



# 基于事件驱动机制的多智能体系统协调控制研究综述

## 摘要

为了减少对网络带宽、通信成本及智能体自身能量等有限资源的不必要的浪费,近些年来,基于事件驱动机制的多智能体系统协调控制的问题受到大量学者的关注,成为了国内外研究的热点.本文集中讨论了几类典型的事件驱动机制,着重介绍了集中式、聚类式、分布式事件驱动机制,自驱动机制和基于边信息的分布式事件驱动机制的设计思想及研究进展.此外,排除芝诺行为作为事件驱动机制研究的关键性问题之一,本文指出了其早期研究存在的盲点,并总结了两类有效的分析方法.最后,给出了事件驱动机制研究亟待解决的问题,并对其理论和应用研究进行了展望.

## 关键词

多智能体系统;事件驱动机制;芝诺行为

中图分类号 TP13;TP18

文献标志码 A

收稿日期 2018-06-01

资助项目 国家自然科学基金(61573096)

## 作者简介

许文盈,女,博士,讲师,主要研究方向为多智能体系统的协调控制.wenyngxuwinnie@gmail.com

曹进德(通信作者),男,博士,教授,欧洲科学院院士、欧洲科学与艺术院院士、巴基斯坦科学院院士、IEEE Fellow,主要研究方向为复杂网络动力学分析.jdcao@seu.edu.cn

<sup>1</sup> 东南大学 数学学院,南京,211189

## 0 引言

近年来,随着计算机技术日益成熟和人工智能的高速发展,传统的集中式控制已经难以胜任科学技术发展的需要,并行式计算和分布式协调控制技术应运而生.多智能体系统分布式协调控制更因其在无人机编队、多机械臂系统配置、卫星姿态控制、分布式传感器网络等领域的广泛应用,而受到国内外众多学者的关注<sup>[1]</sup>.多智能体系统通常由多个具有独立感知、通信、计算和决策能力的智能体组成,每个智能体具有相对独立的性能指标和行为准则<sup>[2]</sup>.由于单个智能体的能力有限,整个多智能体系统控制和执行过程是由多个智能体之间通过信息交互的方式协调合作完成的.因此分布式协调控制具有更高的鲁棒性、扩展性、容错性和自适应性,使其研究在多个领域得到了迅速发展.多智能体系统协调控制是指多个智能体通过网络结构拓扑进行信息传递与共享以完成共同的目标,这意味着信息传输与共享是影响整个多智能体系统协调合作控制过程至关重要的因素之一.由于网络带宽、通信成本及多智能体自身能量都是有限的,因此需要合理地设计信息传输与共享机制来保证智能体间的协调合作.众所周知,传统的周期采样机制在一定程度上降低了通信频率,节省了对各项资源的需求,但从资源利用的角度来看,基于该机制的通信和控制策略通常具有保守性,尤其是当系统在相对理想的状况下运行时或者系统状态逐渐趋于一致时,依旧以同一固定周期进行采样将会造成通信资源的浪费.因此事件驱动机制被引入了多智能体系统的通信和协调控制中<sup>[3]</sup>.与周期采样不同,事件驱动机制是一种基于“需要”驱动的行为,即仅当某一特定事件发生(如状态量误差超过规定的阈值)时才执行一次操作(如信息传输、控制协议更新).事件触发机制的优点在于它既能保证系统的性能又能节约网络或计算资源.

近些年,伴随着智慧交通、智慧城市建设需要的不断提高,事件驱动的思想被运用到各种智能算法中,例如文献[4]提出一类基于事件驱动的城市交通信号控制方法.相比于时间驱动控制而言,基于事件驱动的控制策略显得更加智慧和灵活.另外,基于事件驱动机制的无线传感器网络被有效地应用于灾难预警等实际问题中,它仅在预设事件发生时才向基站发送报警消息,而非周期性汇报.在多智能体系统中,事件驱动通信机制是指当单个智能体的某一状态变化量超过了给定的阈值时,才与其邻居个体传递信息.因此事件驱动机制的

引入极大程度地减少了信息传输的频率,同时也可以节约网络资源、延长网络生存期,从而使得网络资源得到更加合理的配置;同时,它也降低了智能体自身能量的消耗,延长了智能体的使用寿命.近些年来,基于事件驱动机制的多智能体系统协调控制的研究引起了大量学者的研究兴趣,成为了国内外研究的热点.

伴随着事件驱动机制展现出来的优良特性及应用前景,越来越多的学者纷纷投身到事件驱动机制的研究领域中.近些年,基于事件驱动机制的多智能体系统协调控制研究已经取得了较多有意义的成果.本文将集中讨论两个重要问题:1)设计有效的事件驱动机制;2)提出排除芝诺行为的方法.

### 1 多智能体系统的事件驱动机制设计问题

事件驱动的概念较先被引入到单个系统控制问题中<sup>[3,5]</sup>.文献[5]提出了基于事件驱动的控制策略,该控制策略可以有效地降低控制器更新频率,并应用到喷射式发动机压缩器的控制中.随后,针对各类控制系统的需求,不同事件驱动控制策略被相继提出,如文献[6-8]等.

在多智能体系统中,多个智能体间信息交互是实现协调控制的关键,同时,分布式控制协议的设计也依赖于局部信息交互.然而有限的网络资源和通信带宽在很大程度上限制了智能体间的信息传输.为了缓解该限制,基于事件驱动的通信机制和协调控制策略被提出,它可以有效地降低智能体间信息传输和智能体控制协议更新的频率.

考虑一类线性的多智能体系统,其动力学如下:

$$\dot{x}_i(t) = \mathbf{A} x_i(t) + \mathbf{B} u_i(t), \quad (1)$$

其中  $x_i(t)$  是第  $i$  个智能体在  $t$  时刻的状态,  $u_i(t)$  表示其控制协议,  $i = 1, 2, \dots, N$ ,  $\mathbf{A}$  和  $\mathbf{B}$  是常矩阵.通常,基于事件驱动的控制协议具有如下形式:

$$u_i(t) = \sum_{j \in N_i} (\hat{x}_j(t) - \hat{x}_i(t)). \quad (2)$$

区别于基于时间驱动的控制协议中所运用的  $x_i(t)$  和  $x_j(t)$ , 这里  $u_i(t)$  中所用的是  $\hat{x}_i(t)$  和  $\hat{x}_j(t)$  的信息.常见的  $\hat{x}_i(t)$  有两种形式:

- 1)  $\hat{x}_i(t) = x_i(t_k^i)$ ;
- 2)  $\hat{x}_i(t) = \exp\{\mathbf{A}(t - t_k^i)\} x_i(t_k^i)$ ,

这里,  $t \in [t_k^i, t_{k+1}^i)$ ,  $t_k^i$  是智能体  $i$  的第  $k$  次事件触发时刻.

接下来,一个关键性问题:如何确定每个智能体

的事件触发时刻,使其仅在事件触发时刻与其邻居进行信息交互就能保证最终协调合作的目标完成?

#### 1.1 集中式事件驱动机制

在多智能体系统中,集中式事件驱动机制较早被提出.这里的“集中式”指的是所有智能体共享同一个驱动机制,即它们的触发时刻是一致的,我们可将其表示为  $t_k^1 = t_k^2 = \dots = t_k^N = t_k$ .注意到,每个智能体的协调控制协议  $u_i(t)$  仍是分布式的,即仅与自身及其邻居信息有关.为了进一步确定事件触发时间序列,不同类型的驱动条件相继被提出.

文献[9]提出一类集中式事件驱动条件如下:

$$t_{k+1} = \max\left\{t > t_k \mid \|\mathbf{e}\| \leq \sigma \frac{\|\mathbf{L}\mathbf{x}\|}{\|\mathbf{L}\|}\right\}, \quad (3)$$

其中  $\mathbf{x} = [x_1, \dots, x_N]^T$ ,  $\mathbf{e} = [e_1, e_2, \dots, e_N]^T$ ,  $e_i(t) = x_i(t_k) - x_i(t)$  是第  $i$  个智能体状态的误差函数,  $t_k$  是第  $k$  个事件触发时刻,  $\sigma > 0$ .在此基础上,文献[10]进一步针对切换网络拓扑设计了集中式事件驱动条件,深入分析了网络结构变化对事件驱动机制的影响,并将事件驱动控制策略下网络拓扑结构的要求弱化到了联合连通.更多的集中式事件驱动机制的研究可见文献[11-12].注意到集中式事件驱动机制的设计通常包含所有智能体的信息,例如式(3)中涉及  $\mathbf{x}$  和  $\mathbf{e}$  的信息.该类事件驱动机制设计思想相对简单,通常适用于规模较小的网络化系统.

#### 1.2 聚类式事件驱动机制

伴随网络化系统规模不断增加,设计分散式事件驱动机制的需要被提出.文献[10]提出了聚类式事件驱动机制的概念.首先,根据预先设定的分类规则,将  $N$  个智能体分入  $m$  类中,基于分类规则,一个大规模的网络结构就被划分成若干个小网络的集成体.聚类式事件驱动机制是指处于同类的多个智能体共享同一个事件驱动机制,而处于不同类的多个智能体其事件驱动机制不同.因此,  $m$  个被分割的小多智能体系统就需要设计  $m$  个不同的驱动条件,每个驱动条件仅涉及该类中智能体的信息,其他类中智能体信息均不可获知.

例如,文献[10]根据跟踪智能体(followers)交互结构的连通性将其分成  $m$  类,针对不同类别,设计了  $m$  个驱动条件.处于第  $c$  类智能体的驱动条件设计如下:

$$t_{k+1}^c = \inf\{t > t_k^c : f(\mathbf{e}_c(t)) > g(\mathbf{x}_c(t), \mathbf{e}_c(t))\}, \quad (4)$$

其中  $t_k^c$  是第  $c$  类智能体的第  $k$  个事件触发时刻,处于第  $c$  类的智能体编号为  $c_1, c_2, \dots, c_{l_c}$ ,

$\mathbf{x}_c = [x_{c_1}, x_{c_1}, \dots, x_{c_{l_c}}]^\top$ ,  $\mathbf{e}_c = [e_{c_1}, e_{c_2}, \dots, e_{c_{l_c}}]^\top$ ,  $f(\cdot)$  和  $g(\cdot)$  是关于变量  $\mathbf{e}_c$  和  $\mathbf{x}_c$  的函数. 显然, 机制 (4) 仅涉及处于第  $c$  类的智能体的信息.

另外, 在文献 [10] 工作的基础上, 文献 [13] 进一步分析了聚类式事件驱动机制的有效性. 区别于文献 [10], 文献 [13] 讨论了基于输出信息的聚类式事件驱动机制, 并分析了如何在扰动介入的情况下实现  $L_\infty$  一致性. 同时, 文献 [14] 提出了一类基于层信息的事件驱动机制, 并将其扩展到有向的网络结构中. 相较于集中式事件驱动机制, 聚类式事件驱动机制可适用于较大规模的网络结构.

### 1.3 分布式事件驱动机制

为了适用于大规模的网络环境, 同时提高事件驱动机制的鲁棒性、扩展性和容错性, 分布式事件驱动机制被进一步提出. 分布式事件驱动机制是指每个智能体都有自己的驱动条件, 其仅与该智能体及其邻居信息有关. 几类常见的分布式事件驱动机制如下:

1) 文献 [9] 提出一类分布式事件驱动机制:

$$t_{k+1}^i = \inf \left\{ t > t_k^i : \left\| \mathbf{e}_i(t) \right\|^2 > \frac{\sigma_i a (1-a |N_i|)}{|N_i|} \left\| \mathbf{z}_i(t) \right\|^2 \right\}, \quad (5)$$

其中  $\mathbf{z}_i(t) = \sum_{j \in N_i} (x_i(t) - x_j(t))$ ,  $\mathbf{e}_i(t) = x_i(t_k^i) - x_i(t)$ ,  $t_k^i$  是智能体  $i$  的第  $k$  个驱动时刻. 此类分布式事件驱动机制的目标是减少控制器更新频率. 注意到驱动条件 (5) 中  $\mathbf{z}_i(t)$  包含了智能体  $i$  邻居的当前信息, 这意味着该驱动机制无法降低通信频率. 类似的事件驱动机制在文献 [10, 15-22] 中得到了进一步推广研究.

2) 为了同时降低通信频率和控制器更新次数, 分布式事件驱动机制得到进一步改进. 试图用已有的邻居智能体的信息  $\hat{x}_j(t)$  代替其当前信息  $x_j(t)$  来设计驱动机制, 因此文献 [23] 提出如下机制:

$$t_{k+1}^i = \inf \left\{ t > t_k^i : \mathbf{e}_i^2 > \frac{\sigma_i}{4 d_i^{\text{out}}} \sum_{j \in N_i^{\text{out}}} w_{ij} (\hat{x}_i - \hat{x}_j)^2 \right\}, \quad (6)$$

其中  $\hat{x}_i(t)$  是智能体  $i$  最近一次传输的信息,  $\mathbf{e}_i(t) = \hat{x}_i(t) - x_i(t)$ . 不等式 (6) 中仅含  $\hat{x}_i(t)$  的信息, 与驱动机制 (5) 相比较而言, 驱动机制 (6) 在降低控制器更新频率的同时进一步减少了通信成本. 类似性质的分布式事件驱动机制的研究见文献 [24-26].

相较于集中式、聚类式事件驱动机制, 分布式事件驱动机制具有更强的鲁棒性、扩展性和容错性. 在分布式事件驱动机制下, 每个智能体都单独进行事件监测, 这有可能产生较多的检测成本. 为了进一步减少检测成本, 一类自驱动机制的概念被提出.

### 1.4 自驱动机制

事件驱动机制的执行过程需要不断地监测预设事件, 一旦事件被触发, 便执行相应的行为 (如信息传输、控制器更新等). 该过程需要有特定的设备执行不断检测事件的任务, 但未必所有情况下都能找到特定的设备执行该任务 (见文献 [27]). 文献 [9, 27-28] 提出了自驱动机制, 它是指下一次驱动时刻可以根据当前驱动时刻的信息计算出来, 不需要不断监测事件的变化. 由于下一次驱动时刻通过计算获知, 因此自驱动机制增加了额外的计算成本, 但它节约了事件监测的成本. 通常情况下, 自驱动机制具有如下形式:

$$t_{k+1}^i = t_k^i + \tau_k^i,$$

其中  $\tau_k^i > 0$  是智能体  $i$  在  $t_k^i$  时刻计算出来的. 显然, 如何计算  $\tau_k^i$  是自驱动机制的关键. 针对不同多智能体系统的特点, 各类有效的自驱动机制被相继提出. 例如, 文献 [9] 较早就一阶多智能体系统提出了自驱动机制的研究框架. 随后, 文献 [17] 进一步在线性多智能体系统中提出了一类自驱动机制, 并分析了其有效性. 最近, 文献 [29] 针对离散二阶多智能体系统设计了有效的自驱动机制, 并实现了一致性行为.

### 1.5 基于边信息的分布式事件驱动机制

除了以上介绍的 4 类驱动机制外, 分布式事件驱动机制还根据驱动条件设计对象的不同分成基于点信息的分布式驱动机制和基于边信息的分布式驱动机制. 基于点信息的分布式驱动机制的设计对象是每个智能体, 即针对智能体的行为设计驱动条件, 当驱动条件被触发时, 智能体同时向其所有邻居智能体发送信息, 例如上述介绍的式 (5)、式 (6). 基于边的分布式驱动机制的设计对象是每一条链接边, 即针对链接边设计驱动条件. 如果某链接边的驱动条件被触发, 则其连接的两个智能体交换信息. 假设  $N$  个智能体中包含  $m$  条链接边, 根据智能体间耦合情况可以将其分成  $m$  对, 每条链接边对应一对 (两个) 智能体. 例如第  $s$  条边连接智能体  $i$  和  $j$ , 针对第  $s$  条边设计其驱动条件, 如果该驱动条件被触发, 则智能体  $i$  和  $j$  间相互交换信息. 注意到, 如果仅第  $s$  条边的驱动条件被触发, 这时即使智能体  $i$  还有除智能体

$j$ 外的其他邻居智能体,智能体  $i$  也只与智能体  $j$  交换信息.基于边信息的分布式事件驱动机制较早文献[30-31]中被提出,随后在文献[32-34]中得到进一步研究.文献[33]引入了窗口(buffer)的概念进一步诠释了基于边信息的分布式事件驱动机制的设计理念,并将其扩展到基于边量化信息的驱动机制的研究中.文献[32,34]在二阶多智能体系统中针对有、无时滞两种情况分别提出了有效的基于边信息的驱动条件.

## 2 有效排除芝诺行为(Zeno behavior)的方法

如果事件在有限时间内被触发无限多次,该现象被称为芝诺行为.在驱动机制的研究中,一个关键的任务就是排除芝诺行为.早期在关于排除芝诺行为的研究方法中存在一个误区,即只证明了  $t_{k+1}^i - t_k^i > 0$ .事实上,该结论并不能充分说明芝诺行为已排除.文献[35]举了一个反例:  $t_{k+1}^i - t_k^i = \frac{1}{k^2} > 0$  满足

$t_{k+1}^i - t_k^i > 0$  这一性质,但  $\sum_{k=1}^{\infty} t_{k+1}^i - t_k^i < 2$ , 这意味在时间区间  $[0, 2]$  内仍有无穷多次事件被触发.这个例子说明了  $t_{k+1}^i - t_k^i > 0$  不足以排除芝诺行为.那么该如何排除芝诺行为呢? 通常有以下两种方法.

### 2.1 证明以任意两次触发时刻为端点的区间段其区间长度必存在某一固定正下界

上文已说明  $t_{k+1}^i - t_k^i > 0$  不足以证明芝诺行为已排除.进一步,如果存在某一固定常数  $\varepsilon > 0$  使得  $t_{k+1}^i - t_k^i \geq \varepsilon, \forall k \in \mathbf{N}^+$ , 则有限时间段内必只存在有限个事件被触发,因此具有该性质的事件必定不会出现芝诺行为.

几类典型的事件驱动条件必须具备该性质  $t_{k+1}^i - t_k^i \geq \varepsilon$ . 例如文献[36]提出的  $t_{k+1}^i = \inf\{t > t_k | h(e(t), \tilde{e}(t), \tilde{\eta}(t)) \geq \delta\}$ , 具有该性质的主要因素是  $\delta > 0$  是固定常数.该类型的驱动条件在很多情况下只能实现有界收敛,而无法实现渐近收敛.文献[25]提出结合周期采样的事件驱动机制,即预先设定固定采样周期为  $h$  的采样点,仅在采样点进行事件检测,因此事件触发时刻仅发生在采样点上,因此该驱动机制保证了  $t_{k+1}^i - t_k^i \geq h$ . 此类型的事件驱动机制有效性在文献[37-39]中得到了进一步的验证.随后,文献[17]提出了一类非周期间歇检测的事件驱动条件,即  $t_{k+1}^i = t_k^i + \max\{\tau_k^i, b_i\}$ , 这里  $\tau_k^i$

是由驱动条件决定的,但  $b_i > 0$  是固定常数.这就保证了  $t_{k+1}^i - t_k^i \geq b_i$ , 因此芝诺行为被排除.

### 2.2 反证法证明芝诺行为不存在

另一类证明芝诺行为不存在的方法就是反证法.假设芝诺行为存在,则这意味着时间序列  $\{t_k^i\}_{k \in \mathbf{N}^+}$  必存在聚点,即  $\lim_{k \rightarrow \infty} t_k^i = T_0, T_0$  是常数.随后进一步证明该结论与已有性质矛盾,从而证明芝诺行为不存在.文献[19,40]运用反证法的思想证明了芝诺行为不存在.

到目前为止,有效排除芝诺行为的方法还很少,已有的一些方法中也常卷入部分全局信息,如上述采样周期  $h$  和  $b_i > 0$  的设计都需要知道网络结构对应的 Laplace 矩阵其特征根的信息.因此,如何在事件驱动机制中有效地排除芝诺行为仍有待进一步深入研究.

## 3 总结与展望

由于网络带宽、通信成本、智能体自身能量通常是有限的,因此近年来基于事件驱动机制的通信和协调控制研究得到了各国学者的广泛关注,并取得丰硕的研究成果.相对于传统周期采样,事件驱动机制的优点在于它既能保证系统的性能又能节约网络或计算资源.然而如何设计合理有效的驱动机制,提出排除芝诺行为行之有效的方法是研究事件驱动机制的关键.分布式事件驱动机制作为一个新兴的研究领域,其研究仍处于发展阶段,有以下问题尚未得到很好的解决:

1) 如何设计完全分布式的事件驱动机制,即驱动机制的设计中不卷入任何全局的信息(如智能体总个数、Laplace 矩阵等),这就对事件驱动机制的设计提出了更高的要求,尤其针对广义线性、非线性多智能体系统.

2) 如何针对切换网络结构设计有效的分布式事件驱动机制.注意到网络结构的切换会对分布式驱动条件的触发产生很大的影响,因此探讨网络结构变化和分布式驱动条件两者之间的内在关系是设计有效驱动机制的关键.

3) 目前为止排除芝诺行为行之有效的方法还很少,其理论分析与方法的普适性还有待深入研究.

4) 事件驱动机制完整的理论框架和评价体系有待进一步完善.

总的来说,基于事件驱动机制的多智能体系统协调控制的研究刚刚起步,远不止本文所讨论的几

个方面,许多重要的理论与实际的问题亟待解决.伴随智慧城市建设进程的加快,人工智能、物联网等技术的持续突破与普及,这必然会极大地拓展事件驱动机制的应用范畴,促使事件驱动控制理论研究不断取得新的成果.

## 参考文献

### References

- [ 1 ] 王晓丽,洪奕光.多智能体系统分布式控制的研究新进展[J].复杂系统与复杂科学,2010,7(2/3):70-81  
WANG Xiaoli, HONG Yiguang. New progress in the study of distributed control of multi-agent system [J]. Complex System and Complexity Science, 2010, 7(2/3): 70-81
- [ 2 ] 李韬,孟扬,张纪峰.多自主体量化趋同与有限数据率趋同综述[J].自动化学报,2013,39(11):1805-1811  
LI Tao, MENG Yang, ZHANG Jifeng. An overview on quantized consensus and consensus with limited data rate of multi-agent systems [J]. Acta Automatica Sinica, 2013, 39(11): 1805-1811
- [ 3 ] Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks [J]. IEEE Transactions on Automatic Control, 2007, 52(9): 1680-1685
- [ 4 ] 王大珊,王松浩,赵玉娟,等.基于事件驱动的城市交通信号控制方法[J].交通标准化,2014,42(23):151-159  
WANG Dashan, WANG Songhao, ZHAO Yujuan, et al. Urban traffic signal control method based on event scheduling [J]. Transportation Standardization, 2014, 42(23): 151-159
- [ 5 ] Anta A, Tabuada P. To sample or not to sample: self-triggered control for nonlinear systems [J]. IEEE Transactions on Automatic Control, 2010, 55(9): 2030-2042
- [ 6 ] Yue D, Tian E G, Han Q L. A delay system method for designing event-triggered controllers of networked control systems [J]. IEEE Transactions on Automatic Control, 2013, 58(2): 475-481
- [ 7 ] Heemels W P M H, Donkers M C F, Teel A R. Periodic event-triggered control for linear systems [J]. IEEE Transactions on Automatic Control, 2013, 58(4): 847-861
- [ 8 ] Dong H L, Wang Z D, Shen B, et al. Variance-constrained  $H_\infty$  control for a class of nonlinear stochastic discrete time-varying systems: the event-triggered design [J]. Automatica, 2016, 72: 28-36
- [ 9 ] Dimarogonas D V, Frazzoli E, Johansson K H. Distributed event-triggered control for multi-agent systems [J]. IEEE Transactions on Automatic Control, 2012, 57(5): 1291-1297
- [ 10 ] Xu W Y, Ho D W C, Li L L, et al. Event-triggered schemes on leader-following consensus of general linear multi-agent systems under different topologies [J]. IEEE Transactions on Cybernetics, 2015, 47(1): 212-223
- [ 11 ] Xu W Y, Ho D W C. Clustered event-triggered consensus analysis: an impulsive framework [J]. IEEE Transactions on Industrial Electronics, 2016, 63(11): 7133-7143
- [ 12 ] Xu W Y, Chen G R, Ho D W C. A layered event-triggered consensus scheme [J]. IEEE Transactions on Cybernetics, 2016, 47(8): 2334-2340
- [ 13 ] Chen M Z Q, Zhang L Y, Su H S, et al. Event-based synchronization of linear discrete-time dynamical networks [J]. IET Control Theory & Applications, 2015, 9(5): 755-765
- [ 14 ] Zhang H, Feng G, Yan H C, et al. Observer-based output feedback event-triggered control for consensus of multi-agent systems [J]. IEEE Transactions on Industrial Electronics, 2014, 61(9): 4885-4894
- [ 15 ] Fan Y, Feng G, Wang Y, et al. Distributed event-triggered control of multi-agent systems with combinational measurements [J]. Automatica, 2013, 49(2): 671-675
- [ 16 ] Lu W L, Han Y J, Chen T P. Synchronization in networks of linearly coupled dynamical systems via event-triggered diffusions [J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(12): 3060-3069
- [ 17 ] Fan Y, Liu L, Feng G. Self-triggered consensus for multi-agent systems with Zeno-free triggers [J]. IEEE Transactions on Automatic Control, 2015, 60(10): 2779-2784
- [ 18 ] Hu W F, Liu L, Feng G. Cooperative output regulation of linear multi-agent systems by intermittent communication: a unified framework of time-and event-triggering strategies [J]. IEEE Transactions on Automatic Control, 2018, 63(2): 548-555
- [ 19 ] Zhu W, Jiang Z P. Event-based leader-following consensus of multi-agent systems with input time delay [J]. IEEE Transactions on Automatic Control, 2015, 60(5): 1362-1367
- [ 20 ] Zhu W, Jiang Z P, Feng G. Event-based consensus of multi-agent systems with general linear models [J]. Automatica, 2014, 50(2): 552-558
- [ 21 ] Ding D, R Wang Z D, Shen B, et al. Event-triggered consensus control for discrete-time stochastic multi-agent systems: the input-to-state stability in probability [J]. Automatica, 2015, 62(C): 284-291
- [ 22 ] Liu Q Y, Wang Z D, He X, et al. Event-based  $H_\infty$  consensus control of multi-agent systems with relative output feedback: the finite-horizon case [J]. IEEE Transactions on Automatic Control, 2015, 60(9): 2553-2558
- [ 23 ] Nowzari C, Cortés J. Distributed event-triggered coordination for average consensus on weight-balanced digraphs [J]. Automatica, 2016, 68(C): 237-244
- [ 24 ] Li L L, Ho D W C, Xu S Y. A distributed event-triggered scheme for discrete-time multi-agent consensus with communication delays [J]. IET Control Theory & Applications, 2014, 8(10): 830-837
- [ 25 ] Meng X G, Chen T W. Event based agreement protocols for multi-agent networks [J]. Automatica, 2013, 49(7): 2125-2132
- [ 26 ] Xu W Y, Wang Z D, Ho D W C. Finite-horizon H-infinity consensus for multi-agent systems with redundant channels via an observer-type event-triggered scheme [J]. IEEE Transactions on Cybernetics, 2017, 48(5): 1567-1576
- [ 27 ] Heemels W P M H, Johansson K H, Tabuada P. An intro-

- duction to event-triggered and self-triggered control [C] // IEEE 51st Annual Conference on Decision and Control, 2012: 3270-3285
- [28] Anta A, Tabuada P. To sample or not to sample: self-triggered control for nonlinear systems [J]. IEEE Transactions on Automatic Control, 2010, 55(9): 2030-2042
- [29] Zhu W, Pu H Z, Wang D, et al. Event-based consensus of second-order multi-agent systems with discrete time [J]. Automatica, 2017, 79: 78-83
- [30] DePersis C, Frasca P. Robust self-triggered coordination with ternary controllers [J]. IEEE Transactions on Automatic Control, 2013, 58(12): 3024-3038
- [31] Xiao F, Meng X Y, Chen T W. Average sampled-data consensus driven by edge events [C] // Proceedings of the 31th Chinese Control Conference, 2012: 6239-6244
- [32] Cao M T, Xiao F, Wang L. Event-based second-order consensus control for multi-agent systems via synchronous periodic event detection [J]. IEEE Transactions on Automatic Control, 2015, 60(9): 2452-2457
- [33] Xu W Y, Ho D W C, Zhong J, et al. Distributed edge event-triggered consensus protocol of multi-agent systems with communication buffer [J]. International Journal of Robust and Nonlinear Control, 2016, 47(3): 483-496
- [34] Cao M T, Xiao F, Wang L. Second-order leader-following consensus based on time and event hybrid-driven control [J]. Systems & Control Letters, 2014, 74: 90-97
- [35] Xu W Y, Ho D W C, Zhong J, et al. An impulsive framework for consensus learning via event-triggered scheme [C] // 2016 IEEE International Conference on Industrial Technology, 2016: 1366-1371
- [36] Liu W, Huang J. Event-triggered global robust output regulation for a class of nonlinear systems [J]. IEEE Transactions on Automatic Control, 2017, 62(11): 5923-5930
- [37] Xiao F, Chen T W, Gao H J. Synchronous hybrid event- and time-driven consensus in multi-agent networks with time delays [J]. IEEE Transactions on Cybernetics, 2016, 46(5): 1165-1174
- [38] Li H Q, Liao X F, Huang T W, et al. Event-triggering sampling based leader-following consensus in second-order multi-agent systems [J]. IEEE Transactions on Automatic Control, 2015, 60(7): 1998-2003
- [39] Wen G H, Chen M Z Q, Yu X H. Event-triggered master-slave synchronization with sampled-data communication [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2016, 63(3): 304-308
- [40] Yi X L, Liu K, Dimarogonas D V, et al. Distributed dynamic event-triggered control for multi-agent systems [C] // IEEE 56th Annual Conference on Decision and Control, 2017: 6683-6688

## An overview of recent progress in the study of event-triggered coordinated schemes of multi-agent systems

XU Wenyong<sup>1</sup> CAO Jinde<sup>1</sup>

<sup>1</sup> School of Mathematics, Southeast University, Nanjing 211189

**Abstract** In order to reduce the unnecessary waste of limit network resources such as network bandwidth, communication cost, and agents' energy, the investigation of event-triggered schemes has become a hot research topic and attracted numerous scholars' attention. This paper mainly discusses several classical event-triggered schemes, i.e. centralized, clustered and distributed event-triggered schemes, self-triggered schemes and edge event-triggered schemes. The research background and the recent results about the above schemes are specifically introduced. Note that the exclusion of Zeno behavior is a key problem of analyzing the effectiveness of event-triggered schemes. This paper points out a flaw of excluding Zeno behavior in the early related research, and then summarizes two basic methods to avoid this behavior. Finally, this paper provides some future research works of event-triggered schemes, and their theoretical or application research.

**Key words** multi-agent systems; event-triggered schemes; Zeno behavior