular, a rule between the average dwell time-based switching topologies and the triggering instants is designed to derive the consensus criterion.

4.2 Quantized control

Some constraints usually exist in practical communication networks such as the limited bandwidth, packet loss, and time-delay. It is known that a quantized control approach can well cope with these constraints [54]. An event-triggered quantized controller is developed to investigate the practical leader-following consensus of nonlinear MASs in [55], where a uniform quantizer is adopted to quantize all states of agents. However, quantization of all states of agents may lead to heavy computational cost for each agent. To save the computation, another quantizer is adopted in [56] that quantizes the relative measurements and the sum of the relative measurements. In this paper, adjustable conditions of achieving consensus are derived where all agents converge to a bounded ball and the radius of the ball can be adjustable by choosing different parameters.

4.3 Input-based triggering mechanism

In most event-triggered control approaches, the triggering condition is based on the measurement error. There is another approach where the triggering condition is based on input [57]. The input-based triggering condition associated with a state-dependent threshold is proposed to derive the consensus criterion by estimating the state values of the relative links between agents. Using this approach, every link has its own triggering instants and a lower bound while each agent has a lower bound in the conventional results [58]. In some scenarios, this approach can allocate the limited resource more efficiently and has faster convergence speed. This approach has also been used in the containment control problem that aims to find a controller such that the subset of agents stays in the region consisting the leaders [59].

4.4 Cyber attacks

Data transmission via a shared communication network in networked systems is vulnerable to cyber attacks such as deception attacks [60–62] and denial-of-service (DoS) attacks [63–65]. In [60], the secure consensus problem is considered by using event-triggered communication protocols. In [62], for a positive discrete-time linear system over a sensor network in the presence of deception attacks, the secure ℓ_1-gain performance is analyzed and the distributed finite-time filter is designed. In [65], observer-based event-triggered control for a continuous networked linear system subject to DoS attacks is addressed. A novel event-based switched system model is established by considering the effect of the event-triggering scheme and DoS attacks simultaneously. In [66], an input triggering approach is developed that is able to achieve secure consensus of MASs under DoS attacks within the switching system framework. In particular, the attack frequency and the attack duration introduced in [67] are used to restrict the adversaries and to derive the condition of achieving consensus.

4.5 Nonlinear systems

The research on event-triggered control of nonlinear systems is mainly on the stabilization problem. A typical approach is to use the small gain theorem based on the prerequisite that the nonlinear system with a continuous-time (not event-triggered) controller can be input-to-state stable from the measurement error to state [68]. The approach is also used to deal with partial state feedback and output feedback scenarios in [69]. When integrated with the internal model technique, the event-triggered stabilization technique can also be used to handle the output regulation problem in [70] and the cooperative output regulation problem in [71, 72]. In these studies, the steady-state tracking error is regulated to a prescribed bound and Zeno behavior is excluded. As the ISS property, from the measurement (sensor) error to state, is technically difficult to achieve (note that global internal stabilizability does not imply global external stabilizability for nonlinear systems), a new approach is introduced in [73]. It requires a mild condition that the controlled system has a certain ISS property, but from the input (actuator) disturbance to state. This new technique is able to handle the asymptotic output regulation problem with the steady-state tracking error regulated to zero. The technique in [73] is motivated by the sampled-data technique developed in [74]. The result has also be used to solve the synchronization problem for heterogeneous nonlinear networked systems in [39].

5 Conclusion and some challenging issues

Networked triggering techniques have been extensively studied over the past years in many control and estimation problems. The bulky literature is difficult to digest for researchers, especially with the triggering mechanisms mixed with or even hidden behind the complexity of networked system dynamics and the variety of control and estimation strategies. This study has reviewed some existing triggering techniques in a unified formulation, peeled from system dynamics and control and estimation strategies, thus provided readers with better understanding of triggering mechanisms. In what follows, we point out some challenging issues that are worthy of future research to conclude the study.

- New triggering techniques for complicated system dynamics and control tasks. Most event-triggered controllers and observers are designed for linear networked systems in simple settings. More advanced triggering techniques are demanded when complicated system dynamics are taken into consideration. Also, the consensus problem is one of the main testbeds for networked triggering techniques. Research efforts are potentially required on new triggering techniques for various practical control tasks such as synchronization, formation control, and distributed optimization.
- New triggering techniques in combination with intelligent communication/scheduling protocols. Most existing triggering mechanisms require that data are transmitted in a single packet manner and there is no collision of data transmissions from multiple senders. This is unreasonable in practical distributed and networked applications. The analysis and synthesis of distributed and networked control

systems in the context of event-triggered control and estimation will be more challenging when there are disorders and competition of multiple data packets. How to design intelligent triggering techniques that can trigger and schedule multiple data packets effectively and efficiently deserves deep investigation.

- New triggering techniques in the context of system and network co-design. The majority of existing triggering mechanisms can be deemed as an afterthought from a resource conservation perspective. This is because the design of such a mechanism involves only the state/output of the control system but not the state/dynamics of the communication network. In this sense, communication and network resources are first assumed to be sufficient and then saved to some extent by resorting to the triggering mechanisms. Design of networked triggering techniques, however, should be combined with the dynamics of the communication network based on current network state such as idle or busy status such that a better and more realistic tradeoff can be achieved among the control system performance and network performance.
- Demand for new triggering techniques in engineering applications. As a typical application of networked cyber-physical systems, a smart grid incorporates power grids, information and communication systems, and human societies. In some practical control design of smart grids such as frequency and voltage regulation [75, 76], the fast convergence property of the control schemes is necessarily required in order to ensure reliability and security of smart grid operation. Note that the adoption of event-triggered mechanisms can significantly reduce data transmission over communication networks while probably degrading the convergence speed of the control scheme seriously. Therefore, in the context of smart grids, it remains a challenging issue on how to devise a triggering technique such that a suitable tradeoff between the faster convergence rate and the reduced communication resource utilization can be realized perfectly.
- New triggering techniques in fixed-time cooperative control. Fixed-time cooperative control is currently a hot research topic in networked systems including multi-agent systems [77–79] because it can provide a guaranteed settling time [80], which does not depend on initial conditions. This settling time characterizes convergence rate of the corresponding closed-loop system and can serve as one performance index for controller design. It should be pointed out that in the context of fixed-time cooperative control, one should carefully design an event-triggered mechanism over the fixed-time interval such that Zeno phenomenon should be effectively excluded. Thus, how to well address this issue is an interesting topic.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 61803167, 51729501) and State Key Laboratory of Industrial Control Technology (Grant No. ICT1900344).

References

1 Huggett N. Zeno's paradoxes. The Stanford Encyclopedia of Philosophy, 2010. https://plato.stanford.edu/archives/win2010/entries/paradox-zeno/
2 Chen T W, Francis B A. Optimal Sampled-Data Control Systems. New York: Springer-Verlag, 1995
3 Wang X F, Lemmon M D. Event-triggering in distributed networked control systems. IEEE Trans Automat Contr, 2011, 56: 586–601
4 Dimarogonas D V, Johansson K H. Stability analysis for multi-agent systems using the incidence matrix: quantized communication and formation control. Automatica, 2010, 46: 695–700
5 Zhang X-M, Han Q-L, Yu X H. Survey on recent advances in networked control systems. IEEE Trans Ind Inf, 2016, 12: 1740–1752
6 Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans Automat Contr, 2007, 52: 1680–1685
7 Qin J H, Ma Q C, Shi Y, et al. Recent advances in consensus of multi-agent systems: a brief survey. IEEE Trans Ind Electron, 2017, 64: 4972–4983
8 Peng C, Li F Q. A survey on recent advances in event-triggered communication and control. Inf Sci, 2018, 457–458: 113–125
9 Ge X H, Han Q-L, Zhang X M, et al. Distributed event-triggered estimation over sensor networks: a survey. IEEE Trans Cybern, 2019. doi: 10.1109/TCYB.2019.2917179
10 Nowzari C, Garcia E, Cortés J. Event-triggered communication and control of networked systems for multi-agent consensus. Automatica, 2019, 105: 1–27
11 Dimarogonas D V, Frazzoli E, Johansson K H. Distributed event-triggered control for multi-agent systems. IEEE Trans Automat Contr, 2012, 57: 1291–1297
12 Dimarogonas D V, Johansson K H. Event-triggered control for multi-agent systems. In: Proceedings of the 48th IEEE

Conference on Decision and Control jointly with the 28th Chinese Control Conference, Shanghai, 2009. 7131–7136
13 Dimarogonas D V, Johansson K H. Event-triggered cooperative control. In: Proceedings of the 2009 European Control Conference, Budapest, 2009. 3015–3020
14 Aström K J, Bernhardsson B. Comparison of periodic and event based sampling for first-order stochastic systems. In: Proceedings of the 14th IFAC World Congress, Beijing, 1999. 301–306
15 Borgers D P, Heemels W P M H. Event-separation properties of event-triggered control systems. IEEE Trans Automat Contr, 2014, 59: 2644–2656
16 Garcia E, Cao Y C, Yu H, et al. Decentralised event-triggered cooperative control with limited communication. Int J Control, 2013, 86: 1479–1488
17 Zhu W, Jiang Z-P, Feng G. Event-based consensus of multi-agent systems with general linear models. Automatica, 2014, 50: 552–558
18 Xing L T, Wen C Y, Guo F H, et al. Event-based consensus for linear multiagent systems without continuous communication. IEEE Trans Cybern, 2017, 47: 2132–2142
19 Fan Y, Feng G, Wang Y, et al. Distributed event-triggered control of multi-agent systems with combinational measurements. Automatica, 2013, 49: 671–675
20 Sun Z Y, Huang N, Anderson B D O, et al. Comments on "distributed event-triggered control of multi-agent systems with combinational measurements". Automatica, 2018, 92: 264–265
21 Hu W F, Liu L, Feng G. Consensus of linear multi-agent systems by distributed event-triggered strategy. IEEE Trans Cybern, 2016, 46: 148–157
22 Du C K, Liu X D, Liu H K, et al. Distributed event-triggered consensus control for general linear multi-agent systems under directed graph. In: Proceedings of the 56th IEEE Conference on Decision and Control, Melbourne, 2017. 4550–4555
23 Liu Z X, Chen Z Q, Yuan Z Z. Event-triggered average-consensus of multi-agent systems with weighted and direct topology. J Syst Sci Complex, 2012, 25: 845–855
24 Nowzari C, Cortés J. Distributed event-triggered coordination for average consensus on weight-balanced digraphs. Automatica, 2016, 68: 237–244
25 Seyboth G S, Dimarogonas D V, Johansson K H. Event-based broadcasting for multi-agent average consensus. Automatica, 2013, 49: 245–252
26 Meng X Y, Xie L H, Soh Y C. Asynchronous periodic event-triggered consensus for multi-agent systems. Automatica, 2017, 84: 214–220
27 Girard A. Dynamic triggering mechanisms for event-triggered control. IEEE Trans Automat Contr, 2015, 60: 1992–1997
28 Sun Z, Huang N, Anderson B D, et al. A new distributed zeno-free event-triggered algorithm for multi-agent consensus. In: Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas, 2016. 3444–3449
29 Meng X Y, Chen T W. Event based agreement protocols for multi-agent networks. Automatica, 2013, 49: 2125–2132
30 Guo G, Ding L, Han Q-L. A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems. Automatica, 2014, 50: 1489–1496
31 Yue D, Tian E G, Han Q-L. A delay system method for designing event-triggered controllers of networked control systems. IEEE Trans Automat Contr, 2013, 58: 475–481
32 Zhang X M, Han Q-L, Zhang B L. An overview and deep investigation on sampled-data-based event-triggered control and filtering for networked systems. IEEE Trans Ind Inf, 2017, 13: 4–16
33 Ding L, Han Q-L, Ge X H, et al. An overview of recent advances in event-triggered consensus of multiagent systems. IEEE Trans Cybern, 2018, 48: 1110–1123
34 Mazo M, Tabuada P. Decentralized event-triggered control over wireless sensor/actuator networks. IEEE Trans Automat Contr, 2011, 56: 2456–2461
35 Tallapragada P, Chopra N. Decentralized event-triggering for control of nonlinear systems. IEEE Trans Automat Contr, 2014, 59: 3312–3324
36 Fan Y, Liu L, Feng G, et al. Self-triggered consensus for multi-agent systems with zeno-free triggers. IEEE Trans Automat Contr, 2015, 60: 2779–2784
37 Hu W F, Liu L, Feng G. Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy. IEEE Trans Cybern, 2017, 47: 1914–1924
38 Xu W Y, Ho D W C, Li L L, et al. Event-triggered schemes on leader-following consensus of general linear multiagent systems under different topologies. IEEE Trans Cybern, 2017, 47: 212–223
39 Khan G D, Chen Z Y, Yan Y M. Event-triggered output synchronization of heterogeneous nonlinear multi-agents. 2019. ArXiv: 1908.07803
40 Sijs J, Lazar M. Event based state estimation with time synchronous updates. IEEE Trans Automat Contr, 2012, 57: 2650–2655
41 Shi D W, Chen T W, Shi L. Event-triggered maximum likelihood state estimation. Automatica, 2014, 50: 247–254
42 Han D, Mo Y L, Wu J F, et al. Stochastic event-triggered sensor schedule for remote state estimation. IEEE Trans

Automat Contr, 2015, 60: 2661–2675

43 Hu S L, Yue D. Event-based H_∞ filtering for networked system with communication delay. Signal Processing, 2012, 92: 2029–2039

44 Zhang X M, Han Q-L. Event-based H_∞ filtering for sampled-data systems. Automatica, 2015, 51: 55–69

45 Zhang X-M, Han Q-L, Ge X H, et al. Networked control systems: a survey of trends and techniques. IEEE/CAA J Autom Sin, 2019. doi: 10.1109/JAS.2019.1911651

46 Ge X H, Han Q-L, Wang Z D. A threshold-parameter-dependent approach to designing distributed event-triggered H_∞ consensus filters over sensor networks. IEEE Trans Cybern, 2019, 49: 1148–1159

47 Wang J, Zhang X M, Han Q-L. Event-triggered generalized dissipativity filtering for neural networks with time-varying delays. IEEE Trans Neural Netw Learning Syst, 2016, 27: 77–88

48 Zhang X M, Han Q-L. A decentralized event-triggered dissipative control scheme for systems with multiple sensors to sample the system outputs. IEEE Trans Cybern, 2016, 46: 2745–2757

49 El Ghaoui L, Calafiore G. Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. IEEE Trans Automat Contr, 2001, 46: 1084–1089

50 Calafiore G. Reliable localization using set-valued nonlinear filters. IEEE Trans Syst Man Cybern A, 2005, 35: 189–197

51 Ge X H, Han Q-L, Wang Z D. A dynamic event-triggered transmission scheme for distributed set-membership estimation over wireless sensor networks. IEEE Trans Cybern, 2019, 49: 171–183

52 Li H Q, Liao X F, Huang T W, et al. Event-triggering sampling based leader-following consensus in second-order multi-agent systems. IEEE Trans Automat Contr, 2015, 60: 1998–2003

53 Wu Z-G, Xu Y, Lu R Q, et al. Event-triggered control for consensus of multiagent systems with fixed/switching topologies. IEEE Trans Syst Man Cybern Syst, 2018, 48: 1736–1746

54 Fu M Y, Xie L H. The sector bound approach to quantized feedback control. IEEE Trans Automat Contr, 2005, 50: 1698–1711

55 Wu Z G, Xu Y, Pan Y J, et al. Event-triggered pinning control for consensus of multiagent systems with quantized information. IEEE Trans Syst Man Cybern Syst, 2018, 48: 1929–1938

56 Wu Z G, Xu Y, Pan Y J, et al. Event-triggered control for consensus problem in multi-agent systems with quantized relative state measurements and external disturbance. IEEE Trans Circuits Syst I, 2018, 65: 2232–2242

57 Wu Y Q, Meng X Y, Xie L H, et al. An input-based triggering approach to leader-following problems. Automatica, 2017, 75: 221–228

58 Garcia E, Cao Y C, Wang X F, et al. Decentralized event-triggered consensus of linear multi-agent systems under directed graphs. In: Proceedings of the 2015 American Control Conference, Chicago, 2015. 5764–5769

59 Xu Y, Fang M, Shi P, et al. Multileader multiagent systems containment control with event-triggering. IEEE Trans Syst Man Cybern Syst, 2019. doi: 10.1109/TSMC.2019.2899967

60 He W L, Gao X Y, Zhong W M, et al. Secure impulsive synchronization control of multi-agent systems under deception attacks. Inf Sci, 2018, 459: 354–368

61 Ding D R, Wang Z D, Ho D W C, et al. Observer-based event-triggering consensus control for multiagent systems with lossy sensors and cyber-attacks. IEEE Trans Cybern, 2017, 47: 1936–1947

62 Xiao S Y, Han Q-L, Ge X H, et al. Secure distributed finite-time filtering for positive systems over sensor networks under deception attacks. IEEE Trans Cybern, 2019. doi: 10.1109/TCYB.2019.2900478

63 de Persis C, Tesi P. Networked control of nonlinear systems under denial-of-service. Syst Control Lett, 2016, 96: 124–131

64 Feng Z, Hu G Q, Wen G H. Distributed consensus tracking for multi-agent systems under two types of attacks. Int J Robust Nonlin Control, 2016, 26: 896–918

65 Hu S L, Yue D, Han Q-L, et al. Observer-based event-triggered control for networked linear systems subject to denial-of-service attacks. IEEE Trans Cybern, 2019. doi: 10.1109/TCYB.2019.2903817

66 Xu Y, Fang M, Wu Z-G, et al. Input-based event-triggering consensus of multiagent systems under denial-of-service attacks. IEEE Trans Syst Man Cybern Syst, 2018. doi: 10.1109/TSMC.2018.2875250

67 de Persis C, Tesi P. Input-to-state stabilizing control under denial-of-service. IEEE Trans Automat Contr, 2015, 60: 2930–2944

68 Liu T F, Jiang Z-P. A small-gain approach to robust event-triggered control of nonlinear systems. IEEE Trans Automat Contr, 2015, 60: 2072–2085

69 Liu T F, Jiang Z-P. Event-based control of nonlinear systems with partial state and output feedback. Automatica, 2015, 53: 10–22

70 Liu W, Huang J. Event-triggered global robust output regulation for a class of nonlinear systems. IEEE Trans Automat Contr, 2017, 62: 5923–5930

71 Liu W, Huang J. Cooperative global robust output regulation for a class of nonlinear multi-agent systems by distributed event-triggered control. Automatica, 2018, 93: 138–148

72 Wang J Q, Sheng A D, Xu D B, et al. Event-based practical output regulation for a class of multiagent nonlinear

systems. IEEE Trans Cybern, 2019, 49: 3689–3698

73 Khan G D, Chen Z Y, Zhu L J. A new approach for event-triggered stabilization and output regulation of nonlinear systems. IEEE Transactions on Automatic Control, 2019. doi: 10.1109/TAC.2019.2949894

74 Chen Z Y, Fujioka H. Performance analysis of nonlinear sampled-data emulated controllers. IEEE Trans Automat Contr, 2014, 59: 2778–2783

75 Ding L, Han Q-L, Wang L Y, et al. Distributed cooperative optimal control of dc microgrids with communication delays. IEEE Trans Ind Inf, 2018, 14: 3924–3935

76 Ding L, Han Q-L, Zhang X M. Distributed secondary control for active power sharing and frequency regulation in islanded microgrids using an event-triggered communication mechanism. IEEE Trans Ind Inf, 2019, 15: 3910–3922

77 Zuo Z Y, Han Q-L, Ning B D, et al. An overview of recent advances in fixed-time cooperative control of multiagent systems. IEEE Trans Ind Inf, 2018, 14: 2322–2334

78 Ning B D, Han Q-L. Prescribed finite-time consensus tracking for multiagent systems with nonholonomic chained-form dynamics. IEEE Trans Automat Contr, 2019, 64: 1686–1693

79 Ning B D, Han Q-L, Zuo Z Y. Distributed optimization for multiagent systems: an edge-based fixed-time consensus approach. IEEE Trans Cybern, 2019, 49: 122–132

80 Parsegov S E, Polyakov A E, Shcherbakov P S. Fixed-time consensus algorithm for multi-agent systems with integrator dynamics. In: Proceedings of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems, Rhine-Moselle-Hall, Koblenz, 2013. 110–115