T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Dynamic Event-Triggered Consensus in Multi-Agent Systems
著者(和文)	MISHRARajiv Kumar
Author(English)	Rajiv Kumar Mishra
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第11858号, 授与年月日:2022年3月26日, 学位の種別:課程博士, 審査員:石井 秀明,三宅 美博,DEFAGO XAVIER,小野 功,下坂 正倫
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第11858号, Conferred date:2022/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Science Tokyo Research Repository)

Dynamic Event-Triggered Consensus in Multi-Agent Systems

Rajiv Kumar Mishra

Department of Computer Science

Tokyo Institute of Technology

Supervisor

Professor Hideaki Ishii

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

January, 2022

Abstract

Distributed coordinated control for multi-agent systems (MAS) due to widespread applications in many fields has attracted researchers from various areas including control, communications, signal processing, robotics, physics and so on. MAS are comprised of a number of autonomous agents that interact with each other over a shared network to reach a common goal. MAS provide higher redundancy and improved operational efficiency compared with a single autonomous agent. Moreover, this class of dynamical systems are known to exploit the advantages of having distributed sensing and actuation. A canonical problem in cooperative control of MAS is consensus, where the goal is that all agents eventually arrive at a group decision value. To this end, the agents locally communicate with each other in the neighborhood to exchange their state information. Effective distributed coordinated control schemes rely on sufficient information transmission among agents and efficient control protocol design.

This thesis focuses on developing distributed control protocols to effectively reduce transmission and update frequencies for the agents while guaranteeing consensus. We develop triggering-based protocols to facilitate opportunistic inter-agent communication and to alleviate communication and computational burden. We develop a unified framework for these protocols to solve the state consensus problems in the discrete-time domain for MAS with single-integrator dynamics

as well as with general linear dynamics. Our approach for triggering-based protocols is to incorporate state information at each agent to determine the next event times for state updates and communication. This feature helps the agents take account of the consensus level locally and be responsive when they are close to reach their goals. Specifically, we develop two event-triggered protocols where the events occur in real time informing the agents to make updates in their states and broadcasts to transmit their state information to their neighbors. We also develop a self-triggered protocol where each agent estimates its next triggering times and broadcast them to their neighbors at the current triggering times.

The thesis consists of three parts to solve consensus problems of MAS in the discrete-time framework as follows:

(1) We focus on solving the average consensus problem of MAS with single-integrator dynamics. We propose three triggering-based protocols intending to reduce communication and update frequencies for agents while ensuring average consensus. Lyapunov-based design methods prescribe when agents should communicate and update their control so that the network converges to the average of agents' initial states asymptotically. Our general approach is to employ triggering protocols with threshold mechanism that is based on the state values received from neighbors. We begin with a static version of the triggering protocol to effectively reduce transmission and control update requirements for agents. When the agent is far from local consensus with its neighbors the threshold is large and the control need not be very accurate. This helps in reducing triggering instants without sacrificing convergence performance.

To further reduce the number of triggering instants, we employ an auxiliary state variable for each agent to regulate the threshold dynamically. The auxiliary variables take account of the state errors and can reduce the number of triggering instants compared with the static-triggering schemes. Moreover, to avoid continuous monitoring of the states to determine control update instants, we propose the self-triggered protocol where the next control update instant is determined using the current state and is sent together with the state information at the current triggering instant. Each agent only needs to make updates/broadcasts at its own triggering instants and listen to its neighbors at their announced triggering instants. Numerical examples are shown using MAS connected over a random network to illustrate these protocols in terms of reducing the frequencies of communication and control updates.

(2) We develop a unified framework for the triggering-based protocols to solve state consensus problems in MAS having general linear dynamics. We discuss three triggering-based protocols with different characteristics and advantages in terms of necessary computational resources and capabilities in reducing the frequency of communications and updates for each agent. In particular, we propose two event-triggered protocols with state-dependent thresholds. When agents are far from local consensus, less frequent communication suffices without sacrificing convergence performance. We start with a static triggering protocol and then generalize it so that it entails an internal auxiliary state variable to regulate the threshold dynamically for each agent. They are referred to as static and dynamic event-triggered protocols, respectively. The third protocol employs self-triggered control, where

the agents in advance determine and broadcast to their neighbors their next triggering instants together with their state information. We discuss an application example of triggering-based protocol to solve the formation control problem of MAS. Through numerical simulations, we validate the efficacy of the proposed protocols.

(3) We further extend our proposed framework to solve the weighted consensus problems in MAS having integrator dynamics with directed topologies. We generalize the three protocols, namely, the static, dynamic and self-triggering protocols, for this case. We characterize their performances in reducing transmission and update frequencies for the agents. We also examine how our triggering-based protocols perform when adversaries are present and launch denial-of-service (DoS) attacks. Simulation examples are provided to demonstrate the efficiency of the methods and to confirm the theoretical analysis.

Acknowledgements

I am forever indebted to my supervisor Professor Hideaki Ishii for his generous support, advice, encouragement and guidance during my Ph.D. studies. He is the best advisor I could ever imagine. His professionalism and humble personality have set up a role model for me. It is a privilege for me to be a member of Ishii Lab.

I am immensely grateful to all the respected members of my thesis committee: Professor Yoshihiro Miyake, Professor Xavier Defago, Professor Isao Ono and Professor Masamichi Shimosaka. Their pertinent comments and valuable suggestions have greatly helped improve my work.

The entire work of this dissertation has been supported and funded by Monbukagakusho (MEXT scholarship).

I am thankful to my colleagues Shinya Ohmori, Rui Kato and Xile Kang for their support and technical discussions.

I would like to make a special mention and thank Dr. Ahmet Cetinkaya for his support and encouragement.

Last but not least, I would like to express my deepest gratitude to my parents and siblings for their endless love and constant support throughout my career.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The recent technological advancement has significantly changed the way we perceive information from the physical world. More recently, a new class of dynamical systems called cyber-physical systems (CPS) has shown potential application in many fields [47] including automated driving systems [6], telesurgery [58] and smart cities [97]. The fusion of communication, control, computing and information theories has sprung up the concept of CPS. CPS is characterized by the combination of physical components and cyber communication [4]. Typically, network in CPS is not only used to facilitate communication but also to regulate the physical process by the connected actuators (e.g., manipulating headings and speeds of vehicles in platoon of vehicles).

Recently, cyber-physical systems have drawn a significant attention in the area of systems control. In particular, networked control systems (NCS) where sensor/actuator data is transmitted via wireless communication networks have become popular [35]. NCS are characterized by traditional feedback control systems closed through a communication channel possibly shared with other devices

[7], [99]. The use of wireless networks as communication media among the agents has broadened the scope of MAS applications in many fields [39]. The inherent tight coupling requirement between physical processes (sampling, actuation) and cyber processes (communication, computation) in CPS and networked MAS calls for a resource-efficient coordination scheme to reduce communication and computational burden. Event-triggered control schemes are used in our study where the sensor/actuator data is transmitted based on certain conditions on states. This scheme is useful as the communication resources are utilized only when required to maintain the desired closed-loop behavior. In this thesis, we examine different triggering conditions which are well designed to balance between utilization of communication resources and control performance.

1.2 Multi-agent Consensus

Multi-agent systems are intelligent systems composed of several autonomous agents working together to reach certain group objective. Research efforts on MAS have significantly grown due to obvious advantages of higher operational efficiency and redundancy, increased feasibility and reliability and also distributed sensing and actuation capabilities. In recent years, the distributed coordinated control of MAS has received tremendous attention in terms of both theoretic developments and practical applications. Consensus is one of the most fundamental research problem associated with the studies of distributed cooperative control [19]. A consensus algorithm or protocol is an interaction rule that specifies the information exchange between agents and its neighbors on the shared network [71]. We are interested in developing distributed control protocols wherein agents interact locally with its neighbors and exchange their relative state information to reach a group agreement.

The autonomous agents in MAS are often equipped with embedded microprocessors, onboard communication and actuation modules having limited energy supplies. The transmission of sensor/actuator data must be viewed as a global resource because the communication channel is shared with other agents. This calls for resource efficient design approaches that can better balance between resources utilization and control performance. A limitation of time-triggered control is that it requires synchronous updates of the controllers and can be energy consuming for agents driven by batteries. Event-triggered control schemes have emerged as a resource efficient method to alleviate communication and computational burden. This scheme facilitates opportunistic inter agent communication without sacrificing the performance.

1.3 Event-triggered Control

Modern feedback control systems are typically realized on digital platforms [5]. In the traditional time-triggered setting, sampling, control and actuation are clock-driven regardless of whether the new control command is required or not. In this setting, sampling and transmission instants are scheduled at fixed sampling period chosen to ensure stability based on the worst-case scenario. This may result in many redundant transmissions since sampling/transmissions are scheduled independent of the current states of the system. It may not be desirable to continuously monitor the states of a well-behaving system. Instead it may be more efficient to check intermittently and make sure the closed-loop response is acceptable. In the context of networked cyber-physical systems, where the sensor data is sent to the actuator over a wireless channel shared with other devices, the optimal usage of resources is of importance [40]. Frequent communication may lead to communication congestion and overtaxing of communication and

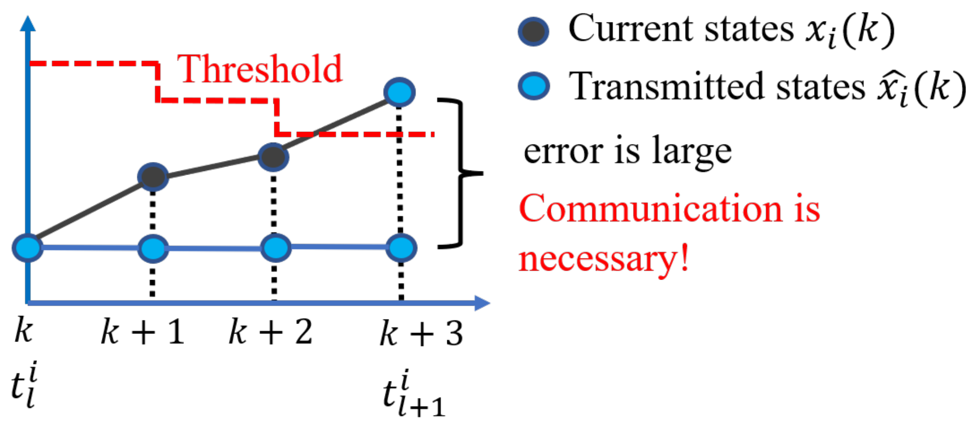

Figure 1.1: Event-triggered control

computational resources.

Event-triggered control systems have emerged as a favorable alternative to traditional time-triggered control to save actuation requirements, computation, and communication resources [38], [44]. The main objective of event-triggering schemes is to update the states of the controllers and also to transmit them only when the local measurement error exceeds a pre-established threshold [72]. Typically, this is done by keeping the last updated/transmitted states and then monitoring their differences from the current states in real-time as shown in Fig. 1.1. The current states $x_i(k)$ and the transmitted states $\hat{x}_i(k)$ are shown in black and blue circles, respectively. As soon as the differences become greater than a predefined threshold shown in dashed red line, the new events at t_{l+1}^i are triggered and transmissions occur. The primary characteristic of this technique is that it enables the system to reduce the frequencies of updates and transmissions of the state values without harming the performance. The importance of this scheme is manifested through several application examples such as battery-operated systems and networked control systems with the shared wireless communication channel.

1.4 Application Examples

In this section we present an overview of some applications related to the consensus problem in MAS with distributed event-triggered algorithms.

1.4.1 Clock Synchronization

In this particular problem of interest, agents synchronize their imperfect clocks via communication. The problem has been well addressed using consensus algorithms with periodic and continuous communications [13], [78]. In particular, we denote the absolute time by $t \in \mathbb{R}_+$, then the local clock time for sensor node i can be given as

$$\tau_i(t) = a_i t + b_i, \tag{1.1}$$

where a_i is the local clock drift and b_i is the local clock offset. We introduce a virtual clock, which is made to synchronize for each agent as

$$T_i(t) = \alpha_i(t)\tau_i(t) + \beta_i(t), \tag{1.2}$$

where $\alpha_i(t)$ and $\beta_i(t)$ are the controlled drift and offset, respectively. The control goal is to obtain consensus on virtual clock as $|T_i(t) - T_j(t)| \to 0$ by all the sensor nodes. The nodes update their drift and offset by local communication. To reduce communication requirement of agents to achieve clock synchronization, event-triggered protocols are presented in the works [30], [45].

1.4.2 Formation Control

Multi-vehicle cooperative control is a popular topic because of having both commercial and military applications. The formation control for networked vehicles which require continuous communication between agents has been well established [74]. Let us denote the position of agent i as $p_i(t) \in \mathbb{R}$. The goal of the formation control is to drive the agents state to $p_i(t) \to \bar{p} + f_i$. Here, $\bar{p}$ is the average position of all agents and f_i corresponds to the fixed formation structure. The agents need to obtain consensus in virtual states $x_i = p_i - f_i$, in order to achieve the desired formation structure. Recently, there has been increased attention on consensus based formation control with event-triggered protocols [1], [18]. Event-triggered consensus based formation control of multi-robot systems has been studied in the work [31]. We will discuss formation control problems of MAS in more detail in Section 4.6.

1.4.3 Power Sharing in Microgrids

Nowadays, due to economic and environmental reasons such as global warming, the research efforts on microgrids have significantly increased. The optimal control of microgrids with MAS based hierarchical hybrid control was presented [24]. A fundamental concern is to economically dispatch power produced by distributed generators to distributed loads. Distributed event-triggered protocols [25], [61] are presented to achieve power sharing in microgrids. Using distributed event-triggered voltage control protocol, the reactive power sharing is obtained in the study [25]. This scheme is effective in reducing communication requirements in microgrids while maintaining similar performance in reactive power sharing with the periodic sampling schemes. We refer to [25] for detailed discussions.

1.5 Contributions of the Thesis

This thesis focuses on developing distributed control protocols to reduce transmission and update frequencies for agents while ensuring consensus. We consider

MAS in the discrete-time domain for practical sampled-data implementation. We develop triggering-based protocols to facilitate opportunistic inter-agent communication and to alleviate communication and computational burden. We develop a unified framework for these triggering based protocols to solve state consensus problem for MAS with single integrator dynamics as well as with general linear dynamics. Continuous communication is not required to monitor triggering conditions and controller updates. In particular, we propose two event triggering protocols with state-dependent thresholds. When agents are far from local consensus with its neighbors the threshold is large and control signals need not change much. This aspect helps the agents take account of the consensus level locally and be responsive when they are close to reach their goals. We start with a static version of the triggering protocol and then generalize it so that it involves an internal auxiliary state variable to regulate the threshold dynamically for each agent. The auxiliary variables take account of the state errors and can reduce the number of triggering instants compared with the static-triggering schemes. The third protocol is based on self-triggering control where the agents in advance determine its next triggering instant and broadcast it to its neighbors together with its state information. We characterize the performances of these protocols in a random network. We further extend our proposed framework to solve the weighted consensus problems in MAS having integrator dynamics with directed topologies. We also examine how our triggering-based protocols perform when adversaries are present and launch denial-of-service (DoS) attacks. Numerical examples are provided to characterize these protocols in terms of reducing the frequencies of communication and control updates for agents.

We would like to discuss the proposed protocols more explicitly as follows:

1. **Protocol 1: Static event triggered**

   We start with a static event-triggered protocol to effectively reduce communications and control updates for agents. The communication for the interaction among neighboring agents occurs opportunistically based on the predefined triggering condition to save energy expenses. Lyapunov-based design methods prescribe when agents should communicate and update their control to ensure asymptotic consensus in their states. The events are triggered as soon as the difference between transmitted states and the current states exceeds a predefined threshold. The threshold depends on the local consensus error, i.e., it depends on the state values received from the neighboring agents. When the agent is far from local consensus the threshold is large and the control need not change much. This helps for reducing triggering instants without sacrificing convergence performance.

2. **Protocol 2: Dynamic event triggered**

   In the static-triggering protocol, the threshold associated with the state error becomes smaller and smaller as the system approaches consensus, which may lead to frequent triggering instants. In order to further reduce the number of triggering instants, we generalize the static-triggering protocol so that it entails an internal auxiliary variable to regulate the threshold dynamically for each agent. The auxiliary variables take account of the state errors and can reduce the number of triggering instants compared with the static-triggering schemes. The static-triggering protocol is a particular case of the dynamic-triggering protocol.

3. **Protocol 3: Self triggered**

   The event-triggered protocols require each agent to continuously monitor its own state and to continuously listen to its neighbors. The third protocol employs self-triggered control, where the agents in advance determine and broadcast to their neighbors their next triggering instants together with their state information. Under this approach, each agent only needs to make updates/broadcasts at its own triggering instants and listen to its neighbors at their announced triggering instants.

1.6 Outline of the Thesis

This thesis is organized as follows:

In Chapter 2, we present an overview of research on distributed control protocols to solve consensus problem in MAS followed by research motivation.

In Chapter 3, we focus on solving the average consensus problem of MAS with single-integrator dynamics. We propose three triggering-based protocols intending to reduce communication and update frequencies for agents while ensuring average consensus. The communication for interaction among neighboring agents occur opportunistically based on certain conditions on the states. Lyapunov-based design methods prescribe when agents should communicate and update their control so that the network converges to the average of agents' initial states asymptotically. We characterize the performances of these protocols over a random network in terms of reducing the frequencies of communication and control updates.

In Chapter 4, we develop a unified framework for the triggering based protocols to solve state consensus problem in MAS having general linear dynamics. We discuss three triggering based protocols with different characteristics and advantages in terms of necessary computational resources and capabilities in reducing the frequency of communications and updates for each agent. We discuss an application example of triggering based protocol to solve the formation control problem of MAS.

In Chapter 5, we further extend our proposed framework to solve the weighted consensus problems in MAS having integrator dynamics with directed topologies. We characterize the advantages of our proposed protocols in terms of reducing communication and control update requirements for agents. We also examine how our triggering-based protocols perform when adversaries are present and launch denial-of-service (DoS) attacks.

Finally, Chapter 6 provides a summary of the results and open research problems. Some potential directions for future research are also given in this chapter.

Chapter 2

Related Research and Motivation

In this chapter current research on distributed control protocols for consensus problem in MAS is discussed, which is followed by motivation for research on reducing communication and control update frequencies for agents.

2.1 Overview of Research on MAS

Observations in nature [8] and intriguing animal behaviors such as schooling of fish, flocking of birds and swarming of bacteria have inspired the development of algorithms for cooperative behaviors of MAS. Reynolds [76] simulated the first flocking behavior. Consensus studies have a rich history in computer science and it arguably forms the foundation of distributed computing [56]. The statistical consensus theory first appeared in the field of management science and statistics [22]. The pioneering work [9] on asymptotic agreement problem has broaden the scope of application of distributed decision making. The theoretical explanation [43] of Vicsek model [86] opened a new direction for development of consensus protocols based on graph theory. The seminal work on networked systems with single-integrator dynamics [71] introduced the concept of algebraic connectivity and Lyapunov based convergence analysis to solve the consensus problem in MAS. Ren et al. presented the protocols for multi-vehicle cooperative control of MAS with double integrator dynamics [74].

With advancement in embedded technology, the control laws for agents are realized on digital platforms. In traditional periodic control, sampling periods are typically chosen under different constraints due to the hardwares in sensors and actuators. This may result in frequent communications and control updates for agents to ensure stability and control performance in the worst-case scenario. The multi-agent networked systems are characterized by a shared communication channel. Event triggered control schemes [36], [60], [73] have emerged as a resource efficient method to reduce communication and computational burden. The sampling and control updates occur based on certain conditions on states, i.e., when state error exceeds a predefined threshold. The advantage of this scheme lies in its ability to reduce transmission and update frequencies for agents without harming the performance of the system. At this point, we emphasize that many of the works dealing with event-triggered control consider systems in the continuous-time domain. However, from the viewpoint of embedded technology, it is important to develop schemes that properly function in the discrete-time domain as well [14], [37], [48].

A Lyapunov-based distributed event-triggered protocol to solve the average consensus problem for MAS with single-integrator dynamics was presented [23]. The limitation of this work is that continuous monitoring of states of neighboring agents is required for controller updates. To overcome this limitation an event-triggering approach where each agent broadcasts its state to neighbors based on changes in its own state was presented [81]. We emphasize that in many works, state-independent thresholds are used. As a consequence, in this setting sampling actions cannot reflect the nature of system dynamics explicitly. Some authors

[89], [83] have considered hybrid event generation where events may be driven by both state and time dependent conditions. Distributed event-triggered consensus protocols have been proposed [67], [68] for weight balanced and strongly connected digraphs. We refer to the study [69], for a recent survey on the distributed protocols to solve consensus problem in MAS. These studies focus on agents with integrator dynamics.

Extensions of MAS consensus to agents having more general dynamics are of importance in view of applications. Observer-based consensus protocols for MAS with linear dynamics have been proposed in [51], [57], [84]. Based on the dynamics, it may not be possible to achieve static consensus. Instead, we aim to drive the states of the system to a common trajectory. The synchronization phenomenon [88] in MAS with continuous communication has been well researched [77], [80], [82]. In these domains, event-triggered protocols have been used to solve state consensus problems in MAS with general linear dynamics [34], [100], [102]. Time-dependent event-triggered protocols to solve state consensus problem have been presented in [29], [91]. Hu et al. [41] proposed an event-triggered distributed consensus protocol for linear MAS. There are a few recent works that address distributed consensus protocols for MAS with heterogeneous agents [101] and with non-linear dynamics [98]. It is noted that many of the works dealing with event-triggering schemes for MAS consider systems in the continuous-time domain.

A limitation of event-triggered schemes is that they require constant monitoring of the current state of the system to determine control update instants. In contrast, self-triggered schemes require only the last state measurement to determine the next control update instant [2], [3]. Self-triggered control for MAS has been well studied [23], [59], [95]. The work [23] demonstrated that self-triggering schemes require more controller updates compared with the event-triggering schemes. However, they are also found to be more robust against various disturbances and uncertainties [20]. One issue with these approaches is that though continuous sensing of states can be avoided, agents may still need to wait continuously for neighbors sending messages. Some works have tackled this problem by using event-triggered control with periodic sampling [62] and self-triggering control [94].

Studies for discrete-time agent systems can be found in [16] where a sufficient condition described by linear matrix inequalities was proposed to solve the average consensus problem based on event-triggered and self-triggered protocols. A particular problem of interest is distributed clock synchronization wherein all agents in the network synchronize their imperfect clocks via local communication. To reduce the communication requirement, Kadowaki and Ishii [45] and Garcia et al. [30] proposed event-triggered control schemes for clock synchronization problems in wireless sensor networks. Another interesting application is consensus-based formation control of multi-robot systems with event-triggered coordination to save energy expenses [31].

Since MAS often utilizes wireless communication, they are vulnerable to malicious attacks, making security and reliability a critical design concern. More recently, event-triggering control has been studied in MAS under cyber attacks as well: In Wang and Ishii [87] and Matsume et al. [59] resilient consensus algorithms for discrete-time agent systems with adversarial agents are developed based on both event- and self-triggering controls, where the adversarial agents may make arbitrary updates in their states. Moreover, Feng and Ishii [27] studied quantized consensus for leaderless and leader-follower linear MAS in the presence of denial-of-service (DoS) attacks.

In the thesis, we employ the new mechanism known as dynamic event-triggering schemes, which has been proposed by Girard [32]. There, controllers are equipped with threshold variables that take account of the state errors and can reduce the number of triggering instants compared with the event-triggering schemes independent of the state information. In Yi et al., [94] this idea has been extended to ensure that the triggering time sequence avoids exhibiting Zeno behaviors. Hu et al. [42] proposed a dynamic distributed event-triggered consensus protocol for MAS with general linear dynamics. These works are again restricted to the continuous-time cases.

2.2 Research Motivation

As we have reviewed in the previous subsection, much attention has been devoted to solve the state consensus problem of MAS with distributed control protocols in the continuous-time domain. However, in practice the control laws for agents have to be realized on digital platforms. We consider consensus problems of MAS in the discrete time. A significant advantage of working with discrete-time systems is that there is no concern of Zeno behavior. In continuous time settings, avoiding Zeno behavior in the sense that an infinite number of triggerings occur over a finite time interval is a challenging design concern.

To the best of our knowledge, there is no study on state-dependent distributed protocols to solve consensus problems of MAS in the discrete-time framework. We develop distributed triggering based protocols where agents interact locally to exchange state information in order to reach a global objective. In particular, we develop triggering based protocols to solve the sate consensus problem with the objective of effectively reducing communication and control update requirement for agents to save energy expenses.

Chapter 3

Average Consensus in MAS with Single-integrator Dynamics

In this chapter, we consider average consensus problem of MAS in the discrete-time domain. We develop three triggering based protocols with different characteristics and advantages in terms of necessary computational resources and capabilities in reducing the frequency of communications and updates for each agent. The first two are event-based triggering protocols, where the agents decide to transmit their states when the difference from the last transmitted data becomes larger than certain thresholds. In both of these event-based protocols, the thresholds are state dependent while the difference lies in whether the thresholds employ auxiliary variables or not. For this reason, they will be referred to as static and dynamic event-triggered protocols and are discussed in Sections 3.2 and 3.3, respectively. The third protocol relies on a slightly different mechanism for the choice of the events. When an agent makes a transmission, it calculates its next transmission time and send it along with the current state information. The self-triggered protocol will be introduced in Section 3.4. A numerical example is provided to demonstrate the performances of the proposed protocols. This part is published in [64], [66].

3.1 Problem Formulation

In this section, we present some preliminaries on graph theory and multi-agent systems for average consensus.

3.1.1 Graph Theory

Consider the network described by the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, comprised of the set of nodes $\mathcal{V} = \{1, \ldots, N\}$ and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edge $\{i, j\} \in \mathcal{E}$ indicates that node i and node j can send messages to each other. The neighbor set of node i is given by $\mathcal{N}_i = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$. Moreover, the degree of node i is the number of its neighbors and is denoted as $d_i = |\mathcal{N}_i|$. The graph is assumed to be connected in the sense that for any pair of nodes, there is a sequence of edges from one to the other.

The adjacency matrix of $\mathcal{G}$ is given by $A = [a_{ij}] \in \mathbb{R}^{N \times N}$, where $a_{ij} = 1$ if $\{i, j\} \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix is given by $L = [\ell_{ij}] \in \mathbb{R}^{N \times N}$ with $\ell_{ij} = -a_{ij}$ for $i \neq j$ and $\ell_{ii} = d_i$. Note that A and L are symmetric, i.e., $A = A^T$ and $L = L^T$. It clearly holds $L\mathbf{1}_N = \mathbf{0}_N$, i.e., the row sums are zero for the Laplacian. Denote by λ_i the eigenvalues of the Laplacian; they are indexed in an increasing order: $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_N$, where $\lambda_1 = 0$ follows from the graph being connected.

The following lemma relates the connectivity of the graph and the eigenvalues of its Laplacian [94].

Lemma 3.1.1. *For a connected graph $\mathcal{G}$, its Laplacian L is positive semi-definite with $\lambda_1(L) = 0$ and $\lambda_2(L) > 0$, and moreover it holds $0 \leq \lambda_2(L) K_N \leq L$, where $K_N = I_N - \frac{1}{N}\mathbf{1}_N \mathbf{1}_N^T$.*

3.1.2 System Model

We consider the multi-agent system of N agents interacting over the network given by the connected graph $\mathcal{G}$. The agents are modeled as single integrators in discrete time and their dynamics are described as

$$x_i(k+1) = x_i(k) + u_i(k), \quad i \in \mathcal{V}, \tag{3.1}$$

where $x_i(k) \in \mathbb{R}$ is the state and $u_i(k) \in \mathbb{R}$ is the control input associated with agent i at step $k \in \mathbb{Z}_+$. For the agents to achieve consensus, they can follow the common protocol given by the control input [70]

$$u_i(k) = -\epsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)), \tag{3.2}$$

where $0 < \epsilon < 1/d_{\max}$ and $d_{\max} = \max_{i \in \mathcal{V}} d_i$.

It is well known that the multi-agent system (3.1) under the control law (3.2) can achieve average consensus in the sense that

$$\lim_{k \to \infty} x_i(k) = \frac{1}{N} \sum_{j \in \mathcal{V}} x_j(0), \quad i \in \mathcal{V},$$

for any initial states. That is, the states of all agents eventually coincide at the value which is the average of the initial states of all agents.

The consensus protocol (3.2) requires each agent i to have access to the exact state information of its neighbors and to update its control input $u_i(k)$ at every step k. This requirement leads to the conservative use of network resources. We are interested in synthesizing control protocols for average consensus that relax both of these requirements, i.e., agents decide when to perform these actions in a more opportunistic manner.

We finally rewrite the agent system dynamics in (3.1) and (3.2) in the vector form. Let $x(k) = [x_1(k) \cdots x_N(k)]^T$. Then, the closed-loop dynamics of the overall system can be given by

$$x(k+1) = x(k) - \epsilon L x(k). \tag{3.3}$$

Let the average of the states of all agents be $\overline{x}(k) = \mathbf{1}_N^T x(k) = (1/N)\sum_{i\in\mathcal{V}} x_i(k)$.

We can easily verify that for the system (3.3), the average is an invariant quantity, i.e.,

$$\overline{x}(k+1) = \overline{x}(k), \quad \forall k. \tag{3.4}$$

This is because the Laplacian L has zero row sums, i.e., $L\mathbf{1}_N = \mathbf{0}_N$.

3.2 Protocol 1: Static Event Triggered

In this section, we develop a distributed triggering protocol that dictates when agents should broadcast their state information and update their control inputs. Here, we introduce the static version of the event-triggering control algorithm. As we will see, this will serve as the basis also for the dynamic version developed in the next section.

A distributed event-triggered implementation of controller (3.2) for agent i is given by

$$u_i(k) = -\epsilon \sum_{j\in\mathcal{N}_i} (\hat{x}_i(k) - \hat{x}_j(k)). \tag{3.5}$$

Here, $\hat{x}_i(k)$ is the last broadcast state of agent i at time step k and is more specifically given by

$$\hat{x}_i(k) = x_i(t_l^i), \quad k \in [t_l^i, t_{l+1}^i),$$

where $t_0^i, t_1^i, \ldots \in \mathbb{Z}_+$ represent the sequence of event times of agent i.

Note that the latest broadcast state $\hat{x}_j(k)$ of neighboring agent $j \in \mathcal{N}_i$ appears in the control protocol for agent i. This is because agent i has access to only the last broadcast states $\hat{x}_j(k)$ of its neighbors instead of their true states $x_j(k)$ from the current time k. As a consequence, in this setting, the control inputs are constant between the triggering times. An alternative approach would be to use $x_i(k)$ instead of $\hat{x}_i(k)$ for agent i's own state, but this will obviously increase the frequency of changes in the input value $u_i(k)$; hence this is not pursued in our work.

We develop an asynchronous triggering protocol that dictates when agents should broadcast their state information to neighboring agents so that the agents achieve average consensus. The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$. Agent i determines its triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min\Big\{k > t_l^i : \big(\hat{x}_i(k-1) - x_i(k)\big)^2 > \alpha_i(k)\hat{q}_i(k-1)\Big\}, \tag{3.6}$$

where the threshold is a product of two variables. One is

$$\alpha_i(k) = \frac{\sigma_i(k)}{2\epsilon d_i},$$

where $\sigma_i(k)$ takes nonnegative values and exponentially decreases to zero. The

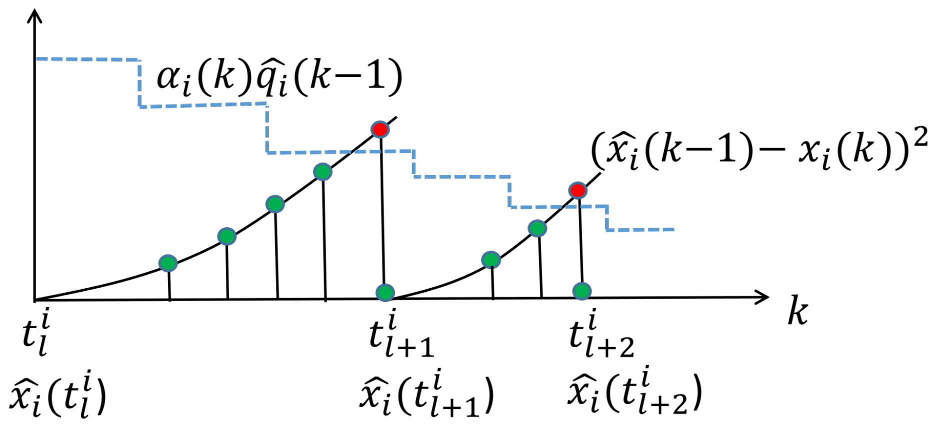

Figure 3.1: Static event-triggering law

other is $\hat{q}_i(k)$ given by

$$\hat{q}_i(k) = \min\left\{\frac{1}{2}\sum_{j\in\mathcal{N}_i}\left(\hat{x}_j(t^j_{l_j}) - \hat{x}_i(t^i_{l_i})\right)^2, M\right\} \geq 0, \tag{3.7}$$

where M is a positive constant. This M enables us to keep $\hat{q}_i(k)$ bounded by a known value. We will discuss its role in this protocol later. The variable $\hat{q}_i(k)$ depends on the states of the neighboring agents, i.e., it depends on local consensus error. The threshold is large, when the agent is far from consensus with its neighbors and hence the control need not be very accurate. This helps in reducing triggering instants without sacrificing convergence performance.

Let $e_i(k) = \hat{x}_i(k) - x_i(k)$, which is the error in the state and its last broadcast value for agent i. Notice that as a consequence of (3.6), it holds

$$|e_i(k)|^2 \leq \alpha_i(k)\hat{q}_i(k-1). \tag{3.8}$$

We refer to (3.6) as the static triggering protocol since it does not involve any extra variable other than the agent states $x_i(k)$, $\hat{x}_i(k)$, and $\hat{x}_j(k)$. The triggering law is distributed since each agent requires only its own and its neighbors' state information. We illustrate the triggering instants in Fig. 3.1. Here, the solid line indicates the state error $(\hat{x}_i(k-1) - x_i(k))^2$ whereas the dashed line is the threshold $\alpha_i(k)\hat{q}_i(k-1)$ used in the triggering law (3.6). As soon as the state error (in green circles) exceeds the threshold (and becomes red), the new state $\hat{x}_i(t^i_{l+1})$ is broadcasted (and the circle goes to zero in green again).

Remark 3.2.1. *Here, we should emphasize that when dealing with triggering rules in the discrete-time domain, the exact amount of the state error exceeding the threshold at the triggering instants is unknown in advance and is in general nonzero. This is a major aspect different from the continuous-time counterpart, where the triggering times are exactly when the state error becomes equal to the threshold [23], [94]. This property makes the analysis somewhat more complicated in the discrete-time case. We also note that in* (3.6)*, there is time difference between* $\hat{x}_i(k-1)$ *and* $x_i(k)$ *in the state error to be watched; this notation is also unique to the discrete-time case studied in this paper. We will discuss these points more later.*

Remark 3.2.2. *A conventional method for reducing controller updates is to use time-scheduled control with periodic updates at a constant period in discrete time [70], [90]. The control update instants of agents can be given as* $t_{l+1} = t_l + \tau_s$ *with* $\tau_s \in \mathbb{Z}_+$ *and* $t_0 = 0$*. It was shown in Xie et al. [90] that the protocol* (3.2) *solves average consensus problem if and only if* $0 < \tau_s < 2/(\epsilon\lambda_N(L))$*. As an event-triggering method, a simpler approach is to use triggering with a threshold independent of the agent states. To ensure asymptotic consensus, the thresholds should be time varying, e.g., as*

$$t^i_{l+1} = \min\Big\{k > t^i_l : \big(\hat{x}_i(k-1) - x_i(k)\big)^2 > c_i e^{-h_i k}\Big\}, \tag{3.9}$$

where the parameters are taken as $c_i, h_i > 0$ *[45], [81]. We will compare the* *performance of these protocols with the proposed ones by a numerical example in Section 3.5.*

Let $\hat{x}(k) = [\hat{x}_1(k) \cdots \hat{x}_N(k)]^T$ be the broadcast state vector, and let $e(k) = \hat{x}(k) - x(k)$ be the vector of errors in the broadcast states. We can rewrite the agent system (3.1) under the protocol (3.5) in the vector form as

$$x(k+1) = x(k) - \epsilon L\hat{x}(k) = x(k) - \epsilon Lx(k) - \epsilon Le(k). \tag{3.10}$$

Even in presence of the error $e(k)$, we can verify that in this system, the average remains an invariant quantity as in (3.4).

The distributed static event-triggered control protocol outlined so far can be summarized in an algorithm form as displayed in Algorithm 1. We are now ready to state the main result demonstrating that this protocol can be used to solve the average consensus problem. In the controller (3.5), take the parameter $\epsilon > 0$ small enough that $\Omega \in (0, 1)$, where

$$\Omega = \frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max})\lambda_2(L). \tag{3.11}$$

Theorem 3.2.1. *Consider the multi-agent system* (3.1) *with the connected underlying communication graph* $\mathcal{G}$. *Under Algorithm 1, the event-triggered control protocol* (3.5) *and* (3.6) *achieves average consensus asymptotically.*

Proof. Consider the Lyapunov candidate given by

$$V(k) = [x(k) - \overline{x}(k)\mathbf{1}_N]^T[x(k) - \overline{x}(k)\mathbf{1}_N]. \tag{3.12}$$

We now look at the difference between its values at times k and $k+1$ as

$$\nabla V(k) = V(k+1) - V(k).$$

Algorithm 1 Static Event-Triggered Control Algorithm

Step 0) Initialize ϵ, σ_i, $l_i = 0$ and $t_0^i = 0$.
At time k, agent i performs the following actions:

Step 1) If $[\hat{x}_i(k-1) - x_i(k)]^2 > \alpha_i(k)\hat{q}_i(k-1)$, then agent i broadcasts the state $x_i(k)$ to its neighbors and updates its broadcast state as $\hat{x}_i(k) = x_i(k)$. Its latest triggering time is set as $t_{l_i+1} = k$.

Step 2) If new states $\hat{x}_j(t^j_{l_j+1})$ are received from some neighbors, then agent i stores these values.

Step 3) Agent i applies the control input $u_i(k)$ given in (3.5).

Step 4) Agent i sets l_i to $l_i + 1$ and goes back to Step 1.

From (3.10) and (3.4), we have

$$\begin{aligned}\nabla V(k) = &[x(k) - \epsilon L\hat{x}(k) - \overline{x}(k)\mathbf{1}_N]^T[x(k) - \epsilon L\hat{x}(k) - \overline{x}(k)\mathbf{1}_N] \\ &- [x(k) - \overline{x}(k)\mathbf{1}_N]^T[x(k) - \overline{x}(k)\mathbf{1}_N].\end{aligned}$$

We drop the time index k to simplify the notation. Then, we have

$$\begin{aligned}\nabla V &= [(x - \overline{x}\mathbf{1}_N) - \epsilon L\hat{x}]^T[(x - \overline{x}\mathbf{1}_N) - \epsilon L\hat{x}] - (x - \overline{x}\mathbf{1}_N)^T(x - \overline{x}\mathbf{1}_N) \\ &= -2\epsilon(x - \overline{x}\mathbf{1}_N)^T L\hat{x} + \epsilon^2\hat{x}^T L^T L\hat{x} \\ &= -2\epsilon(\hat{x} - e)^T L\hat{x} + \epsilon^2\hat{x}^T L^T L\hat{x} \\ &= -2\epsilon\hat{x}^T L\hat{x} + 2\epsilon e^T L\hat{x} + \epsilon^2\hat{x}^T L^T L\hat{x}. \end{aligned} \tag{3.13}$$

We now upper bound the far right-hand side of (3.13) by applying to the second term the inequality $ab \leq a^2 + \frac{1}{4}b^2, \ \ \forall a, b \in \mathbb{R}$, as

$$e^T L\hat{x} = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} e_i(\hat{x}_i - \hat{x}_j) \leq \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} e_i^2 + \frac{1}{4}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}(\hat{x}_i - \hat{x}_j)^2 \leq \sum_{i\in\mathcal{V}} d_i e_i^2 + \frac{1}{2}\hat{x}^T L\hat{x}$$

and to the third term the relation

$$\hat{x}^T L^T L \hat{x} = \sum_{i\in\mathcal{V}} \Big[\sum_{j\in\mathcal{N}_i} (\hat{x}_i - \hat{x}_j) \Big]^2 \le \sum_{i\in\mathcal{V}} d_i \sum_{j\in\mathcal{N}_i} (\hat{x}_i - \hat{x}_j)^2 \le 2d_{\max} \hat{x}^T L \hat{x}.$$

Then, we obtain

$$\nabla V \le -(\epsilon - 2\epsilon^2 d_{\max}) \hat{x}^T L \hat{x} + 2\epsilon \sum_{i\in\mathcal{V}} d_i e_i^2. \tag{3.14}$$

From (3.8) and then (3.7), we have

$$\begin{aligned}\nabla V &\le -(\epsilon - 2\epsilon^2 d_{\max}) \hat{x}^T(k) L \hat{x}(k) + \sum_{i\in\mathcal{V}} \sigma_i(k) \hat{q}_i(k-1) \\ &\le -(\epsilon - 2\epsilon^2 d_{\max}) \hat{x}^T(k) L \hat{x}(k) + \sigma_{\max}(k) NM,\end{aligned} \tag{3.15}$$

where $\sigma_{\max}(k) = \max_{i\in\mathcal{V}} \sigma_i(k)$. Notice that

$$\begin{aligned}x^T L x = (\hat{x} - e)^T L (\hat{x} - e) &\le 2\hat{x}^T L \hat{x} + 2e^T L e \\ &\le 2\hat{x}^T L \hat{x} + 2\|L\| \|e\|^2,\end{aligned} \tag{3.16}$$

where the first inequality holds since the Laplacian L is positive semi-definite and thus $2a^T L b \le a^T L a + b^T L b$, $\forall a, b \in \mathbb{R}^N$. Also, $\|\cdot\|$ denotes the Euclidean norm for vectors and its induced norm for matrices [63]. We apply (3.8) to (3.16) and obtain

$$\begin{aligned}x^T(k) L x(k) &\le 2\hat{x}^T(k) L \hat{x}(k) + 2\|L\| \sum_{i\in\mathcal{V}} \alpha_i(k) \hat{q}_i(k-1) \\ &\le 2\hat{x}^T(k) L \hat{x}(k) + \frac{\|L\| \sigma_{\max}(k) NM}{\epsilon d_{\min}},\end{aligned}$$

where $d_{\min} = \min_{i \in \mathcal{V}} d_i$. Hence,

$$\hat{x}^T(k) L \hat{x}(k) \geq \frac{1}{2} x^T(k) L x(k) - \frac{\|L\| \sigma_{\max}(k) N M}{2 \epsilon d_{\min}}.$$

Using this relation in (3.15), we have

$$\nabla V \leq -\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max}) \lambda_2(L) V(k) + \sigma_{\max}(k) N M \left(1 + \|L\| \frac{1 - 2\epsilon d_{\max}}{2 d_{\min}} \right),$$

where the first term on the right-hand side is due to Lemma 3.1.1. Hence,

$$V(k+1) \leq (1 - \Omega) V(k) + \Delta_1(k), \tag{3.17}$$

where Ω is given in (3.11) and

$$\Delta_1(k) = \sigma_{\max}(k) N M \left(1 + \|L\| \frac{1 - 2\epsilon d_{\max}}{2 d_{\min}} \right). \tag{3.18}$$

Note that $\Omega \in (0, 1)$ from (3.11) and moreover, $\sigma_{\max}(k)$ is exponentially decreasing to zero. Thus, it follows that $V(k) \to 0$ as $k \to \infty$. By (3.4), this implies that average consensus is obtained asymptotically. □

Here, our algorithm can be viewed as the discrete-time version of the event-triggering based algorithm from [94], which is for the continuous-time case. The main difference is that in continuous time, the events can occur as soon as the threshold such as the one in (3.6) is exceeded, and hence the relation similar to (3.8) holds but in the form $|e_i(t)|^2 \leq \alpha_i \hat{q}_i(t)$, where the time variable t is the same on both sides of the inequality. In fact, this enables the use of a constant for σ_i in the threshold. In contrast, in the discrete-time case, we found it difficult to follow the same logic and have chosen this parameter σ_i as a decreasing function of time for the proof of Theorem 3.2.1. On the other hand, this makes the threshold in (3.6) to be a product of two variables, $\alpha_i(k)$ and $\hat{q}_i(k-1)$. These are both decreasing, which may be redundant. In fact, we will show by a numerical example later that with constant σ_i we can achieve average consensus. The static triggering law in the continuous-time case [94] may result in Zeno behavior, or fast switching, which can be problematic. When the network reaches consensus, the threshold approaches zero and the triggering condition will be violated every time, possibly leading to Zeno behavior. In the discrete-time setting there is no concern for the Zeno phenomenon.

In the proof, we observe that the convergence rate of the MAS network can be bounded using the Lyapunov function in (3.12) as

$$V(k) \leq (1-\Omega)^k V(0) + \sum_{l=0}^{k-1} (1-\Omega)^{k-l-1} \Delta_1(l). \tag{3.19}$$

This indicates that if we select a large value of M in (3.7), then there is a chance that the convergence rate becomes slower. That is, there is a potential trade-off between the convergence rate and triggering frequency.

3.3 Protocol 2: Dynamic Event Triggered

The motivation to use event-triggered control is to reduce actuation requirements and communication burden. As we have seen in the static event-triggering protocol from the last section, the threshold associated with the state error becomes smaller as the system approaches consensus, which may lead to frequent triggering instants. In order to further reduce the number of triggering instants, in this section, we next propose a dynamic triggering law that involves an auxiliary variable for each agent to regulate its threshold dynamically. This approach is based on the static triggering protocol (3.6) and similarly our dynamic triggering law is also distributed.

For the dynamic case, each agent i is equipped with an additional variable χ_i following the update rule given by

$$\chi_i(k+1) = (1-\beta_i)\chi_i(k) + \delta_i\big(\sigma_i(k)\hat{q}_i(k-1) - 2\epsilon d_i e_i^2(k)\big), \tag{3.20}$$

where initial value is set as $\chi_i(0) > 0$ and the parameters $\beta_i \in (0,1)$ and $\delta_i \in [0,1)$ are to be designed. The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$. Agent i determines its triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min\Big\{k > t_l^i : \theta_i\big[2\epsilon d_i\big(\hat{x}_i(k-1) - x_i(k)\big)^2 - \sigma_i(k)\hat{q}_i(k-1)\big] > \chi_i(k)\Big\} \tag{3.21}$$

with $\hat{q}_i(k)$ and $\chi_i(k)$ defined in (3.7) and (3.20), respectively. Also, $\theta_i > 0$. From (3.21), it follows that at all times k,

$$\theta_i\big(2\epsilon d_i e_i^2(k) - \sigma_i(k)\hat{q}_i(k-1)\big) \le \chi_i(k), \tag{3.22}$$

that is,

$$e_i^2(k) \le \frac{1}{2\epsilon\theta_i d_i}\big(\chi_i(k) + \theta_i\sigma_i(k)\hat{q}_i(k-1)\big). \tag{3.23}$$

The design parameters θ_i, β_i and δ_i are chosen to satisfy the conditions given as

$$\theta_i > \frac{1-\delta_i}{\beta_i},\ \ \delta_i + \beta_i < 1 \text{ and } \delta_i < \theta_i\beta_i. \tag{3.24}$$

From (3.20) and (3.22), we have

$$\chi_i(k+1) \geq \left(1 - \beta_i - \frac{\delta_i}{\theta_i}\right) \chi_i(k).$$

Hence, under the conditions in (3.24), we can show that $\chi_i(k)$ is always positive:

$$\chi_i(k) \geq \left(1 - \beta_i - \frac{\delta_i}{\theta_i}\right)^k \chi_i(0) > 0. \tag{3.25}$$

The static triggering protocol can be considered as a limiting case of the dynamic triggering protocol when θ_i attains a very large value as we see by comparing the bounds (3.8) and (3.23) for the state errors. On the other hand, from (3.21) and (3.25), we can conclude that a larger initial value of $\chi_i(0)$ results in larger inter-event times, i.e., larger differences between consecutive triggering instants. In particular, we also observe that when $\chi_i(0) = 0$, the dynamic triggering rule reduces to the static triggering rule and therefore the dynamic event-triggered law generates fewer triggering instants in comparison to the static event-triggered law.

The distributed dynamic event-triggered control protocol described so far is presented as Algorithm 2 for the average consensus problem. The following theorem is the main result for the agent systems under this control law.

Theorem 3.3.1. *Consider the multi-agent system* (3.1) *with the connected underlying communication graph $\mathcal{G}$. Under Algorithm 2, the event-triggered control protocol* (3.5) *and* (3.21) *achieves average consensus asymptotically.*

Proof. Consider the Lyapunov candidate given by (3.12). Then, we look at its time difference and upper bound it as it was done in (3.14) in the proof of Theorem

Algorithm 2 Dynamic Event-Triggered Control Algorithm

Step 0) Initialize ϵ, σ_i, $\delta_i \in [0,1)$, $\beta_i \in (0,1)$, $\chi_i(0) > 0$, $\theta_i > 0$, $l_i = 0$ and $t_0^i = 0$. At time k, agent i performs the following actions:

Step 1) If $[\hat{x}_i(k-1) - x_i(k)]^2 > \frac{\chi_i(k)}{2\epsilon\theta_i d_i} + \alpha_i(k)\hat{q}_i(k-1)$, then agent i broadcasts the state $x_i(k)$ to its neighbors and updates its broadcast state as $\hat{x}_i(k) = x_i(k)$. Its latest triggering time is set as $t_{l_i+1} = k$.

Step 2) If new states $\hat{x}_j(t^j_{l_j+1})$ are received from some neighbors, then agent i stores these values.

Step 3) Agent i applies the control input $u_i(k)$ given in (3.5).

Step 4) Agent i sets l_i to $l_i + 1$ and goes back to Step 1.

3.2.1:

$$\nabla V \leq -(\epsilon - 2\epsilon^2 d_{\max})\hat{x}^T(k)L\hat{x}(k) + 2\epsilon \sum_{i\in\mathcal{V}} d_i e_i^2(k). \tag{3.26}$$

Now from (3.16) and (3.23), we have

$$\begin{aligned} x^T(k)Lx(k) &\leq 2\hat{x}^T(k)L\hat{x}(k) + \frac{\|L\|\sigma_{\max}(k)}{\epsilon \min_i d_i}\sum_{i\in\mathcal{V}}\hat{q}_i(k-1) + \frac{\|L\|}{\epsilon \min_i \theta_i d_i}\sum_{i\in\mathcal{V}}\chi_i(k) \\ &\leq 2\hat{x}^T(k)L\hat{x}(k) + \frac{\|L\|\sigma_{\max}(k)NM}{\epsilon d_{\min}} + \frac{\|L\|}{\epsilon \min_i \theta_i d_i}\sum_{i\in\mathcal{V}}\chi_i(k). \end{aligned}$$

Using this relation in (3.26), we have

$$\begin{aligned} \nabla V \leq & -\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max})x^T(k)Lx(k) + \sigma_{\max}(k)NM\left(1 + \|L\|\frac{1-2\epsilon d_{\max}}{2d_{\min}}\right) \\ & + \left[\frac{\|L\|(1-2\epsilon d_{\max}) + 2d_{\max}}{2\min_i \theta_i d_i}\right]\sum_{i\in\mathcal{V}}\chi_i(k). \end{aligned}$$

From (3.18), we have

$$\nabla V \leq -\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max})\lambda_2(L)V(k) + \Delta_1(k) + K_1 \sum_{i\in\mathcal{V}} \chi_i(k),$$

where $K_1 = \frac{\|L\|(1-2\epsilon d_{\max})+2d_{\max}}{2\min_i \theta_i d_i}$. Recall that $\Delta_1(k)$ goes to zero exponentially. From (3.11), we arrive at

$$V(k+1) \leq (1-\Omega)V(k) + \Delta_1(k) + K_1 \sum_{i\in\mathcal{V}} \chi_i(k). \tag{3.27}$$

Now, by (3.20), (3.23) and (3.25),

$$\chi_i(k+1) \leq \left(1 - \beta_i + \frac{\delta_i}{\theta_i}\right)\chi_i(k) + 2\delta_i\sigma_i(k)M.$$

Therefore, we can bound $\chi_i(k)$, which goes to zero by (3.24). Hence, from (3.27), it follows that $V(k) \to 0$ as $k \to \infty$. This implies that average consensus is obtained asymptotically. □

As we discussed earlier, the dynamic event-triggering approach can be seen as a generalization of the static counterpart of Section 3. For this reason, the proof of Theorem 3.3.1 is built upon that of Theorem 3.2.1. While the dynamic protocol is capable to reduce the frequency of the triggering instants, this point may not be obvious in the proof. Indeed, comparing the bounds on $V(k+1)$ in (3.17) for the static case and (3.27) for the dynamic case, we notice that the latter one is larger because of the additional terms resulting from the auxiliary variables $\chi_i(k)$. We will compare the two protocols through numerical simulations in Section 3.5 and confirm that the dynamic protocol indeed performs better in reducing the frequency of triggering instants.

3.4 Protocol 3: Self Triggered

So far, we have presented the static and dynamic event-triggered control protocols, which are both effective in reducing data transmissions among agents. However, one disadvantage of these approaches is that they require each agent to continuously monitor its own state and to continuously listen to its neighbors. Such continuous monitoring can be expensive in terms of computation resources. This requirement can be avoided if each agent determines its next triggering time at the current triggering time and broadcasts this information to its neighbors. In this section, we discuss such a self-triggered control strategy wherein each agent only needs to make updates/broadcasts at its own triggering instants and listen to its neighbors at their triggering instants. To this end, each agent must find an estimate of its future state at the time of making a transmission. We present how this can be done.

First, recall that the agents update their states based on (3.1) and (3.5) as

$$x_i(k+1) = x_i(k) - \epsilon \sum_{j \in \mathcal{N}_i} (\hat{x}_i(k) - \hat{x}_j(k)). \tag{3.28}$$

Because of the event-based protocol, the state of agent i at step $k \in [t_l^i, t_{l+1}^i)$ can be expressed as

$$x_i(k) = x_i(t_l^i) - (k - t_l^i)\epsilon \sum_{j \in \mathcal{N}_i} (\hat{x}_i(k) - \hat{x}_j(k)),$$

where t_l^i is the latest triggering time of agent i. By definition of the triggering times t_l^i for agent i, we have $x_i(t_l^i) = \hat{x}_i(k)$. Thus, the error $e_i(k) = \hat{x}_i(k) - x_i(k)$

can be expressed as

$$e_i(k) = (k - t_l^i)\epsilon \sum_{j \in \mathcal{N}_i} u_{ij}(k),$$

where we set $u_{ij}(k)$ as the difference in the broadcast values of agents i and j:

$$u_{ij}(k) = \hat{x}_i(k) - \hat{x}_j(k).$$

We need to emphasize here that $u_{ij}(k)$ may not be constant for all $k \in [t_l^i, t_{l+1}^i)$ since neighbors may trigger any time in this interval. Consequently, it is hard to know the value of $e_i(k)$ in the interval $k \in (t_l^i, t_{l+1}^i)$. To proceed to find some estimate on $e_i(k)$, we next look for an upper bound of $u_{ij}(k)$. Then the next triggering instant t_{l+1}^i can be determined at current triggering instant t_l^i accordingly.

Our approach here is to ensure that the error $e_i(k)$ decreases exponentially as

$$\big|e_i(k)\big| \leq \frac{\eta}{\sqrt{d_i}} \gamma^{\frac{k}{2}}, \tag{3.29}$$

where $\eta > 0$ and $\gamma \in (0, 1)$. Note that this is a sufficient condition for the inequality in (3.22) since $\hat{q}_i(k) \geq 0$ and (3.25) holds.

We now provide the protocol based on self-triggered control. First, the parameter $\epsilon > 0$ in the control (3.28) is chosen sufficiently small that $\Phi \in (0, 1)$, where

$$\Phi = 1 - (\epsilon - 4\epsilon^2 d_{\max})\lambda_2(L). \tag{3.30}$$

Also let

$$\Gamma = 2\epsilon(1 + \epsilon)N\eta^2. \tag{3.31}$$

Furthermore, we introduce the following function to be used by agent i:

$$g_i(k) = \epsilon \left| \sum_{j \in \mathcal{N}_i} (t_{ij}^1 - t_l^i) u_{ij}(t_l^i) \right| + \epsilon \sum_{j \in \mathcal{N}_i} \sum_{m=t_{l+1}^j}^{t_{ij}^2 - 1} \left[\left(\frac{\eta}{\sqrt{d_i}} + \frac{\eta}{\sqrt{d_j}} \right) \gamma^{\frac{m}{2}} + f(m) \right], \tag{3.32}$$

where

$$t_{ij}^1(k) = \min\{k, t_{l_j+1}^j\} \text{ and } t_{ij}^2(k) = \max\{k, t_{l_j+1}^j\} \text{ for } k \in [t_{l_i}^i, t_{l_i+1}^i) \text{ and} \tag{3.33}$$

$$f(m) = \left[2\Big(\Phi^m V_{\max} + \frac{\Phi^m - \gamma^m}{\Phi - \gamma} \Gamma \Big) \right]^{1/2}. \tag{3.34}$$

The parameter $V_{\max}$ in (3.34) must be chosen so as to upper bound the initial value of the Lyapunov candidate as $V(0) \leq V_{\max}$. Such a bound can be found in practice if the agents' state values are known to start in a certain interval. It is however clear that taking a tighter bound $V_{\max}$ would help the performance of the resulting control.

The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$ for the self-triggering algorithm. Agent i determines its next triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min \left\{ k > t_l^i : g_i(k) > \frac{\eta}{\sqrt{d_i}} \gamma^{\frac{k}{2}} \right\}. \tag{3.35}$$

We must note that in this triggering law, $g_i(k)$ in (3.32) increases while $\gamma^{k/2}$ decreases with respect to $k \in [t_l^i, t_{l+1}^i)$. Hence, given the past triggering instant t_l^i agent i can determine the next one t_{l+1}^i by solving (3.35).

We propose the self-triggered control protocol given as Algorithm 3 to solve the average consensus problem. Its capability to achieve average consensus is

Algorithm 3 Self-Triggered Control Algorithm

Step 0) Initialize η, γ, $l_i = 0$, and $t_0^i = 0$.

Step 1) At time t_l^i agent i performs the following three actions:
(i) Agent i uses its own state $x_i(t_l^i)$ and its neighbors' states $\hat{x}_j(t_l^i)$ to compute the control input $u_i(t_l^i)$ by (3.5).
(ii) Agent i determines the next triggering instant by (3.35).
(iii) Agent i broadcasts $(t_{l+1}^i, x_i(t_l^i))$ to its neighbors.

Step 2) At the triggering time $t_{l'}^j$ of neighbor $j \in \mathcal{N}_i$ with $t_{l'}^j \in [t_l^i, t_{l+1}^i)$, agent i updates its control input $u_i(k)$ as in (3.5).

Step 3) Agent i sets l_i to $l_i + 1$ and goes back to Step 1.

guaranteed by the following theorem.

Theorem 3.4.1. *Consider the multi-agent system* (3.1) *with the connected underlying communication graph* $\mathcal{G}$*. Under Algorithm 3, the self-triggered control protocol* (3.35) *achieves the average consensus asymptotically.*

Proof. We again use the Lyapunov candidate in (3.12). Its time difference can be upper bounded as in (3.14) in the proof of Theorem 3.2.1:

$$\begin{aligned}
\nabla V &= -2\epsilon x^T L x - 2\epsilon x^T L e + \epsilon^2 (x+e)^T L^T L (x+e) \\
&\le -2\epsilon \sum_{i\in\mathcal{V}} q_i + 2\epsilon \sum_{i\in\mathcal{V}} d_i e_i^2 + 2\epsilon \sum_{i\in\mathcal{V}} \frac{q_i}{2} + 4\epsilon^2 \sum_{i\in\mathcal{V}} d_i q_i + 2\epsilon^2 \sum_{i\in\mathcal{V}} d_i e_i^2 \\
&\le -2\epsilon x^T L x + 2\epsilon \sum_{i\in\mathcal{V}} d_i e_i^2 + \epsilon x^T L x + 4\epsilon^2 d_{\max} x^T L x + 2\epsilon^2 \sum_{i\in\mathcal{V}} d_i e_i^2, \qquad (3.36)
\end{aligned}$$

where the time index k is dropped for notational simplicity and

$$q_i = \frac{1}{2} \sum_{j\in\mathcal{N}_i} (x_i - x_j)^2 \ge 0.$$

Here, the difference from (3.14) is that q_i is used instead of $(1/2)\sum_{j\in\mathcal{N}_i}(\hat{x}_i - \hat{x}_j)^2$ used in (3.7) for the variable $\hat{q}_i(k)$. Thus, in comparison to (3.14), there are additional terms in (3.36). We can further bound ∇V using (3.29) as

$$\begin{aligned}\nabla V &\leq -\sum_{i\in\mathcal{V}}(\epsilon - 4\epsilon^2 d_i)q_i + 2\epsilon(1+\epsilon)\sum_{i\in\mathcal{V}} d_i e_i^2 \\ &\leq -(\epsilon - 4\epsilon^2 d_{\max})\lambda_2(L)V(k) + 2\epsilon(1+\epsilon)N\eta^2\gamma^k.\end{aligned}$$

From (3.30) and (3.31), we have

$$V(k+1) \leq \Phi V(k) + \Gamma\gamma^k.$$

By iteration we can write

$$V(k) \leq \Phi^k V_{\max} + \sum_{l=0}^{k-1}\Phi^{k-l-1}\Gamma\gamma^l = \Phi^k V_{\max} + \frac{\Phi^k - \gamma^k}{\Phi - \gamma}\Gamma.$$

We now find an upper bound on $|x_i(k) - x_j(k)|$. This is done using (3.34) as

$$\begin{aligned}|x_i(k) - x_j(k)| &\leq |x_i(k) - \overline{x}(0)| + |x_j(k) - \overline{x}(0)| \leq \sqrt{2(|x_i(k) - \overline{x}(0)|^2 + |x_j(k) - \overline{x}(0)|^2)} \\ &\leq \left[2\Big(\Phi^k V_{\max} + \frac{\Phi^k - \gamma^k}{\Phi - \gamma}\Gamma\Big)\right]^{1/2} \\ &= f(k).\end{aligned}$$

We thus obtain an upper bound of $|u_{ij}(k)|$ as

$$\begin{aligned}|u_{ij}(k)| &= |\hat{x}_i(k) - \hat{x}_j(k)| \\ &\leq |\hat{x}_i(k) - x_i(k)| + |x_i(k) - x_j(k)| + |x_j(k) - \hat{x}_j(k)| \\ &\leq \left(\frac{\eta}{\sqrt{d_i}} + \frac{\eta}{\sqrt{d_j}}\right)\gamma^{\frac{k}{2}} + f(k). \qquad (3.37)\end{aligned}$$

Next, we proceed to find an upper bound of $|e_i(k)|$. At time t_l^i, agent i already knows t_l^j, $x_j(t_l^j)$, and t_{l+1}^j for $j \in \mathcal{N}_i$. We know that $u_{ij}(k)$ is constant for $k \in [t_l^i, t_{ij}^1]$ where t_{ij}^1 is given in (3.33), and for $k > t_{ij}^1$, it can be upper bounded by (3.37). Hence, we have

$$|e_i(k)| = \left|\epsilon(k - t_l^i) \sum_{j \in \mathcal{N}_i} u_{ij}(k)\right| \leq g_i(k), k \in [t_l^i, t_{l+1}^i),$$

that is, $g_i(k)$ bounds $|e_i(k)|$ from above. Hence, under the self-triggering law of (3.35), it holds that

$$|e_i(k)| \leq \frac{\eta}{\sqrt{d_i}} \gamma^{\frac{k}{2}}, \;\; \forall k \in [t_l^i, t_{l+1}^i).$$

Therefore, all state errors decrease exponentially. This indicates that the multi-agent system following the self-triggered control law achieves average consensus asymptotically. □

Our algorithm can be viewed as a discrete-time version of the self-triggering algorithm of [94], which is for the continuous-time case. The main difference is that in continuous time, the events can take place as soon as the threshold such as the one in (3.35) is exceeded with an equality.

3.5 Numerical Example

In this section, we illustrate performance of the proposed protocols via numerical simulations. We first examine a small-scale network and then focus on the scalability for larger networks.

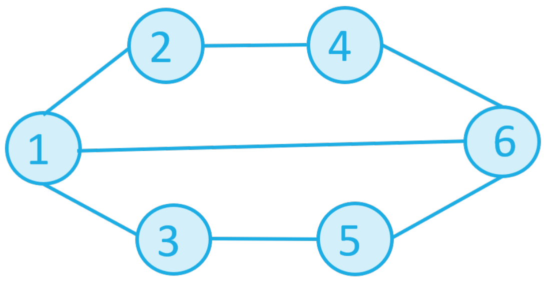

Figure 3.2: Network topology

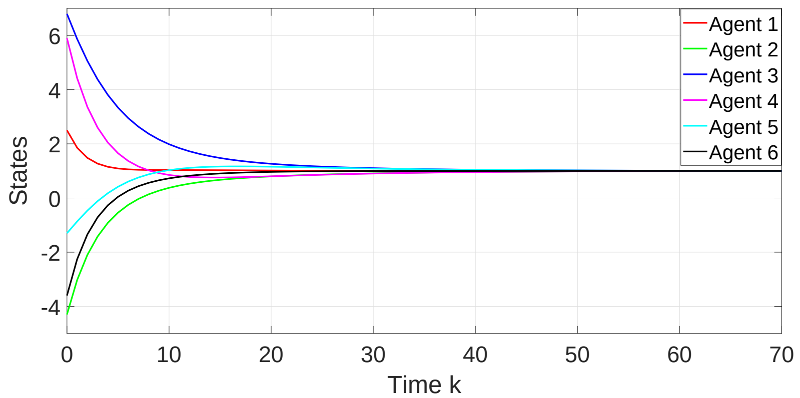

Figure 3.3: State evolutions under the original protocol (3.2)

3.5.1 Small-scale Network

Consider the connected network consisting of six agents shown in Fig. 3.2. We choose the initial states as $x(0) = \begin{bmatrix} 2.5 & -4.3 & 6.8 & 5.9 & -1.3 & -3.6 \end{bmatrix}$. The control objective for the network is to reach the average of agents' initial states, which is $\overline{x}(0) = 1$, in this case. Using the original control (3.2) with computation at every step, we show the time responses of the agent states in Fig. 3.3. The state errors become small around $k = 30$. In the following simulations, we ran the three proposed algorithms by choosing the parameters so that the states of the agents change at similar rates for achieving consensus with this case of conventional control.

First, we examined the static event-triggering protocol (3.6) with $\epsilon = .075$ and $\sigma_i = .1$. From (3.11), we obtained $\Omega = .021 < 1$, which thus satisfies the

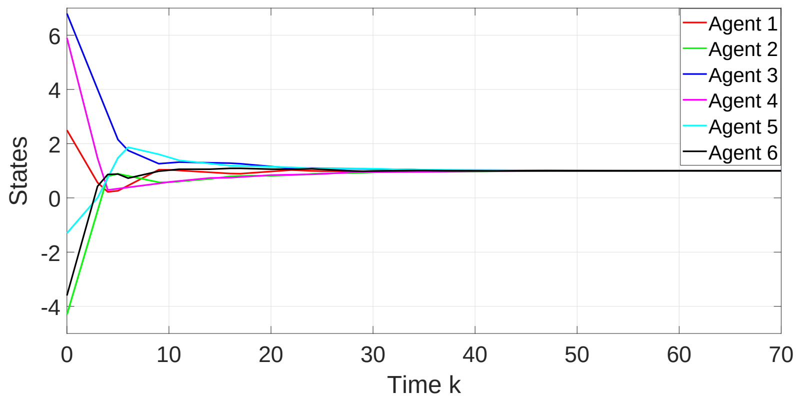

Figure 3.4: State evolutions under the static event-triggering law (3.6)

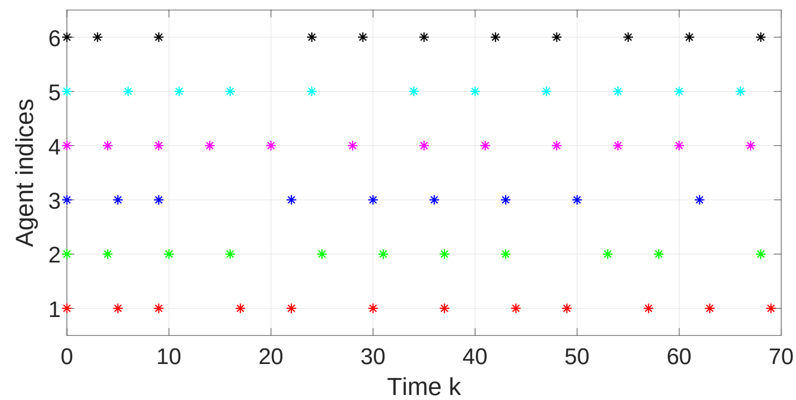

Figure 3.5: Triggering instants under the static event-triggering law (3.6)

condition there. The state evolution and the corresponding triggering instants for each agent are shown in Figs. 3.4 and 3.5, respectively. We observe that the triggering for the agents takes place infrequently, even after consensus is reached around time 45. The average number of triggering per agent is 11 times in 70 steps. We notice that the trajectories of the agent states are not as smooth as the original control case in Fig. 3.3. This may be because the control inputs are constant between the triggering times under the proposed protocol.

We now make comparisons with two existing triggering approaches discussed as Remark 3.2.2. We choose the parameters for these protocols so that the average number of triggering becomes similar to (or even larger than) that in the static

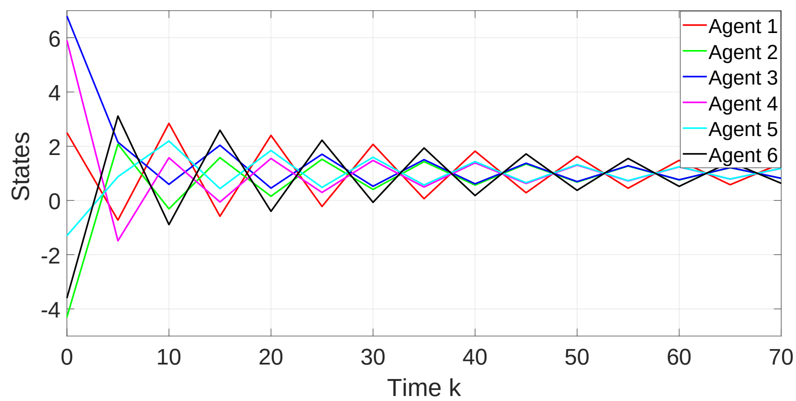

Figure 3.6: State evolutions under the periodic triggering law with period $\tau_s = 5$

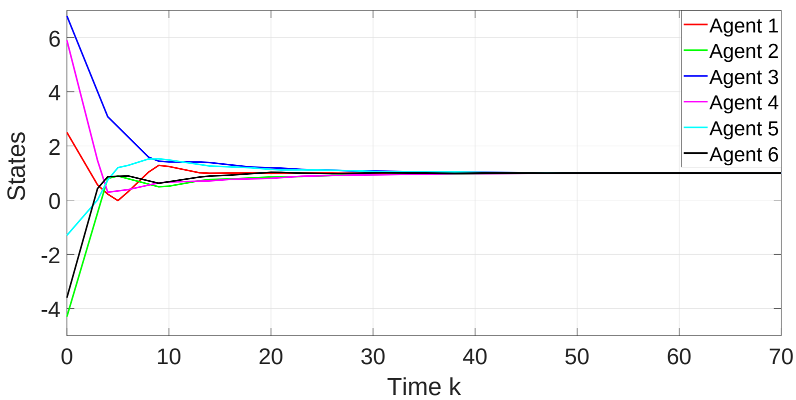

Figure 3.7: State evolutions under the state-independent triggering law (3.9) with $c_i e^{-h_i k} = 0.1e^{-.01k}$

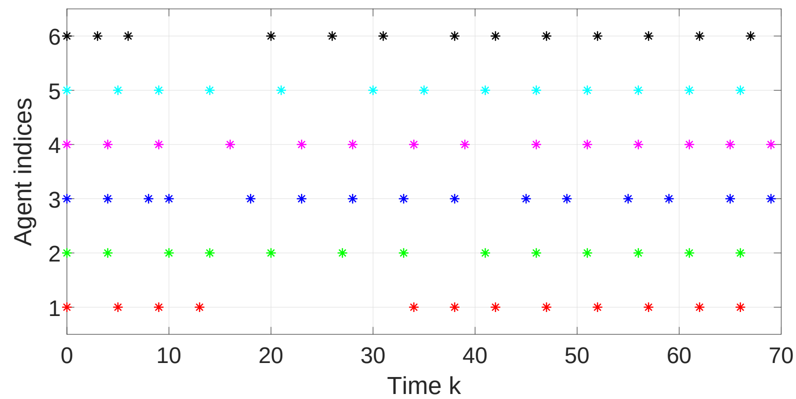

Figure 3.8: Triggering instants under the state-independent triggering law (3.9) with $c_i e^{-h_i k} = 0.1e^{-.01k}$

event-triggering protocol result above. For the key parameter ϵ, we kept the value of $\epsilon = 0.075$ as in the proposed protocols. The first is the time-scheduled control with periodic sampling in discrete time [70], [90]. The period is set as $\tau_s = 5$, in which case the average triggering number is 14 in 70 steps. The agent states are shown in Fig. 3.6, where they exhibit oscillations. In fact, by increasing the period to $\tau_s = 6$, the system becomes unstable because the consensus condition $\tau_s < 2/(\epsilon\lambda_N(L)) = 5.333$ from [90] is violated where for the graph in Fig 3.2, we have $\lambda_N(L) = 5$. Using smaller ϵ will result in smoother state evolutions and consensus, but the convergence becomes slower.

The second conventional case is with the event-triggered control [45], [81] in (3.9) with state independent threshold $0.1\mathrm{e}^{-.01k}$. The state evolution and the corresponding triggering instants for each agents are shown in Figs. 3.7 and 3.8, respectively. It is clear that this protocol demonstrates similar convergence rates but with more transmissions.

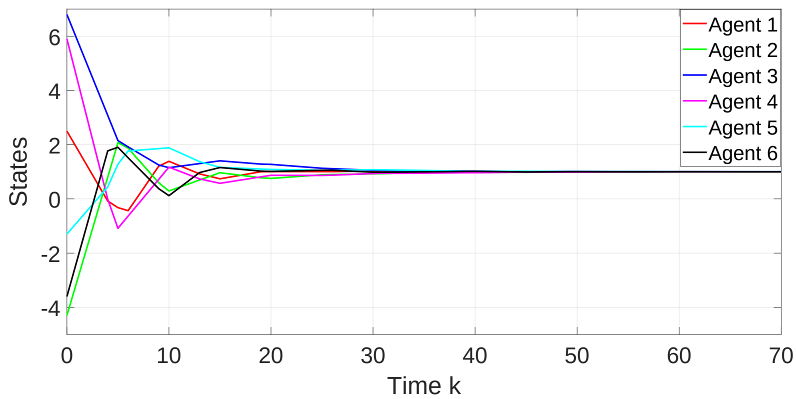

Figure 3.9: State evolutions under the dynamic event-triggering law (3.21)

Next, we used the dynamic event-triggering protocol (3.21). Here for the common parameters ϵ and σ_i, we used the same values as above. For the rest, we set $\delta_i = .6$, $\beta_i = .35$, $\chi_i(0) = 20$, $\theta_i = 10$. The state evolution and the corresponding triggering instants for each agent are shown in Figs. 3.9 and 3.10, respectively.

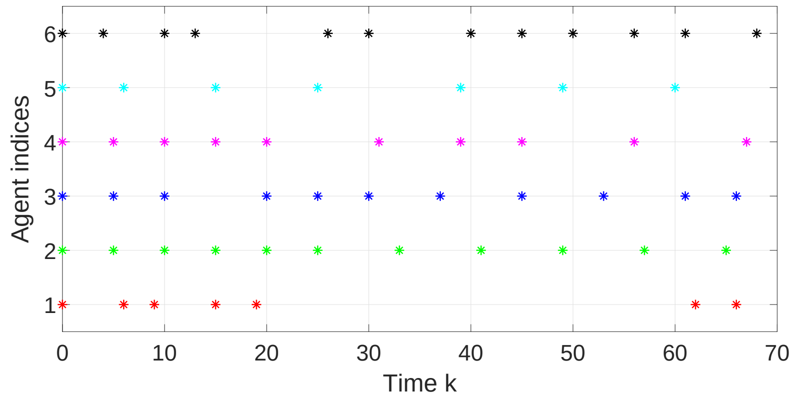

Figure 3.10: Triggering instants under the dynamic event-triggering law (3.21)

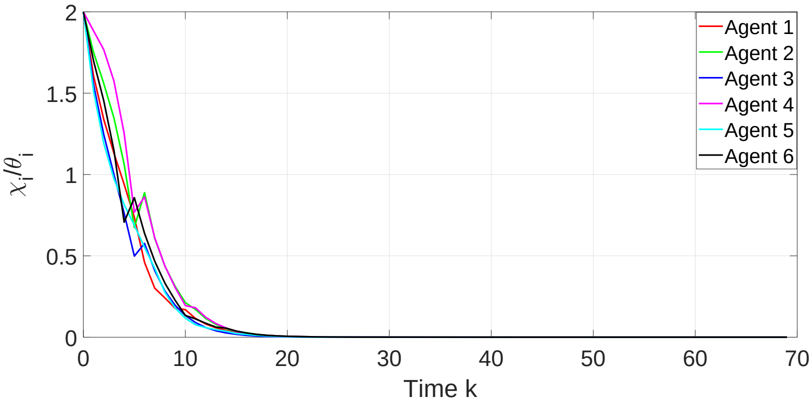

Figure 3.11: Thresholds for agents under the dynamic event-triggering law (3.21)

The state trajectories roughly follow the same convergence rates for both event-triggering laws. However, it is evident from Figs. 3.5 and 3.10 that the dynamic triggering law generates fewer triggering instants as compared to the static triggering law. The average number is 9.66 times in 70 steps. It is interesting to note that agent 1 has an especially low number of triggering instants; in the dynamic case in Fig. 3.10, there are 40 steps without any triggering starting around $k = 20$. Fig. 3.11 depicts the threshold variable $\chi_i(k)$ for each agent under the dynamic triggering law. As shown in Theorem 3.3.1, these thresholds converge to zero, but may increase sometimes. We also note that in these simulations, constant values are used for σ_i and yet, consensus is achieved. In the theoretical result, these were assumed to be converging to zero (at arbitrarily chosen rates).

Third, we discuss the self-triggering protocol (3.35). We selected the parameters $\eta = 15$ and $\gamma = .99$. We can confirm that the condition in (3.30) is satisfied since we get $\Phi = .9925 < 1$. The state evolution with self-triggered control law is presented in Fig. 3.12 and the corresponding triggering times for each agent are shown in Fig. 3.13. We observe that with the self-triggered control, the agents arrive at average consensus with about the same convergence performance as the other two protocols. However, it is evident that the number of triggering instants are much higher. While in the beginning, the triggering instants are reduced to about the same level as those in the event-triggered cases, they increase afterwards, indicating the conservatisms in the approach.

The average Lyapunov functions for the the static, dynamic and self-triggering protocols are shown in the left plot of Fig. 3.14. We ran Monte Carlo simulations of 100 times for these protocols. The initial states are chosen in the set $[-10\ 10]$ uniformly randomly and we used the same initial states for all these protocols. The convergence speeds are almost the same for three protocols. The difference lies in the average number of triggering instants per agent, which is 10.91, 9.92 and 31.33 for the static, dynamic and self-triggering protocols respectively. The right plot of Fig. 3.14 shows the results as a box plot.

In summary, we have confirmed that for the considered network with six agents in Fig. 3.2, the proposed triggering-based protocols perform well in achieving consensus at convergence rates similar to the original control law (3.2) with different characteristics in terms of transmission frequencies and necessary computational resources at agent levels. The dynamic triggered protocol is particularly effective as the threshold can be regulated dynamically for each agent.

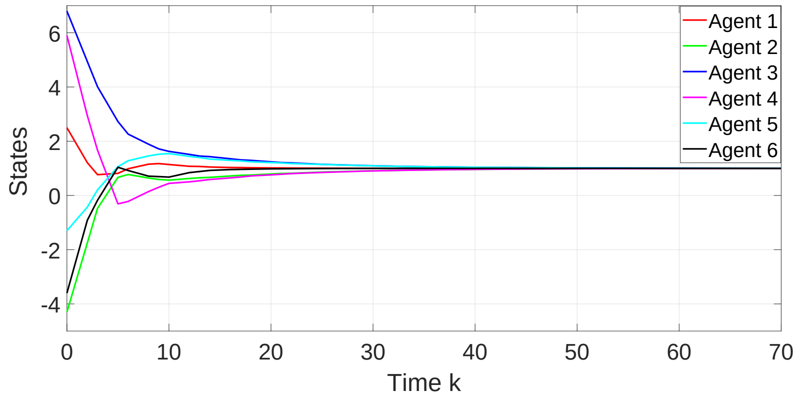

Figure 3.12: State evolutions under the self-triggering law (3.35)

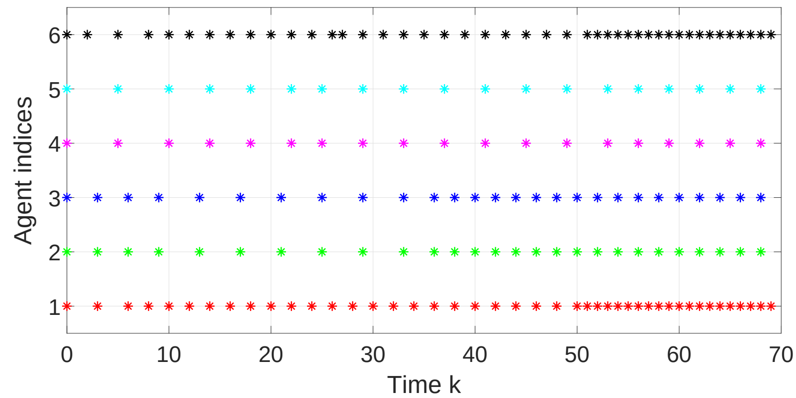

Figure 3.13: Triggering instants under the self-triggering law (3.35)

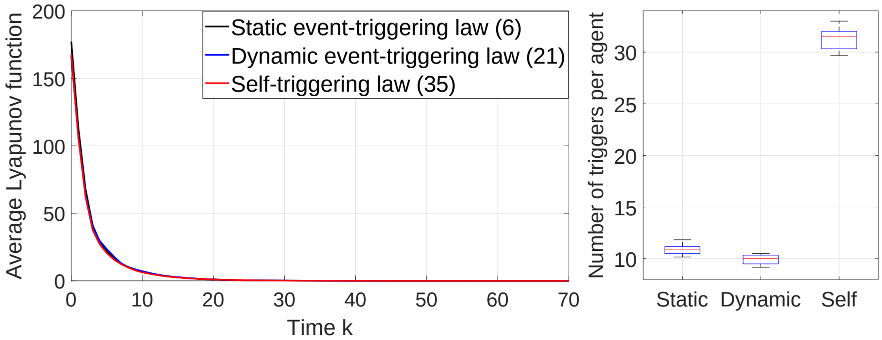

Figure 3.14: Average Lyapunov functions and the number of triggering instants per agent under the static (3.6), dynamic (3.21) and self-triggering protocols (3.35) from 100 Monte Carlo simulations

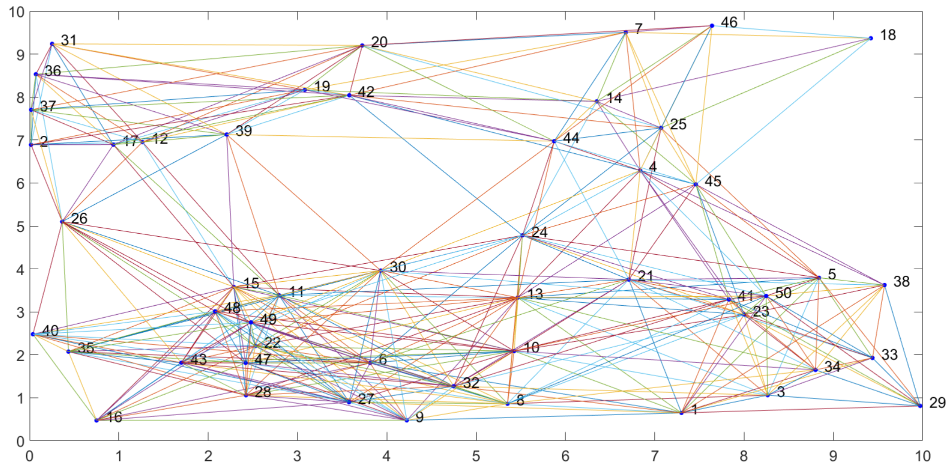

Figure 3.15: Random network with communication radius $r = 4$

3.5.2 Random Network

We next examine the scalability of the proposed protocols in a random network. Let us consider the network comprising of fifty agents randomly located in $[10 \times 10]$ XY plane. In order to compare the performances of the nominal, static, dynamic, and self-triggered protocols, we ran Monte Carlo simulations for 100 times. We set the initial state of nodes in the set $[-10\ 10]$ uniformly randomly and used the same value for all protocols.

We consider two nodes are connected if the communication radius $r = 4$, as shown in Fig 3.15. The average Lyapunov functions plots for the proposed protocols are shown in Fig 3.16. We observe that these plots almost coincide exhibiting similar convergence speed. The numbers of average triggering times for the proposed protocols to reach consensus, which is chosen at the point where Lyapunov candidate value becomes 0.1, are presented in Table 3.1. The consensus is obtained around $k = 59$.

We next demonstrate the performances of these protocols with $r = 5$, i.e., with increased number of edges. The average Lyapunov functions plots for the protocols are shown in Fig 3.17. We observe that these plots almost coincide. The numbers of average triggering times for the proposed protocols to reach consensus are shown in Table 3.2. We get consensus around $k = 27$.

We further increased the number of edges by selecting $r = 6$. The average Lyapunov functions plots for the protocols are shown in Fig 3.18, where these plots almost coincide. The numbers of average triggering times for the proposed protocols to reach around consensus are shown in Table 3.3. We obtain consensus around $k = 16$.

The proposed protocols exhibit similar convergence rates to the nominal protocol in achieving average consensus. From, these plots it appears that with more number of edges the convergence rate for the network become faster as well as there is a slight improvement in reducing communication and control update frequencies for agents.

Table 3.1: Performance under the proposed protocols with communication radius $r = 4$ in random network

Protocols	Static	Dynamic	Self
Average numbers of triggering times	15.4	12.8	39.3

Table 3.2: Performance under the proposed protocols with communication radius $r = 5$ in random network

Protocols	Static	Dynamic	Self
Average numbers of triggering times	10.6	8.4	21.2

Table 3.3: Performance under the proposed protocols with communication radius $r = 6$ in random network

Protocols	Static	Dynamic	Self
Average numbers of triggering times	7.8	5.7	12.3

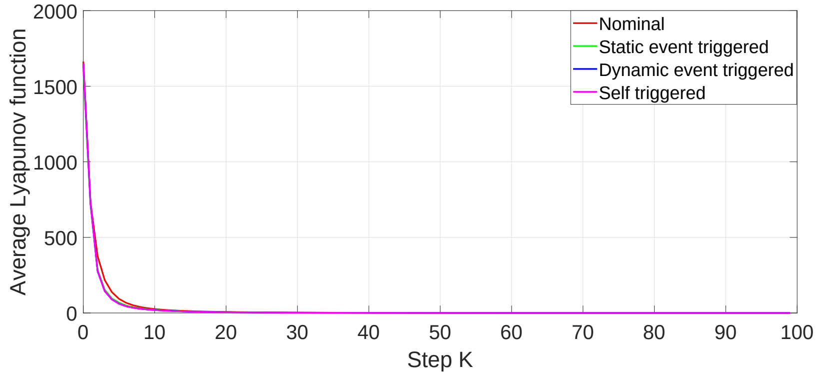

Figure 3.16: Average Lyapunov functions plots with $r = 4$ for 100 Monte Carlo simulations

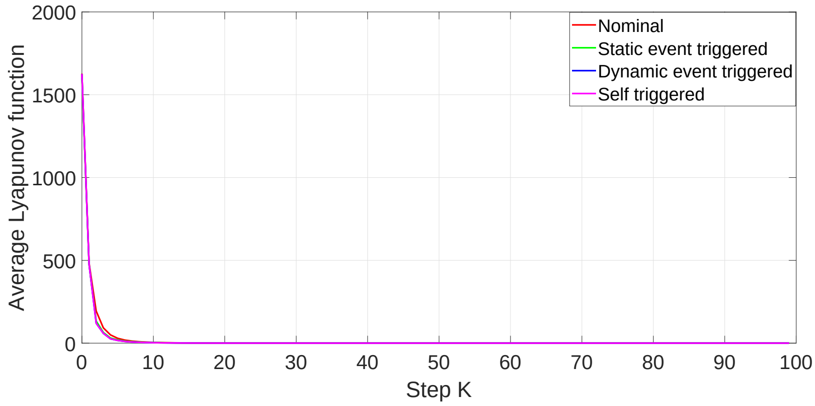

Figure 3.17: Average Lyapunov functions plots with $r = 5$ for 100 Monte Carlo simulations

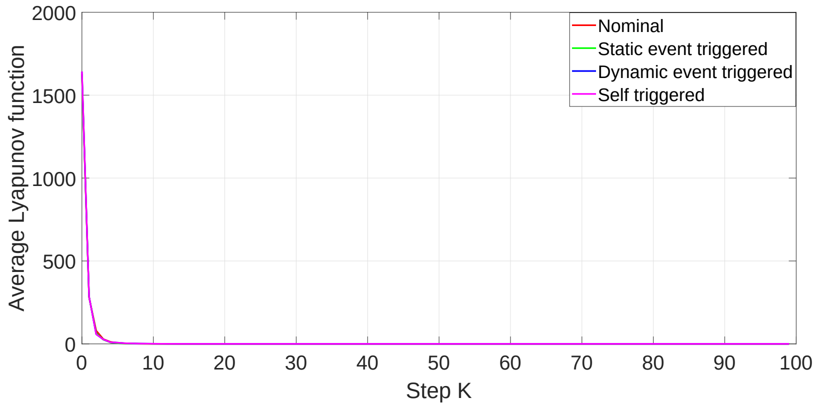

Figure 3.18: Average Lyapunov functions plots with $r = 6$ for 100 Monte Carlo simulations

The opportunistic triggering nature of the event-triggered protocol makes it appealing for conceptual and design purpose as compared with periodic sampling. However, the quantitative performance specifications are surprisingly yet to be explored for different triggering based protocols. The choice of the triggering protocol to be implemented is task specific. For example, in some applications, it may be desirable to reduce the triggering instants as much as possible while in other cases, it may be required to get faster convergence without paying much attention on frequent communications.

Chapter 4

State Consensus in MAS with General Linear Dynamics

In the previous chapter, we discussed average consensus in MAS with single-integrator dynamics. For states corresponding to real physical process we now consider state consensus problem in MAS having general linear dynamics. It is noted that many practical agent systems have dynamics to look beyond simple integrator case. In this chapter, to address this issue, we develop a unified framework for triggering based protocols to solve the state consensus problem in MAS with linear dynamics. Our general approach is to employ triggering protocols that depend on the state values received from neighbors. We start with a static version and then generalize it so that it involves an internal auxiliary variable to regulate the threshold dynamically for each agent. They are referred to as static and dynamic event-triggered protocols and are discussed in Sections 4.2 and 4.3, respectively. The third protocol is based of self-triggering control, where agents determine its next transmission time and send it along with the state information at current triggering times. This one will be introduced in Section 4.4. A numerical example is provided to demonstrate the performances of the proposed protocols. This part is published in [65].

4.1 Problem Formulation

In this section, we present multi-agent systems for state consensus. For the preliminaries on graph theory we refer to the section 3.1.1.

4.1.1 System Model

We consider the multi-agent system of N agents interacting over the network given by the connected graph $\mathcal{G}$. The agents are modeled by the following linear dynamics in discrete time:

$$x_i(k+1) = Ax_i(k) + Bu_i(k), \tag{4.1}$$

where $x_i(k) \in \mathbb{R}^n$ is the state and $u_i(k) \in \mathbb{R}$ is the control input associated with agent i at time step $k \in \mathbb{Z}_+$. Here, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^n$ are, respectively, the system and input matrices of (4.1). We assume that the matrix pair (A, B) is controllable.

Each agent i is equipped with the distributed control protocol of the following form:

$$u_i(k) = -K \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i(k) - \hat{x}_j(k)). \tag{4.2}$$

Here, $K \in \mathbb{R}^{1\times n}$ is the feedback gain matrix. The last broadcast state of agent i at time step k is denoted by $\hat{x}_i(k)$ and more specifically it is given by

$$\hat{x}_i(k) = x_i(t_l^i), \quad k \in [t_l^i, t_{l+1}^i),$$

where $t_0^i, t_1^i, \ldots \in \mathbb{Z}_+$ represent the sequence of event times of agent i. Note that the latest broadcast state $\hat{x}_j(k)$ of neighboring agent $j \in \mathcal{N}_i$ appears in the control protocol (4.2) for agent i. This is because agent i has access to only the last broadcasted states $\hat{x}_j(k)$ of its neighbors instead of their true states $x_j(k)$. As a consequence, in this setting, the control inputs are constant between the triggering times.

The objective of this study is to design event-triggering control protocols so that the N agents can reach state consensus asymptotically as $\lim_{k\to\infty} \|x_i(k) - x_j(k)\| = 0$ for all $i, j \in \mathcal{V}$. That is, the entire states of all agents must coincide eventually. Note that this notion of consensus does not constrain the exact value of the states. Thus it is possible that the state values may not converge to a particular state, and remain to change over time even after they take the same values.

It is known from [96] that in the nominal case when the original state and the broadcast state match at all times as $\hat{x}_i(k) = x_i(k)$ for $i \in \mathcal{V}$ and k, a necessary and sufficient condition to achieve state consensus is that there exists a feedback gain matrix K such that the spectral radius of

$$J = \operatorname{diag}(A - \lambda_2 BK, \ldots, A - \lambda_N BK) \tag{4.3}$$

satisfies $\rho(J) < 1$. We assume that such a matrix K exists.

We now rewrite the agent system in a matrix form. Let $x(k) = [x_1(k) \cdots x_N(k)]^T$, $\hat{x}(k) = [\hat{x}_1(k) \cdots \hat{x}_N(k)]^T$ and let

$$e(k) = \hat{x}(k) - x(k)$$

be the vector of errors in the broadcast states. We can rewrite the closed-loop system from (4.1) and (4.2) in compact form as

$$x(k+1) = (I_N \otimes A - L \otimes BK)x(k) - (L \otimes BK)e(k). \tag{4.4}$$

We denote the state average of the agent as $\overline{x}(k) = \frac{1}{N}\sum_{i\in\mathcal{V}} x_i(k)$ and the disagreement vector as $\delta_i(k) = x_i(k) - \overline{x}(k)$. In compact form, we have

$$\delta(k) = (\Upsilon \otimes I_n)x(k), \tag{4.5}$$

where $\delta(k) = [\delta_1(k) \cdots \delta_N(k)]$ and $\Upsilon = I_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$.

Now, we provide the design procedure for the control law. First, for the Laplacian, we take its left eigenvectors as ϕ_i, that is, they satisfy $\phi_i^T L = \lambda_i \phi_i^T$ for $i \in \mathcal{V}$. Then let $\Phi = [\phi_2 \cdots \phi_N]$ and moreover let $U = [\mathbf{1}_N/\sqrt{N}, \Phi] \in \mathbb{R}^{N\times N}$, which is an orthogonal matrix. The following properties hold:

$$U^T U = I_N, \quad L\Upsilon = \Upsilon L = L \ \text{ and } \ U^T L U = \text{diag}(0, \lambda_2, \ldots, \lambda_N). \tag{4.6}$$

The following modified algebraic Riccati inequality from [96] in the discrete-time domain is solved to obtain a positive-definite matrix $P > 0$:

$$P - A^T P A + (1-\zeta^2)A^T PB(B^T PB)^{-1}B^T PA = Q > 0, \tag{4.7}$$

where the parameter $\zeta > 0$ is chosen such that

$$\max\left\{\left(\frac{\lambda_N - \lambda_2}{\lambda_N + \lambda_2}\right)^2,\ 1 - 2\lambda_2\right\} \leq \zeta^2 < \frac{1}{\prod_j |\lambda_j^u(A)|^2} \tag{4.8}$$

and $\lambda_j^u(A)$ denotes the unstable eigenvalues of A. Furthermore, let the controller gain be

$$K = \frac{2B^T PA}{(\lambda_2 + \lambda_N)B^T PB}. \tag{4.9}$$

The design parameter Ω is chosen as $\Omega \in (0, 1)$ to ensure asymptotic consensus, where

$$\Omega = \frac{1}{\lambda_{\max}(P)}[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1 + \kappa_1\lambda_N^2)] \tag{4.10}$$

and

$$\kappa_0 = \|A^T PBK\|, \quad \kappa_1 = \|K^T B^T PBK\|. \tag{4.11}$$

We next introduce three triggering based protocols with different characteristics and advantages in terms of necessary computational resources and capabilities in reducing the frequency of communications and updates for each agent.

4.2 Protocol 1: Static Event Triggered

In this section, we develop an asynchronous triggering protocol that dictates when agents should broadcast their state information to neighboring agents so that the agents achieve state consensus.

The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$. Agent i determines its triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min\Big\{k > t_l^i : \|\hat{x}_i(k-1) - x_i(k)\|^2 \geq \alpha_i(k)\hat{q}_i(k-1)\Big\}, \tag{4.12}$$

where the threshold is a product of two variables. The first variable $\alpha_i(k)$ takes nonnegative values and exponentially decreases to zero. The other is $\hat{q}_i(k)$ given by

$$\hat{q}_i(k) = \min\left\{\frac{1}{2}\sum_{j\in\mathcal{N}_i} a_{ij}\|\hat{x}_j(t^j_{l_j}) - \hat{x}_i(t^i_{l_i})\|^2, M\right\} \geq 0, \tag{4.13}$$

where M is a positive constant. This M enables us to keep $\hat{q}_i(k)$ bounded by a known value. The variable $\hat{q}_i(k)$ depends on local consensus error. The threshold in (4.12) becomes large when the agent is far from consensus with its neighbors and hence the control need not be very accurate. This helps in reducing triggering instants without sacrificing convergence performance. Notice that by (4.12) it holds

$$\|e_i(k)\|^2 \leq \alpha_i(k)\hat{q}_i(k-1). \tag{4.14}$$

The triggering protocol in (4.12) is said to be static since it involves only the agent states $x_i(k)$, $\hat{x}_i(k)$, and $\hat{x}_j(k)$ and there is no extra variable. The triggering law is fully distributed since each agent requires only its own and its neighbors' state information.

We would like to present the main result for the static event-triggering protocol.

Theorem 4.2.1. *Consider the multi-agent system* (4.1) *whose underlying graph* $\mathcal{G}$ *is connected. Under the static event-triggering control protocol* (4.2) *and* (4.12), *the agents achieve state consensus asymptotically.*

Proof. Consider the Lyapunov candidate given by

$$V(k) = \delta^T(k)(I_N \otimes P)\delta(k). \tag{4.15}$$

We now look at the difference between its values at times $k+1$ and k as

$$\begin{aligned}\nabla V(k) &= V(k+1) - V(k) \\ &= \delta^T(k+1)(I_N \otimes P)\delta(k+1) - \delta^T(k)(I_N \otimes P)\delta(k).\end{aligned}$$

We drop the time index k to simplify the notation. Then by (4.4),

$$\begin{aligned}\nabla V &= [(I_N \otimes A - L \otimes BK)\delta - (L \otimes BK)e]^T(I_N \otimes P) \\ &\quad \cdot [(I_N \otimes A - L \otimes BK)\delta - (L \otimes BK)e] - \delta^T(I_N \otimes P)\delta \\ &= \delta^T[I_N \otimes (A^TPA - P)]\delta - 2\delta^T(L \otimes A^TPBK)\delta - 2\delta^T(L \otimes A^TPBK)e \\ &\quad + \delta^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)\delta + 2\delta^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)e \\ &\quad + e^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)e. \end{aligned} \tag{4.16}$$

By (4.6), we bound the first and second terms of the far right-hand side of (4.16) as

$$\delta^T[I_N \otimes (A^TPA - P) - 2(L \otimes A^TPBK)]\delta \leq \sum_{i=2}^{N} \tilde{\delta}_i^T[A^TPA - P - 2\lambda_i A^TPBK]\tilde{\delta}_i,$$

where $\tilde{\delta} = (U^T \otimes I_n)\delta$. Note that by (4.5) and the definition of U, it holds $\tilde{\delta}_1 = 0$. From (4.8) and (4.9), we have

$$\begin{aligned}&\delta^T[I_N \otimes (A^TPA - P) - 2(L \otimes A^TPBK)]\delta \\ &\leq \sum_{i=2}^{N} \tilde{\delta}_i^T[A^TPA - P - (1-\zeta^2)\frac{A^TPB}{B^TPB}B^TPA]\tilde{\delta}_i.\end{aligned}$$

We can further rewrite this using (4.7) as

$$\delta^T[I_N \otimes (A^TPA - P) - 2(L \otimes A^TPBK)]\delta \leq -\sum_{i=2}^{N} \tilde{\delta}_i^T Q \tilde{\delta}_i \leq -\lambda_{\min}(Q) \sum_{i=2}^{N} ||\tilde{\delta}_i||^2.$$

We now bound the cross-terms in the third term of (4.16) using Young's inequality: $a^T b \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$ for $a, b \in \mathbb{R}^n$ as

$$-2\delta^T(L \otimes A^TPBK)e \leq \kappa_0^2 \sum_{i=2}^{N} \lambda_i^2 \|\tilde{\delta}_i\|^2 + \sum_{i=1}^{N} \|e_i\|^2 \leq \kappa_0^2 \lambda_N^2 \sum_{i=1}^{N} \|\delta_i\|^2 + \sum_{i=1}^{N} \|e_i\|^2,$$

where κ_0 is given in (4.11). Similarly, we bound the fifth term of (4.16) as

$$\begin{aligned} 2\delta^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)e &\leq \kappa_1^2 \sum_{i=2}^{N} \lambda_i^4 \|\tilde{\delta}_i\|^2 + \sum_{i=1}^{N} \|e_i\|^2 \\ &\leq \kappa_1^2 \lambda_N^4 \sum_{i=1}^{N} \|\delta_i\|^2 + \sum_{i=1}^{N} \|e_i\|^2, \end{aligned}$$

where κ_1 is given in (4.11).

We now bound the fourth term of (4.16) using (4.11) and (4.6) as

$$\delta^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)\delta \leq \kappa_1 \sum_{i=2}^{N} \lambda_i^2 \|\tilde{\delta}_i\|^2 \leq \kappa_1 \lambda_N^2 \sum_{i=1}^{N} \|\delta_i\|^2.$$

Similarly, the last term of (4.16) can be bounded as

$$e^T(L \otimes BK)^T(I_N \otimes P)(L \otimes BK)e \leq \kappa_1 \sum_{i=1}^{N} \lambda_i^2 \|\tilde{e}_i\|^2 \leq \kappa_1 \lambda_N^2 \sum_{i=1}^{N} \|e_i\|^2,$$

where $\tilde{e} = (U^T \otimes I_n)e$.

Combining all the terms obtained above, we have from (4.16)

$$\nabla V \leq -[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 + (2+\kappa_1\lambda_N^2)\sum_{i=1}^{N}\|e_i\|^2. \tag{4.17}$$

From (4.13) and (4.14), we have

$$\nabla V \leq -[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 + (2+\kappa_1\lambda_N^2)\alpha_{\max}(k)NM, \tag{4.18}$$

where $\alpha_{\max}(k) = \max_{i\in\mathcal{V}} \alpha_i(k)$. Hence from (4.10),

$$V(k+1) \leq (1-\Omega)V(k) + \Delta_1(k),$$

where

$$\Delta_1(k) = (2+\kappa_1\lambda_N^2)\alpha_{\max}(k)NM. \tag{4.19}$$

Note that $\Omega \in (0,1)$ from (4.10) and moreover, $\alpha_{\max}(k)$ is exponentially decreasing to zero. Thus, it follows that $V(k) \to 0$ as $k \to \infty$. This implies that consensus is obtained asymptotically by the multi-agent system. □

The proposed event-triggered control can be viewed as an extension of the consensus protocol in [64] from the scalar-integrator agent case to the case of agents with linear dynamics. We also note that our controller design process is based on the approach of [96]; there, it is shown that the solvability of the Riccati equation in (4.7) is a necessary and sufficient condition. We have additional technical assumption on ζ, which is the lower bound expressed as $1 - 2\lambda_2$ in (4.8). This design constraint may have a limited effect especially when the second largest eigenvalue of the Laplacian is large as $\lambda_2 \geq 1/2$.

4.3 Protocol 2: Dynamic Event Triggered

In this section, we extend the static triggering protocol from the previous section to the dynamic version. The intuition behind the dynamic triggering law is to regulate the threshold dynamically for each agent based on the state information of itself and neighbors. With the static triggering law, the threshold associated with the state error becomes smaller and smaller as the system approaches consensus. This may lead to frequent triggering instants. The dynamic triggering law that we propose employs an internal auxiliary state variable for each agent to regulate its threshold. This approach is based on the static triggering protocol (4.12) and is also fully distributed.

Agent i employs an internal dynamic variable χ_i satisfying the update rule given by

$$\chi_i(k+1) = \beta_i \chi_i(k) + \alpha_i(k)\hat{q}_i(k-1) - \|e_i(k)\|^2, \tag{4.20}$$

where the initial value is set as $\chi_i(0) > 0$ and the parameter $\beta_i \in (0,1)$ is to be designed. This variable $\chi_i(k)$ will be used in the dynamic threshold and will help us in reducing triggering frequencies compared with the static-triggering protocol.

The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$. Agent i determines its triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min\left\{k > t_l^i : \theta_i\big[\|\hat{x}_i(k-1) - x_i(k)\|^2 - \alpha_i(k)\hat{q}_i(k-1)\big] \geq \chi_i(k)\right\} \tag{4.21}$$

with $\hat{q}_i(k)$ and $\chi_i(k)$ given in (4.13) and (4.20), respectively. We also take $\theta_i > 0$.

Using (4.21), we can easily show that for all k

$$\theta_i\big(\|e_i(k)\|^2 - \alpha_i(k)\hat{q}_i(k-1)\big) \le \chi_i(k). \tag{4.22}$$

This implies

$$\|e_i(k)\|^2 \le \frac{1}{\theta_i}\big(\chi_i(k) + \theta_i\alpha_i(k)\hat{q}_i(k-1)\big). \tag{4.23}$$

Take the parameter θ_i such that

$$\theta_i > \frac{1}{\beta_i}. \tag{4.24}$$

On the other hand, we can show that $\chi_i(k)$ is positive for all k. This is because from (4.20) and (4.22), we obtain

$$\chi_i(k+1) \ge \left(\beta_i - \frac{1}{\theta_i}\right)\chi_i(k).$$

Thus, by (4.24), it follows

$$\chi_i(k) \ge \left(\beta_i - \frac{1}{\theta_i}\right)^k \chi_i(0) > 0. \tag{4.25}$$

We should highlight that the dynamic triggering protocol (4.21) is a generalization of the static triggering protocol (4.12). In particular, by taking the initial values of the auxiliary states $\chi_i(0)$ to be zero (and hence $\chi_i(k) = 0$ at all times), the two protocols coincide. Further, by taking larger positive values for $\chi_i(0)$, the threshold remains large and thus the number of triggering instants can be reduced with the dynamic event-triggering protocol. This can be seen from (4.21) and (4.25).

The main result for the dynamic event triggered protocol is stated below.

Theorem 4.3.1. *Consider the multi-agent system* (4.1) *whose underlying graph* $\mathcal{G}$ *is connected. Under the dynamic event-triggering control protocol* (4.2) *and* (4.21)*, the agents achieve state consensus asymptotically.*

Proof. This proof follows similar steps as in that of Theorem 4.2.1. We use the same Lyapunov candidate from (4.15) and upper bound its time difference as in (4.17). Then, we obtain

$$\nabla V \leq -[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 + (2+\kappa_1\lambda_N^2)\sum_{i=1}^{N}\|e_i\|^2.$$

The inequality in (4.23) implies

$$\begin{aligned}\nabla V \leq &-[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 \\ &+ (2+\kappa_1\lambda_N^2)\sum_{i=1}^{N}\alpha_i(k)\hat{q}_i(k-1) + \left(\frac{2+\kappa_1\lambda_N^2}{\min_i\theta_i}\right)\sum_{i=1}^{N}\chi_i\end{aligned}$$

From (4.19), we can show

$$\nabla V \leq -[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 + \Delta_1(k) + \kappa_2\sum_{i=1}^{N}\chi_i,$$

where $\kappa_2 = \frac{2+\kappa_1\lambda_N^2}{\min_i\theta_i}$. As we have seen earlier, from the form given in (4.19), $\Delta_1(k)$ exponentially converges to zero. By (4.10),

$$\nabla V \leq -\Omega V(k) + \Delta_1(k) + \kappa_2\sum_{i=1}^{N}\chi_i(k). \tag{4.26}$$

Now, by (4.20), (4.23) and (4.25)

$$\chi_i(k+1) \leq \left(\beta_i - \frac{1}{\theta_i}\right)\chi_i(k) + 2\alpha_i(k)M.$$

Therefore, we can bound $\chi_i(k)$, which goes to zero by (4.24). Hence, from (4.26), it follows that $V(k) \to 0$ as $k \to \infty$. This implies that consensus is obtained asymptotically by the multi-agent system. $\square$

4.4 Protocol 3: Self Triggered

As mentioned in section 3.4, one weakness of event-triggered schemes is that the agents require constant monitoring of the current state of the system to determine control update instants. In this section, we present self-triggered control strategy wherein each agent only needs to make updates/broadcasts at its own triggering instants and listen to its neighbors at their triggering instants. To this end, each agent must find an estimate of its future state at the time of making a transmission. We present how this can be done.

First, recall that the agents update their states based on (4.1) and (4.2) as

$$x_i(k+1) = Ax_i(k) - BK \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_i(k) - \hat{x}_j(k)). \tag{4.27}$$

Because of the event-based protocol, the state of agent i at step $k \in [t_l^i, t_{l+1}^i)$ can be expressed as

$$x_i(k) = A^{k-t_l^i}\, x_i(t_l^i) - \sum_{l=t_l^i}^{k-1} A^{k-1-l} BK \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_i(l) - \hat{x}_j(l)),$$

where t_l^i is the latest triggering time of agent i. By definition of the triggering times t_l^i for agent i, we have $x_i(t_l^i) = \hat{x}_i(k)$. Thus, the error $e_i(k) = \hat{x}_i(k) - x_i(k)$

can be expressed as

$$e_i(k) = \sum_{l=t_l^i}^{k-1} A^{k-1-l} BK \sum_{j \in \mathcal{N}_i} a_{ij}\ u_{ij}(k) + \left[I - A^{k-t_l^i} \right] x_i(t_l^i)$$

where we set $u_{ij}(k)$ as the difference in the broadcast values of agents i and j:

$$u_{ij}(k) = \hat{x}_i(k) - \hat{x}_j(k).$$

We need to emphasize here that $u_{ij}(k)$ may not be constant for all $k \in [t_l^i, t_{l+1}^i)$ since neighbors may trigger any time in this interval. Consequently, it is hard to know the value of $e_i(k)$ in the interval $k \in (t_l^i, t_{l+1}^i)$. To proceed to find some estimate on $e_i(k)$, we next look for an upper bound of $u_{ij}(k)$. Then the next triggering instant t_{l+1}^i can be determined at current triggering instant t_l^i accordingly.

Our approach here is to ensure that the error $e_i(k)$ decreases exponentially as

$$\left\| e_i(k) \right\| \leq \eta\ \gamma^{\frac{k}{2}}, \tag{4.28}$$

where $\eta > 0$ and $\gamma \in (0, 1)$.

We now provide the protocol based on self-triggered control. First, the design parameter Ψ is chosen as $\Psi \in (0, 1)$, where

$$\Psi = 1 - \Omega. \tag{4.29}$$

Also let

$$\Gamma = (2 + \kappa_1 \lambda_N^2) N \eta^2. \tag{4.30}$$

Furthermore, we introduce the following function to be used by agent i:

$$g_i(k) = \left\| \left[I - A^{k-t_l^i} \right] x_i(t_l^i) \right\| + \left\| \sum_{l=t_l^i}^{k-1} A^{k-1-l} BK \sum_{j \in \mathcal{N}_i} (t_{ij}^1 - t_l^i)\ a_{ij}\ u_{ij}(t_l^i) \right\|$$
$$+ \left\| \sum_{l=t_l^i}^{k-1} A^{k-1-l} \sum_{j \in \mathcal{N}_i} a_{ij} \sum_{m=t_{l+1}^j}^{t_{ij}^2 - 1} 2\eta\ \gamma^{\frac{m}{2}} + f(m) \right\|, \tag{4.31}$$

where

$$t_{ij}^1(k) = \min\{k, t_{l_j+1}^j\} \text{ and } t_{ij}^2(k) = \max\{k, t_{l_j+1}^j\} \text{ for } k \in [t_{l_i}^i, t_{l_i+1}^i) \text{ and} \tag{4.32}$$

$$f(m) = \left[\frac{2}{\lambda_{\min}(P)} \left(\Psi^m\ V_{\max} + \frac{\Psi^m - \gamma^m}{\Psi - \gamma} \Gamma \right) \right]^{1/2}. \tag{4.33}$$

The parameter $V_{\max}$ in (4.33) must be chosen so as to upper bound the initial value of the Lyapunov candidate as $V(0) \le V_{\max}$. Such a bound can be found in practice if the agents' state values are known to start in a certain interval.

The first triggering instants for all agents are chosen to be the initial time as $t_0^i = 0$ for the self-triggering algorithm. Agent i determines its next triggering instants $\{t_l^i\}_{l=1}^{\infty}$ by the triggering law

$$t_{l+1}^i = \min \left\{ k > t_l^i : g_i(k) > \eta\ \gamma^{\frac{k}{2}} \right\}. \tag{4.34}$$

We must note that in this triggering law, $g_i(k)$ in (4.31) increases while $\gamma^{k/2}$ decreases with respect to $k \in [t_l^i, t_{l+1}^i)$. Hence, given the past triggering instant t_l^i agent i can determine the next one t_{l+1}^i by solving (4.34).

We now propose the self-triggered protocol to solve state consensus problem. Its capability to achieve state consensus is guaranteed by the following theorem.

Theorem 4.4.1. *Consider the multi-agent system* (4.1) *whose underlying graph* $\mathcal{G}$ *is connected. Under the self-triggering control protocol* (4.34)*, the agents achieve state consensus asymptotically.*

Proof. We again use the Lyapunov candidate in (4.15). Its time difference can be upper bounded as in (4.17) in the proof of Theorem 4.2.1:

$$\nabla V \leq -[\lambda_{\min}(Q) - \kappa_0^2\lambda_N^2 - \kappa_1\lambda_N^2(1+\kappa_1\lambda_N^2)]\sum_{i=1}^{N}\|\delta_i\|^2 + (2+\kappa_1\lambda_N^2)\sum_{i=1}^{N}\|e_i\|^2.$$

We can further bound ∇V using (4.28) as

$$\nabla V \leq -\Omega\ V(k) + (2+\kappa_1\lambda_N^2)N\eta^2\gamma^k.$$

From (4.29) and (4.30), we have

$$V(k+1) \leq \Psi\ V(k) + \Gamma\gamma^k.$$

By iteration we can write

$$V(k) \leq \Psi^k\ V_{\max} + \sum_{l=0}^{k-1}\Psi^{k-l-1}\ \Gamma\gamma^l = \Psi^k\ V_{\max} + \frac{\Psi^k - \gamma^k}{\Psi - \gamma}\Gamma.$$

We now find an upper bound on $\|x_i(k) - x_j(k)\|$. This is done using (4.33) as

$$\begin{aligned}
\|x_i(k) - x_j(k)\| &\leq \|x_i(k) - \overline{x}(k)\| + \|x_j(k) - \overline{x}(k)\| \\
&\leq \sqrt{2\Big(\|x_i(k) - \overline{x}(k)\|^2 + \|x_j(k) - \overline{x}(k)\|^2\Big)} \\
&\leq \left[\frac{2}{\lambda_{\min}(P)}\Big(\Psi^k\ V_{\max} + \frac{\Psi^k - \gamma^k}{\Psi - \gamma}\Gamma\Big)\right]^{1/2} \\
&= f(k).
\end{aligned}$$

We thus obtain an upper bound of $\|u_{ij}(k)\|$ as

$$\begin{aligned}\|u_{ij}(k)\| &= \|\hat{x}_i(k) - \hat{x}_j(k)\| \\ &\leq \|\hat{x}_i(k) - x_i(k)\| + \|x_i(k) - x_j(k)\| + \|x_j(k) - \hat{x}_j(k)\| \\ &\leq 2\eta\ \gamma^{\frac{k}{2}} + f(k). \end{aligned} \tag{4.35}$$

Next, we proceed to find an upper bound of $\|e_i(k)\|$. At time t_l^i, agent i already knows t_l^j, $x_j(t_l^j)$, and t_{l+1}^j for $j \in \mathcal{N}_i$. We know that $u_{ij}(k)$ is constant for $k \in [t_l^i, t_{ij}^1]$ where t_{ij}^1 is given in (4.32), and for $k > t_{ij}^1$, it can be upper bounded by (4.35). Hence, we have

$$\|e_i(k)\| \leq \left\| \left[I - A^{k-t_l^i} \right] x_i(t_l^i) \right\| + \left\| \sum_{l=t_l^i}^{k-1} A^{k-1-l} BK \sum_{j\in\mathcal{N}_i} a_{ij} u_{ij}(k) \right\| \leq g_i(k), k \in [t_l^i, t_{l+1}^i),$$

that is, $g_i(k)$ bounds $\|e_i(k)\|$ from above. Hence, under the self-triggering law of (4.34), it holds that

$$\|e_i(k)\| \leq \eta\ \gamma^{\frac{k}{2}},\ \ \forall k \in [t_l^i, t_{l+1}^i).$$

Therefore, all state errors decrease exponentially. This indicates that the multi-agent system following the self-triggered control law achieves state consensus asymptotically. □

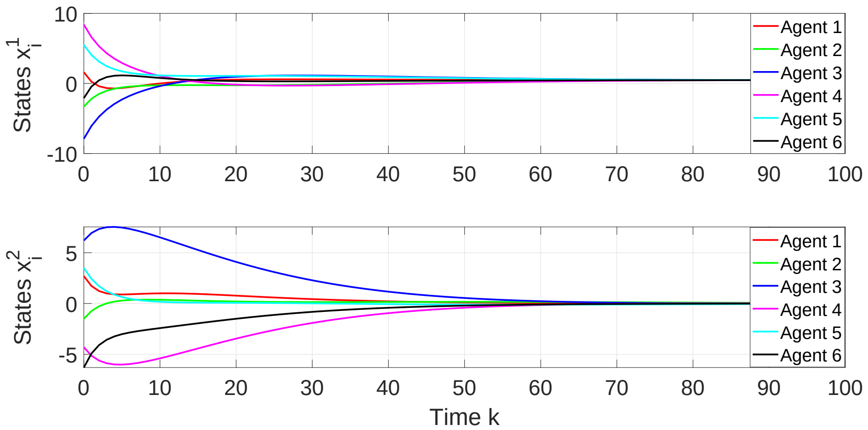

Figure 4.1: State evolutions under the nominal control (4.2)

4.5 Numerical Example

We consider the same six-agent network shown as Fig. 3.2 in Section 3.5. Each agent has the system and input matrices of the form

$$A = \begin{bmatrix} 1 & 0.0593 \\ 0 & 0.9763 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0489 \\ 0.0296 \end{bmatrix}.$$

The controller gain obtained from (4.9) is $K = [5.75\ 1.95]$. In the simulations, we start with the nominal protocol with computation at every step k (i.e., we use (4.2) after replacing $\hat{x}_i(k)$ with $x_i(k)$ for all $i \in \mathcal{V}$). We next illustrate the performances of the proposed static, dynamic and self-triggering protocols; to focus on the comparison of the number of events triggered, we choose their parameters so that the convergence rates for the three protocols are similar.

In the first part of the simulations, we fixed the initial states as

$$x_1(0) = \begin{bmatrix} 1.6 \\ 2.7 \end{bmatrix}, \ x_2(0) = \begin{bmatrix} -3.3 \\ -1.5 \end{bmatrix}, \ x_3(0) = \begin{bmatrix} -7.9 \\ 6.2 \end{bmatrix}, \ x_4(0) = \begin{bmatrix} 8.4 \\ -4.3 \end{bmatrix},$$

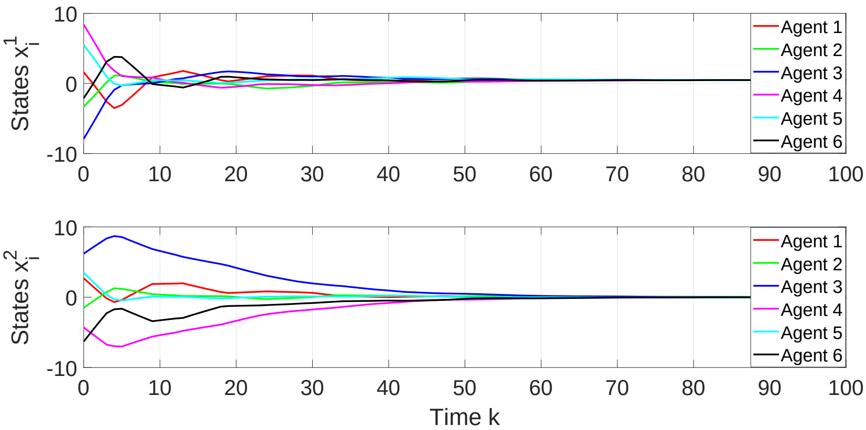

Figure 4.2: State evolutions under the static triggering law (4.12)

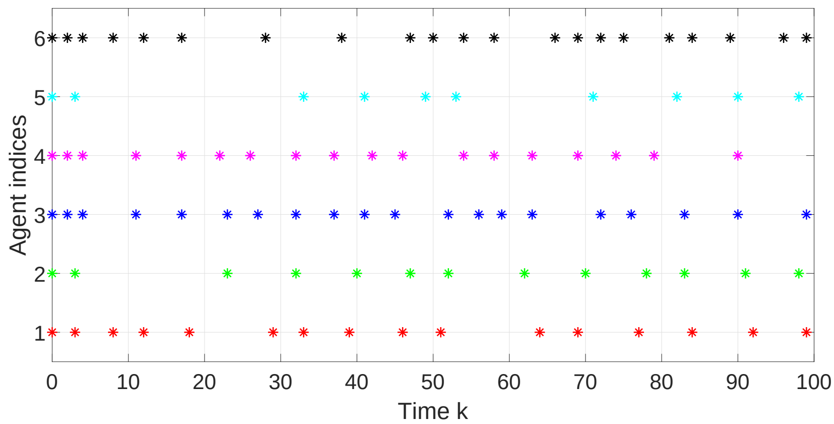

Figure 4.3: Triggering instants under the static triggering law (4.12)

$$x_5(0) = \begin{bmatrix} 5.5 \\ 3.5 \end{bmatrix}, \; x_6(0) = \begin{bmatrix} -2.1 \\ -6.3 \end{bmatrix}$$

and obtained the state evolutions. Fig. 4.1 shows the result under the nominal protocol. State consensus is achieved by all the agents in the network. For the static event-triggering protocol (4.12), we set the variable $\alpha_i = 0.1$. The state responses of agents are presented in Fig. 4.2 and the corresponding triggering instants for each agent are shown in Fig. 4.3. The average number of triggered events per agent is 16.33 times in 100 steps. The state evolutions roughly follow the same convergence rates as the nominal protocol.

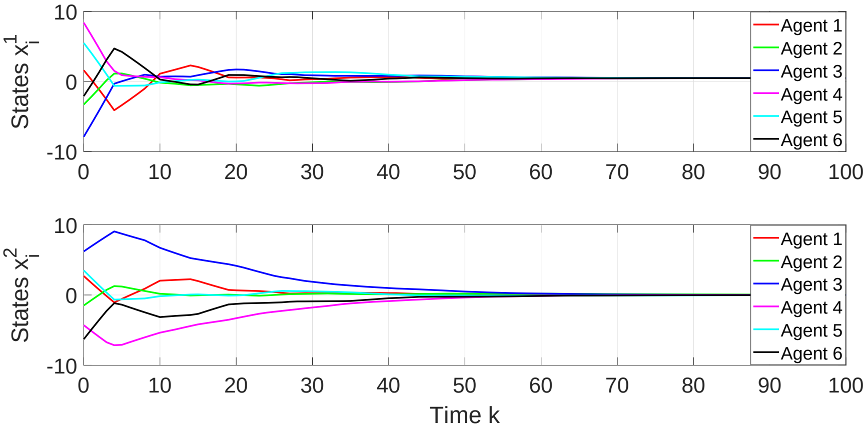

Figure 4.4: State evolutions under the state-independent thresholds

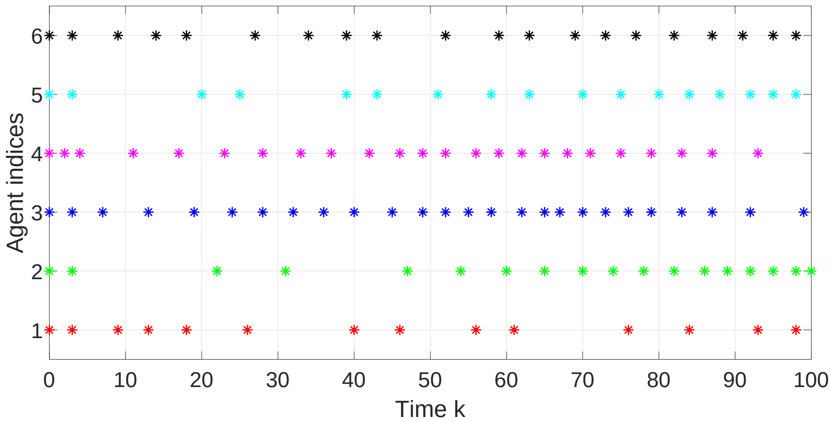

Figure 4.5: Triggering instants under the state-independent thresholds

We now make comparisons with conventional case, i.e., event-triggered control with state independent threshold. We set the threshold value as $0.1\mathrm{e}^{-.01k}$. The state evolution and the corresponding triggering instants for each agents are shown in Figs. 4.4 and 4.5, respectively. It is clear that this protocol demonstrates similar convergence rates but with more transmissions.

Next, we examine the dynamic event-triggering protocol (4.21). The value of the common parameter α_i is taken the same as above. We chose the design parameters $\beta_i = .85$ and $\theta_i = 20$. The state evolutions and the triggering instants

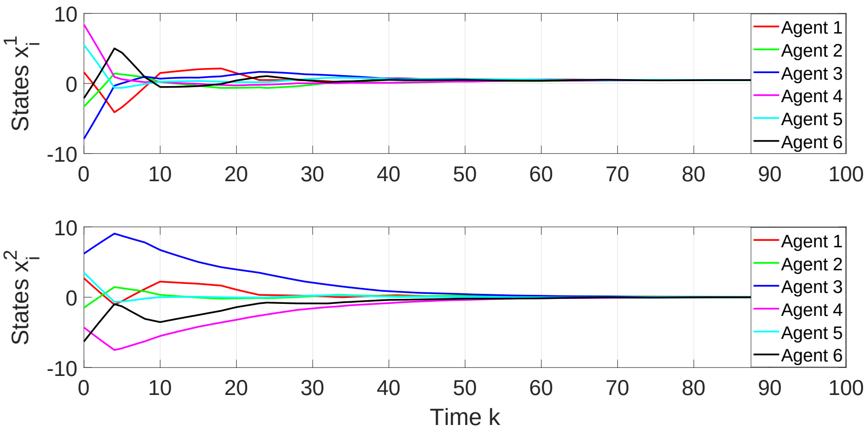

Figure 4.6: State evolutions under the dynamic triggering law (4.21)

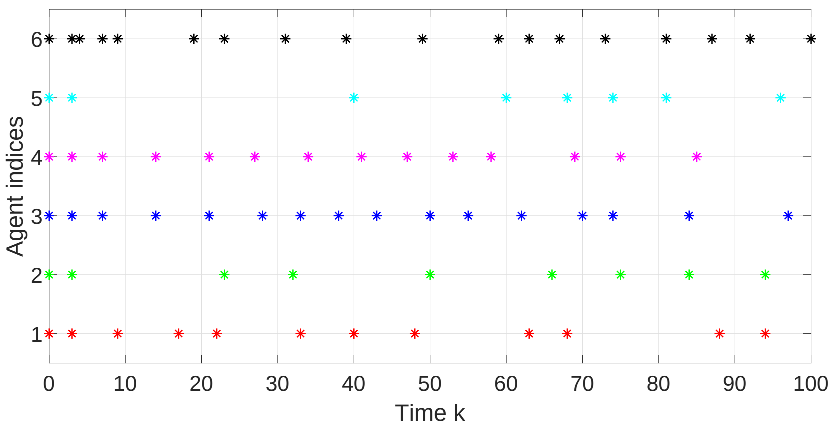

Figure 4.7: Triggering instants under the dynamic triggering law (4.21)

for each agent are shown in Figs. 4.6 and 4.7, respectively. Compared to the static triggering law, the dynamic one generates less triggering instants. This can be seen from Figs. 4.3 and 4.7. The average number of triggered events per agent is 12.83 times in 100 steps. From these results, we observe that the frequency of events can be reduced by using event-triggering protocols and especially the dynamic version.

Third, we discuss the self-triggering protocol (4.34). We selected the parameters $\eta = 20$ and $\gamma = .99$. The state evolution with self-triggered control law is

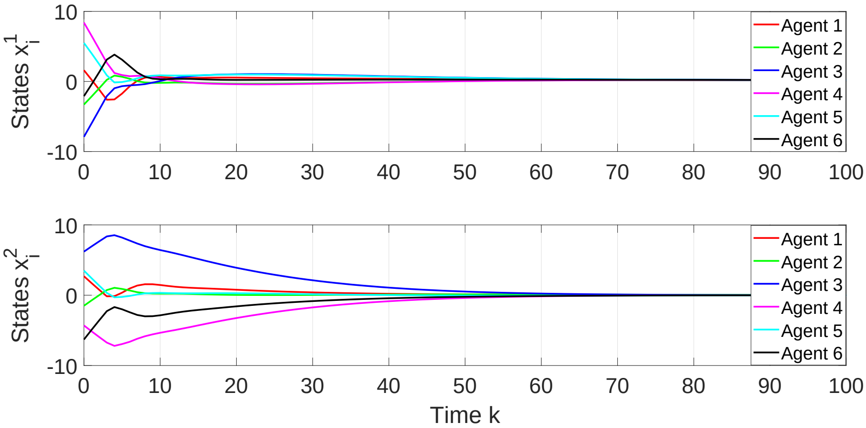

Figure 4.8: State evolutions under the self triggering law (4.34)

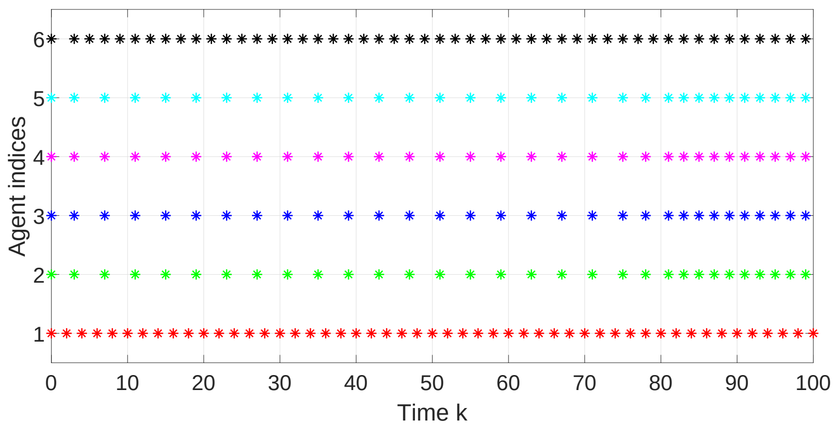

Figure 4.9: Triggering instants under the self triggering law (4.34)

presented in Fig. 4.8 and the corresponding triggering times for each agent are shown in Fig. 4.9. We observe that with the self-triggered control, the agents arrive at state consensus with about the same convergence performance as the other two proposed protocols. However, it is evident that the number of triggering instants are much higher.

In summary, we observe that the proposed protocols perform well in achieving state consensus at convergence rates similar to the nominal control law with different characteristics in terms of transmission frequencies and necessary computational resources at agent levels. The dynamic triggering protocol is particularly effective in reducing the number of triggering instants.

4.6 Formation Control Example

In this section, we discuss the event-triggered control protocol to solve the formation control problem of MAS. We consider the agents with double-integrator dynamics in discrete time given as

$$\begin{aligned} p_i(k+1) &= p_i(k) + \tau_s v_i(k) + \frac{1}{2}\tau_s^2 u_i(k), \\ v_i(k+1) &= v_i(k) + \tau_s u_i(k), \quad i \in \mathcal{V}, \end{aligned} \tag{4.36}$$

where $p_i(k) \in \mathbb{R}$ and $v_i(k) \in \mathbb{R}$ are the position and velocity, respectively. The control input is denoted by $u_i(k) \in \mathbb{R}$ and sampling period by $\tau_s > 0$. We denote the variable $\zeta_i(k) = [p_i(k),\ v_i(k)]^T$. The agent dynamics (4.36) can be written in vector form as

$$\zeta_i(k+1) = \begin{bmatrix} 1 & \tau_s \\ 0 & 1 \end{bmatrix} \zeta_i(k) + \begin{bmatrix} \frac{1}{2}\tau_s^2 \\ \tau_s \end{bmatrix} u_i(k), \quad i \in \mathcal{V} \tag{4.37}$$

It is known from [96] that the control protocol of the following form solves the second-order consensus problem

$$u_i(k) = -K \sum_{j \in \mathcal{N}_i} a_{ij} \big(\zeta_i(k) - \zeta_j(k)\big), \tag{4.38}$$

where $K \in \mathbb{R}^{1\times 2}$ is the feedback gain matrix.

We next discuss the formation control problem of MAS. In order to maintain a fixed formation structure the velocity of all agents must be in consensus. Given the formation vector $\bar{H} = [h_1, \ldots, h_N]^T \in \mathbb{R}^N$, the agent system (4.37) is said to reach formation [50] for any initial states $\zeta_i(0)$ if the following conditions hold:

$$\lim_{k\to\infty} \|(p_i(k) - h_i) - (p_j(k) - h_j)\| = 0,$$
$$\lim_{k\to\infty} \|v_i(k) - v_j(k)\| = 0, \quad i, j \in \mathcal{V}. \tag{4.39}$$

The formation control problem can be solved for $H_i = [h_i \ 0]^T$ by modifying the control protocol (4.38) as (see, e.g.,[96])

$$u_i(k) = -K \sum_{j\in\mathcal{N}_i} a_{ij}\big((\zeta_i(k) - H_i) - (\zeta_j(k) - H_j)\big). \tag{4.40}$$

The second-order consensus problem can be solved only for the formation structures satisfying $(A - I)(H_i - H_j) = 0$, i.e., the velocity of all agents must reach the same value asymptotically.

It can be seen that in the control protocol (4.40) agents require exact state information of neighboring agents to update the control input $u_i(k)$ at every step k. We now present the event-triggered formation control protocol that can better balance between utilization of communication resources and control performance. The control protocol (4.40) is modified as

$$u_i(k) = -K \sum_{j\in\mathcal{N}_i} a_{ij}\big((\hat{\zeta}_i(k) - H_i) - (\hat{\zeta}_j(k) - H_j)\big), \tag{4.41}$$

where $\hat{\zeta}_i(k)$ is the latest broadcast state of agent i. The triggering law (4.12) is modified as

$$t_{l+1}^i = \min\Big\{k > t_l^i : \|\hat{\zeta}_i(k-1) - \zeta_i(k)\|^2 \geq \alpha_i(k)\bar{q}_i(k-1)\Big\}, \tag{4.42}$$

where the variable $\bar{q}_i(k)$ is given as

$$\bar{q}_i(k) = \min\left\{\frac{1}{2}\sum_{j\in\mathbb{N}_i} a_{ij}\|\hat{\zeta}_j(t^j_{l_j}) - \hat{\zeta}_i(t^i_{l_i})\|^2, M\right\} \geq 0. \tag{4.43}$$

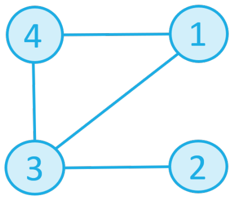

Figure 4.10: Network topology

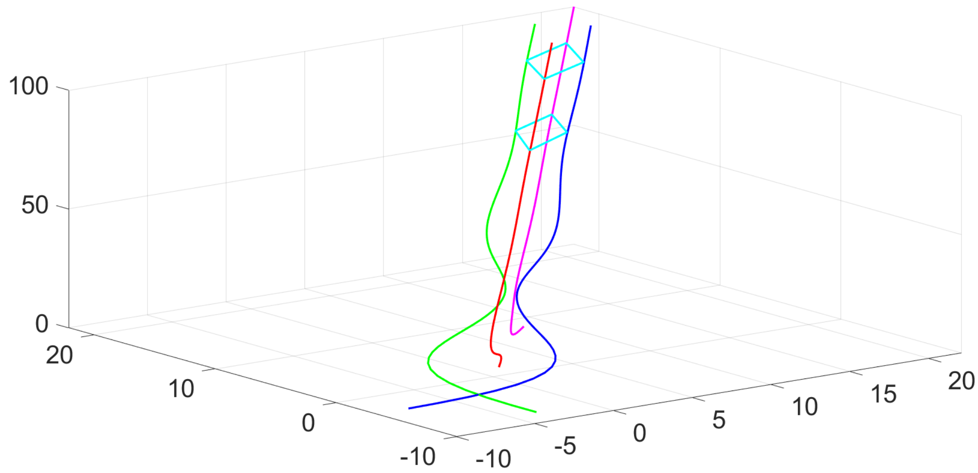

Figure 4.11: Formation control under the protocol (4.40)

We now illustrate a simulation example to show the formation control of the four agent network shown in Fig. 4.10. The system and input matrices with sampling period $\tau_s = .25$ are

$$A = \begin{bmatrix} 1 & 0.25 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.031 \\ 0.25 \end{bmatrix}.$$

The controller gain obtained from (4.9) is $K = [1.42\ 1.77]$. The control objective here is that the agents evolve to form a parallelogram. The state trajectories of agents with protocol (4.40) are shown in Fig. 4.11. We next demonstrate the performance of the event-triggered protocol (4.42). The parameter is set as $\alpha_i = 0.01$. The state evolutions of agents are shown in Fig. 4.12 and their triggering instants are shown in Fig. 4.13. The agents achieve the desired formation structure with the event-triggered protocol (4.42).

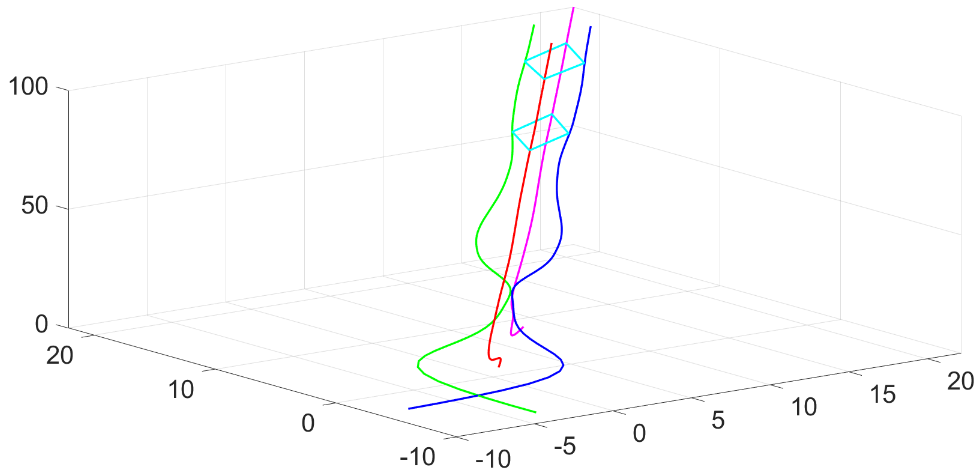

Figure 4.12: Formation control under the protocol (4.41)

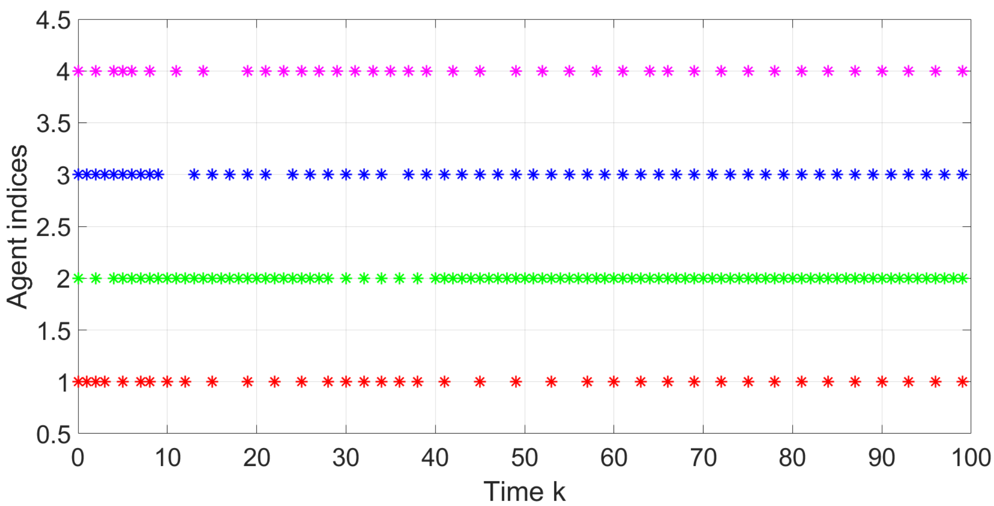

Figure 4.13: Triggering instants under the protocol (4.41)

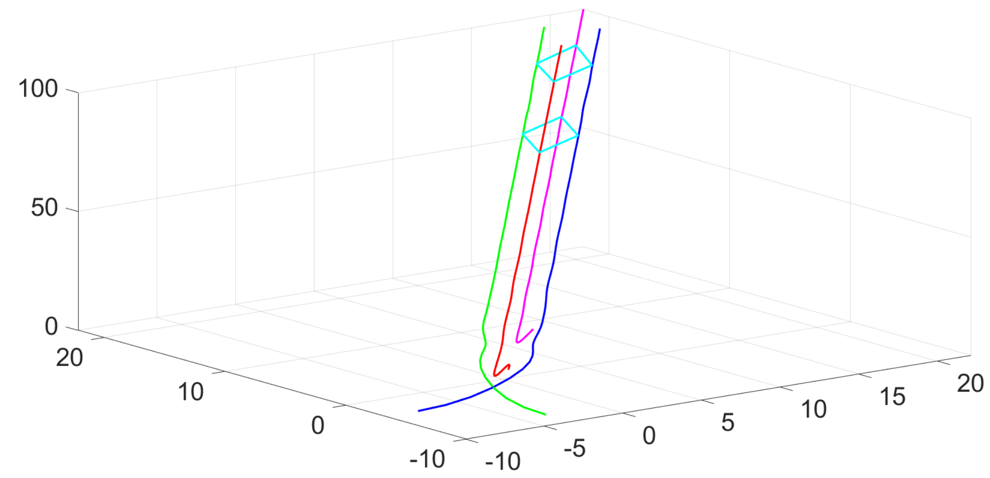

Figure 4.14: Formation control under the protocol (4.41) for the complete network

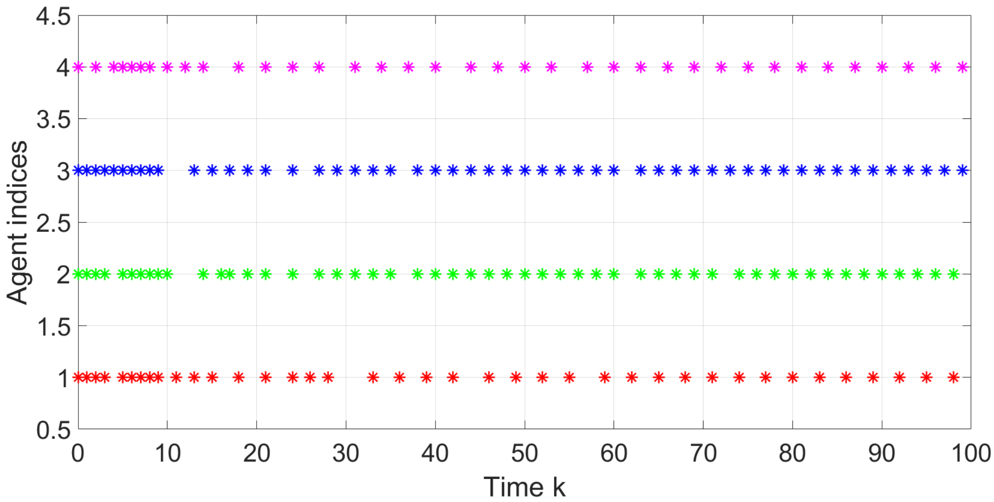

Figure 4.15: Triggering instants under the protocol (4.41) for the complete network

We next demonstrate the performance of the event-triggered protocol for a complete network consisting of four agents. The state evolutions of agents and their triggering times are shown in Figs. 4.14 and 4.15, respectively. The agents achieve the desired formation structure with less number of triggering instants.

Chapter 5

Weighted Consensus in MAS with Directed Topologies

In this chapter, we study the consensus problem in multi-agent networks with directed topologies. This is an extension of the problem setting of the one studied in Chapter 3, which was limited to graphs with undirected topologies. We generalize the three protocols proposed there, namely, the static, dynamic, and self-triggering protocols to solve the weighted consensus problem in strongly connected agent network. Our approach here relies on the triggering protocols that we proposed in Chapter 3, depending on the state values received from neighbors. We begin with a static version of the triggering protocol to effectively reduce transmission and control update requirements for agents. In order to further reduce the number of triggering instants, we employ auxiliary state variable for each agent to regulate the threshold dynamically. The third protocol uses self-triggering control mechanism. Each agent determines its next transmission instant at current triggering times and send it to their neighbors along with the state information. Simulation examples are illustrated to demonstrate the effectiveness of the proposed protocols.

5.1 Problem Formulation

We present preliminaries on graph theory [11] and the multi-agent system for the weighted consensus problem in this section.

5.1.1 Graph Theory

Consider the network described by directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, comprised of the node set $\mathcal{V} = \{1, \ldots, N\}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The edge $(j, i) \in \mathcal{E}$ indicates that node j can send message to node i and is called an incoming edge of node i. The set of (incoming) neighbors of node i is given by $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. Moreover, the weighted degree of agent i is denoted as $d_i = \sum_{j \in \mathcal{N}_i} a_{ji}$. A directed graph $\mathcal{G}$ is strongly connected if there is a directed path from every node to every other node in the graph. It holds that for a strongly connected graph $\mathcal{G}$, the corresponding Laplacian matrix L is irreducible.

We introduce the following lemmas (e.g., [75]) that will be useful in our development.

Lemma 5.1.1. *If L is irreducible, then zero is an algebraically simple eigenvalue of L and there exists a positive vector $\xi = [\xi_1, \cdots, \xi_N]^T$ such that $\xi^T L = 0$ and $\sum_{i \in \mathcal{V}} \xi_i = 1$.*

Lemma 5.1.2. *Suppose that L is irreducible. Let $\Xi = diag[\xi_1, \cdots, \xi_N]$. Then $R = \frac{1}{2}(\Xi L + L^T \Xi)$ is positive semi-definite matrix with row sums equal to zero and has zero eigenvalue with algebraic multiplicity one.*

We denote the eigenvalues of R in increasing order as $0 = \lambda_1 \leq \cdots \leq \lambda_N$. Also, let us denote the matrix $U = \Xi - \xi\xi^T$ and its eigenvalues in increasing order as $0 = \mu_1 \leq \cdots \leq \mu_N$. Then it holds that for all $x \in R^N$ satisfying $x \perp 1$, we have $\lambda_2 x^T x \leq x^T R x \leq \lambda_N x^T x$ and $\mu_2 x^T x \leq x^T U x \leq \mu_N x^T x$.

5.1.2 System Model

Let us consider the agent network comprising of N identical agents. The digraph $\mathcal{G}$ describes the interaction among agents. Each agent $i \in \mathcal{V}$ takes the dynamics in a discrete-time integrator form given as

$$x_i(k+1) = x_i(k) + u_i(k), \tag{5.1}$$

where $x_i(k) \in \mathbb{R}$ is the state and $u_i(k) \in \mathbb{R}$ is the control input for agent i at step $k \in \mathbb{Z}_+$. The agents are said to achieve weighted consensus asymptotically if $x_i(k) \to \sum_j \xi_j x_j(0)$ as $k \to \infty$ for all i. It is well known that consensus can be reached through the common control protocol (see, e.g., [70]) given by

$$u_i(k) = -\epsilon \sum_{j \in \mathcal{N}_i} a_{ij}(x_i(k) - x_j(k)), \tag{5.2}$$

where $0 < \epsilon < 1/d_{\max}$ and $d_{\max} = \max_{i \in \mathcal{V}} d_i$.

Agent i in consensus protocol (5.2) requires exact state information from neighboring agents to update the control input at every step k. This may result in excessive usage of communication and computational resources. We next introduce triggering-based control protocols that can better balance between utilization of communication resources and control performances.

5.2 Protocol 1: Static Event Triggered

In this section, we present the first event-triggered protocol, which employs a state-dependent but static threshold for determining the event times.

In our triggering-based control protocols, the control input for agent i is slightly modified from (5.2) to the form

$$u_i(k) = -\epsilon \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_i(k) - \hat{x}_j(k)). \tag{5.3}$$

Here, $\hat{x}_i(k)$ and $\hat{x}_j(k)$ denote the last broadcast states of agents i and j, respectively.

Let $x(k) = [x_1(k) \cdots x_N(k)]^T$ and $\hat{x}(k) = [\hat{x}_1(k) \cdots \hat{x}_N(k)]^T$. Also, let $e(k) = \hat{x}(k) - x(k)$ be the vector of the errors between the broadcast state and the original state. We now rewrite the agent system (5.1) and (5.3) in the vector form as

$$\begin{aligned} x(k+1) &= x(k) - \epsilon L\hat{x}(k) \\ &= x(k) - \epsilon Lx(k) - \epsilon Le(k). \end{aligned} \tag{5.4}$$

We introduce an asynchronous triggering protocol that specifies the transmission times for the agents to interact so as to seek consensus among neighbors. For all agents, the starting time $t_0^i = 0$ is set as the first triggering instant. The triggering instants $\{t_l^i\}_{l=1}^{\infty}$ for agents are arbitrated via the triggering protocol

$$t_{l+1}^i = \min\Big\{k > t_l^i : \big(\hat{x}_i(k-1) - x_i(k)\big)^2 > \alpha_i(k)\hat{q}_i(k-1)\Big\}. \tag{5.5}$$

Here, the threshold is composed of two parts: The first one is $\alpha_i(k) = \frac{\sigma_i(k)}{2\epsilon d_i}$, where $\sigma_i(k)$ is a function that is exponentially decreasing in time. The second one is state dependent given by

$$\hat{q}_i(k) = \min\left\{\frac{1}{2}\sum_{j \in \mathcal{N}_i} a_{ij}\big(\hat{x}_j(t_l^j) - \hat{x}_i(t_l^i)\big)^2, M\right\} \geq 0, \tag{5.6}$$

where M is a positive scalar, setting a known bound on the threshold. It should be noted that under the triggering protocol (5.5), we have

$$|e_i(k)|^2 \leq \alpha_i(k)\hat{q}_i(k-1). \tag{5.7}$$

We now state our first result for the multi-agent systems under the static event-triggered protocol.

Theorem 5.2.1. *Consider the multi-agent system* (5.1) *with the strongly connected underlying communication digraph* $\mathcal{G}$*. The event-triggered control protocol* (5.3) *and* (5.5) *achieves weighted consensus asymptotically.*

Proof. As a Lyapunov candidate, we employ

$$V(k) = [x(k) - \overline{x}(k)\mathbf{1}_N]^T \Xi [x(k) - \overline{x}(k)\mathbf{1}_N] = x^T(k)Ux(k), \tag{5.8}$$

where the group decision value is the weighted average given as $\overline{x}(k) = \sum_{i\in\mathcal{V}} \xi_i x_i(k)$. It is decided by the left eigenvector ξ associated with $\lambda = 0$ of L. It is easy to check that this value $\overline{x}(k)$ remains constant over time, that is,

$$\overline{x}(k+1) = \xi^T[x(k) - \epsilon L\hat{x}(k)] = \overline{x}(k). \tag{5.9}$$

We now examine the difference between Lyapunov candidate values at times $k+1$ and k given by

$$\nabla V(k) = V(k+1) - V(k).$$

From (5.4) and (5.9), we have

$$\begin{aligned}\nabla V(k) = {} & [x(k) - \epsilon L\hat{x}(k) - \overline{x}(k)\mathbf{1}_N]^T \Xi [x(k) - \epsilon L\hat{x}(k) - \overline{x}(k)\mathbf{1}_N] \\ & - [x(k) - \overline{x}(k)\mathbf{1}_N]^T \Xi [x(k) - \overline{x}(k)\mathbf{1}_N].\end{aligned}$$

For simplicity, we drop the time index k. Then, we get

$$\begin{aligned}\nabla V =& [(x-\overline{x}\mathbf{1}_N)-\epsilon L\hat{x}]^T\Xi[(x-\overline{x}\mathbf{1}_N)-\epsilon L\hat{x}]-(x-\overline{x}\mathbf{1}_N)^T\Xi(x-\overline{x}\mathbf{1}_N)\\ =& -2\epsilon(\hat{x}-e)^T\Xi L\hat{x}+\epsilon^2\hat{x}^TL^T\Xi L\hat{x}\\ =& -2\epsilon\hat{x}^T\Xi L\hat{x}+2\epsilon e^T\Xi L\hat{x}+\epsilon^2\hat{x}^TL^T\Xi L\hat{x}. \end{aligned} \tag{5.10}$$

We next apply the inequality $ab \le a^2+\frac{1}{4}b^2,\ \forall a,b\in\mathbb{R}$, to the second term of (5.10) in order to get its upper bound as

$$\begin{aligned} e^T\Xi L\hat{x} &\le \sum_{i\in V}\sum_{j\in N_i}\xi_i a_{ij}e_i^2+\frac{1}{4}\sum_{i\in V}\sum_{j\in N_i}\xi_i a_{ij}(\hat{x}_i-\hat{x}_j)^2\\ &\le \sum_{i\in\mathcal{V}}\xi_i d_i e_i^2+\frac{1}{2}\hat{x}^T\Xi L\hat{x}. \end{aligned}$$

For the third term in (5.10), using the relation $(\sum_{k=1}^p y_k)^2 \le p\sum_{k=1}^p y_k^2$ for any $y_1,\ldots,y_p\in\mathbb{R}$ and $p\in\mathbb{Z}_+$ from the Cauchy–Schwarz inequality, we have

$$\begin{aligned} \hat{x}^TL^T\Xi L\hat{x} &= \sum_{i\in\mathcal{V}}\xi_i\Big[\sum_{j\in\mathcal{N}_i}a_{ij}(\hat{x}_i-\hat{x}_j)\Big]^2 \le \sum_{i\in\mathcal{V}}d_i\sum_{j\in\mathcal{N}_i}\xi_i a_{ij}^2(\hat{x}_i-\hat{x}_j)^2\\ &\le 2d_{\max}\ a_{\max}\hat{x}^T\Xi L\hat{x}, \end{aligned}$$

where $a_{\max}=\max_{i,j\in\mathcal{V}}a_{ij}$. Then, we get

$$\nabla V \le -(\epsilon-2\epsilon^2 d_{\max}\ a_{\max})\hat{x}^T\Xi L\hat{x}+2\epsilon\sum_{i\in\mathcal{V}}\xi_i d_i e_i^2. \tag{5.11}$$

Now from (5.7) and then (5.6), we have

$$\begin{aligned} \nabla V &\le -(\epsilon-2\epsilon^2 d_{\max}\ a_{\max})\hat{x}^T\Xi L\hat{x}+\sum_{i\in\mathcal{V}}\xi_i\sigma_i(k)\hat{q}_i(k-1)\\ &\le -(\epsilon-2\epsilon^2 d_{\max}\ a_{\max})\hat{x}^T(k)\Xi L\hat{x}(k)+\xi_{\max}\sigma_{\max}(k)NM, \end{aligned} \tag{5.12}$$

where $\xi_{\max} = \max_{i\in\mathcal{V}} \xi_i$ and $\sigma_{\max}(k) = \max_{i\in\mathcal{V}} \sigma_i(k)$. We note that

$$\begin{aligned} x^T \Xi L x &= (\hat{x} - e)^T \Xi L (\hat{x} - e) \leq 2\hat{x}^T \Xi L \hat{x} + 2e^T \Xi L e \\ &\leq 2\hat{x}^T \Xi L \hat{x} + 2\|\Xi L\| \|e\|^2. \end{aligned} \tag{5.13}$$

We now use (5.7) in (5.13) to get

$$\begin{aligned} x^T(k)\Xi L x(k) &\leq 2\hat{x}^T(k)\Xi L \hat{x}(k) + 2\|\Xi L\| \sum_{i\in\mathcal{V}} \alpha_i(k)\hat{q}_i(k-1) \\ &\leq 2\hat{x}^T(k)\Xi L \hat{x}(k) + \frac{\|\Xi L\|\sigma_{\max}(k)NM}{\epsilon d_{\min}}, \end{aligned}$$

where $d_{\min} = \min_{i\in\mathcal{V}} d_i$. Therefore,

$$\hat{x}^T(k)\Xi L \hat{x}(k) \geq \frac{1}{2} x^T(k)\Xi L x(k) - \frac{\|\Xi L\|\sigma_{\max}(k)NM}{2\epsilon d_{\min}}.$$

Using the above relation in (5.12), we get

$$\begin{aligned} \nabla V \leq &-\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max}\ a_{\max})\frac{\lambda_2}{\mu_N} V(k) \\ &+ \sigma_{\max}(k)NM\left(\xi_{\max} + \|\Xi L\| \frac{1 - 2\epsilon d_{\max}\ a_{\max}}{2d_{\min}}\right). \end{aligned}$$

Now let

$$\Omega = \frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max}\ a_{\max})\frac{\lambda_2}{\mu_N}. \tag{5.14}$$

Take the parameter ϵ in (5.3) to be small so that $\Omega \in (0, 1)$ holds. Then, it follows that

$$V(k+1) \leq (1 - \Omega)V(k) + \Delta_1(k),$$

where

$$\Delta_1(k) = \sigma_{\max}(k)NM\left(\xi_{\max} + \|\Xi L\|\frac{1-2\epsilon d_{\max}\ a_{\max}}{2d_{\min}}\right). \tag{5.15}$$

Recall that $\Omega \in (0,1)$ and $\sigma_{\max}(k)$ is exponentially decreasing to zero. As a result, $V(k) \to 0$ as $k \to \infty$. From (5.9), we conclude that weighted consensus can be reached asymptotically over the directed network. □

In this section, we have introduced the static event-triggered update rule for networks having directed topologies. This update rule in fact takes a very similar form as the one for undirected networks studied in Section 3.2. As we will see, the same holds for the other two protocols as well. This aspect may highlight the generality of this framework of the event-triggered protocols. In the proofs of consensus results, however, the Lyapunov functions used are different for the undirected and directed cases: For the former, we use $[x(k) - \overline{x}(k)\mathbf{1}_N]^T[x(k) - \overline{x}(k)\mathbf{1}_N]$ and for the latter $[x(k) - \overline{x}(k)\mathbf{1}_N]^T\Xi[x(k) - \overline{x}(k)\mathbf{1}_N]$. These Lyapunov functions are consistent with those used in the conventional consensus schemes (see, e.g., [11]). It is noticed that for the directed topology case, the eigenvector ξ of the Laplacian L plays a crucial role in the proof though in the protocol itself, there is no need for the agents to have such global information.

5.3 Protocol 2: Dynamic Event Triggered

In this section, we introduce the dynamic event-triggered protocol to further reduce the triggering times for agents. In this protocol, each agent regulates its threshold dynamically by having an auxiliary state variable.

Each agent i entails χ_i that follows the update rule

$$\chi_i(k+1) = \beta_i\chi_i(k) + \sigma_i(k)\hat{q}_i(k-1) - 2\epsilon d_i e_i^2(k), \tag{5.16}$$

where $\chi_i(0) > 0$ and $\beta_i \in (0,1)$.

For all agents, the starting time $t_0^i = 0$ is set as the first triggering instant. The triggering instants $\{t_l^i\}_{l=1}^{\infty}$ for agent i is set via the following triggering protocol:

$$t_{l+1}^i = \min\Big\{k > t_l^i : \theta_i\big[2\epsilon d_i\big(\hat{x}_i(k-1) - x_i(k)\big)^2 - \sigma_i(k)\hat{q}_i(k-1)\big] > \chi_i(k)\Big\}. \tag{5.17}$$

The design parameter is $\theta_i > 0$. It holds from (5.17) that

$$\theta_i\big(2\epsilon d_i e_i^2(k) - \sigma_i(k)\hat{q}_i(k-1)\big) \leq \chi_i(k) \tag{5.18}$$

at all time k, that is,

$$e_i^2(k) \leq \frac{1}{2\epsilon\theta_i d_i}\big(\chi_i(k) + \theta_i\sigma_i(k)\hat{q}_i(k-1)\big). \tag{5.19}$$

We select the parameters θ_i so that

$$\theta_i > \frac{1}{\beta_i}. \tag{5.20}$$

Now from (5.16) and (5.18), we get

$$\chi_i(k+1) \geq \left(\beta_i - \frac{1}{\theta_i}\right)\chi_i(k).$$

The auxiliary variable $\chi_i(k)$ is always positive under (5.20) as

$$\chi_i(k) \geq \left(\beta_i - \frac{1}{\theta_i}\right)^k \chi_i(0) > 0. \tag{5.21}$$

It is apparent from (5.18) that when θ_i is very large, then the dynamic protocol reduces to the static protocol. Larger values for $\chi_i(0)$ at the initial time may result in longer intervals between events. This can be seen from (5.17) and (5.21). Setting $\chi_i(0) = 0$, these protocols coincide. Furthermore, since we select the threshold parameter $\chi_i(0) > 0$, it remains large, which helps in reducing triggering times for agents compared with the static version.

We now present the main result for the dynamic event-triggered protocol.

Theorem 5.3.1. *Consider the multi-agent system* (5.1) *with the strongly connected underlying communication digraph* $\mathcal{G}$. *The event-triggered control protocol* (5.3) *and* (5.17) *achieves weighted consensus asymptotically.*

Proof. We consider the Lyapunov candidate given by (5.8). We follow the similar process as it was done in the proof for Theorem 5.2.1:

$$\nabla V \leq -(\epsilon - 2\epsilon^2 d_{\max}\ a_{\max})\hat{x}^T(k)\Xi L\hat{x}(k) + 2\epsilon \sum_{i\in\mathcal{V}} \xi_i d_i e_i^2(k). \tag{5.22}$$

Using (5.13) and (5.19), we have

$$\begin{aligned} x^T(k)\Xi Lx(k) &\leq 2\hat{x}^T(k)\Xi L\hat{x}(k) + \frac{\|\Xi L\|\sigma_{\max}(k)}{\epsilon \min_i d_i}\sum_{i\in\mathcal{V}} \hat{q}_i(k-1) + \frac{\|\Xi L\|}{\epsilon \min_i \theta_i d_i}\sum_{i\in\mathcal{V}}\chi_i(k) \\ &\leq 2\hat{x}^T(k)\Xi L\hat{x}(k) + \frac{\|\Xi L\|\sigma_{\max}(k)NM}{\epsilon d_{\min}} + \frac{\|\Xi L\|}{\epsilon \min_i \theta_i d_i}\sum_{i\in\mathcal{V}}\chi_i(k). \end{aligned}$$

We use the above relation in (5.22) to obtain

$$\begin{aligned} \nabla V \leq &-\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max}\ a_{\max})x^T(k)\Xi Lx(k) \\ &+ \sigma_{\max}(k)NM\left(\xi_{\max} + \|\Xi L\|\frac{1-2\epsilon d_{\max}\ a_{\max}}{2d_{\min}}\right) \\ &+ \left[\frac{\|\Xi L\|(1-2\epsilon d_{\max}\ a_{\max}) + 2d_{\max}\xi_{\max}}{2\min_i \theta_i d_i}\right]\sum_{i\in\mathcal{V}}\chi_i(k). \end{aligned}$$

From (5.15), we have

$$\nabla V \leq -\frac{1}{2}(\epsilon - 2\epsilon^2 d_{\max}\ a_{\max})\frac{\lambda_2}{\mu_N}V(k) + \Delta_1(k) + K_1 \sum_{i\in\mathcal{V}} \chi_i(k),$$

where $K_1 = \frac{\|\Xi L\|(1-2\epsilon d_{\max}\ a_{\max})+2d_{\max}\xi_{\max}}{2\min_i \theta_i d_i}$. We recall that $\Delta_1(k)$ goes to zero exponentially. From (5.14), we have

$$\nabla V \leq -\Omega V(k) + \Delta_1(k) + K_1 \sum_{i\in\mathcal{V}} \chi_i(k). \tag{5.23}$$

Now, by (5.16) and (5.19),

$$\chi_i(k+1) \leq \left(\beta_i - \frac{1}{\theta_i}\right)\chi_i(k) + 2\sigma_i(k)M.$$

Hence, we can bound $\chi_i(k)$, which goes to zero by (5.20). From (5.23), it follows that $V(k) \to 0$ as $k \to \infty$. This implies that weighted consensus is obtained asymptotically. □

5.4 Protocol 3: Self Triggered

We notice from (5.5) and (5.17) that the static and dynamic event-based triggering protocols require continuous monitoring of agent states, which can require extra energy. This aspect can be improved if each agent makes predictions regarding its time for the next triggering based on current states. This is the underlying idea of the self-triggered control mechanism to be discussed in this section.

We recall that agents update their states using the protocol (5.1) and (5.3) as

$$x_i(k+1) = x_i(k) - \epsilon \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i(k) - \hat{x}_j(k)). \tag{5.24}$$

Following the event driven protocol, the state of agent i at $k \in [t_l^i, t_{l+1}^i)$ can be written as

$$x_i(k) = x_i(t_l^i) - \epsilon \sum_{l=t_l^i}^{k-1} \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{x}_i(l) - \hat{x}_j(l)),$$

where t_l^i denotes the most recent triggering instant of agent i. We express the broadcast error $e_i(k) = \hat{x}_i(k) - x_i(k)$ as

$$e_i(k) = \epsilon \sum_{l=t_l^i}^{k-1} \sum_{j \in \mathcal{N}_i} a_{ij} u_{ij}(l),$$

where $u_{ij}(l) = \hat{x}_i(l) - \hat{x}_j(l)$.

We note that as the neighbors might trigger at any point in the interval $k \in [t_l^i, t_{l+1}^i)$, $u_{ij}(k)$ may not be a constant in this case. As a consequence, determining the value of $e_i(k)$ in this interval is difficult. In order to obtain an estimate on $e_i(k)$, we need to determine an upper bound of $u_{ij}(k)$. This will enable us to make an estimate on the future triggering time t_{l+1}^i at t_l^i.

Our strategy is to guarantee that the error $e_i(k)$ goes to zero exponentially under the bound

$$\left|e_i(k)\right| \leq \frac{\eta}{\sqrt{d_i}} \gamma^{\frac{k}{2}}, \tag{5.25}$$

where the design parameters are $\eta > 0$ and $\gamma \in (0, 1)$.

We now present the self-triggering algorithm. First, for the control protocol (5.24), the parameter ϵ should be taken small enough that $\Phi \in (0, 1)$, where

$$\Phi = 1 - (\epsilon - 4\epsilon^2 d_{\max}) \frac{\lambda_2}{\mu_N}. \tag{5.26}$$

Also let

$$\Gamma = 2\epsilon(1+\epsilon)N\eta^2. \tag{5.27}$$

Moreover, we define the following function to be used later:

$$\begin{aligned} g_i(k) = \epsilon &\left| \sum_{l=t_l^i}^{k-1} \sum_{j\in\mathcal{N}_i} a_{ij}(t_{ij}^1 - t_l^i)u_{ij}(t_l^i) \right| \\ &+ \epsilon \sum_{l=t_l^i}^{k-1} \sum_{j\in\mathcal{N}_i} a_{ij} \sum_{m=t_{l+1}^j}^{t_{ij}^2-1} \left[\left(\frac{\eta}{\sqrt{d_i}} + \frac{\eta}{\sqrt{d_j}} \right) \gamma^{\frac{m}{2}} + f(m) \right], \end{aligned}$$

where $t_{ij}^1(k) = \min\{k, t_{l_j+1}^j\}$, and $t_{ij}^2(k) = \max\{k, t_{l_j+1}^j\}$ for $k \in [t_{l_i}^i, t_{l_i+1}^i)$. Moreover, let

$$f(m) = \left[2\Big(\Phi^m V_{\max} + \frac{\Phi^m - \gamma^m}{\Phi - \gamma}\Gamma\Big) \right]^{1/2}, \tag{5.28}$$

where $m \geq 0$ and $V_{\max}$ satisfies $V(0) \leq V_{\max}$, i.e., it serves as an upper bound on the Lyapunov candidate at the initial time.

For all agents, the starting time $t_0^i = 0$ is set as the first triggering instant. The triggering instants $\{t_l^i\}_{l=1}^{\infty}$ for agent i is set via the following triggering protocol

$$t_{l+1}^i = \min\left\{ k > t_l^i : g_i(k) > \frac{\eta}{\sqrt{d_i}}\gamma^{\frac{k}{2}} \right\}. \tag{5.29}$$

It is worth noting that in (5.29), $g_i(k)$ is a function of the current state values of the neighbors of agent i and itself and increases while $\gamma^{k/2}$ exponentially decreases over $k \in [t_l^i, t_{l+1}^i)$. Hence, at any triggering time t_l^i, agent i can calculate in real time from (5.29) its next triggering time t_{l+1}^i.

The following is the third result of this chapter, concerning multi-agent systems employing the self-triggered protocol.

Theorem 5.4.1. *Consider the multi-agent system* (5.1) *with the strongly connected underlying communication digraph* $\mathcal{G}$*. The self-triggered control protocol* (5.29) *achieves weighted consensus asymptotically.*

Proof. We follow the similar process as the one in the proof for Theorem 5.2.1 by employing the same Lyapunov candidate from (5.8):

$$\begin{aligned}\nabla V &= -2\epsilon x^T \Xi L x - 2\epsilon x^T \Xi L e + \epsilon^2 (x+e)^T L^T \Xi L (x+e) \\ &\leq -2\epsilon \sum_{i\in\mathcal{V}} \xi_i q_i + 2\epsilon \sum_{i\in\mathcal{V}} \xi_i d_i e_i^2 + 2\epsilon \sum_{i\in\mathcal{V}} \xi_i \frac{q_i}{2} + 4\epsilon^2 \sum_{i\in\mathcal{V}} \xi_i d_i q_i + 2\epsilon^2 \sum_{i\in\mathcal{V}} d_i e_i^2,\end{aligned}$$

where

$$q_i = \frac{1}{2} \sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j)^2 \geq 0.$$

We can further bound ∇V using (5.25) as

$$\begin{aligned}\nabla V &\leq -\sum_{i\in\mathcal{V}} (\epsilon - 4\epsilon^2 d_i)\xi_i q_i + 2\epsilon(1+\epsilon)\sum_{i\in\mathcal{V}} d_i e_i^2 \\ &\leq -(\epsilon - 4\epsilon^2 d_{\max})\frac{\lambda_2}{\mu_N} V(k) + 2\epsilon(1+\epsilon) N \eta^2 \gamma^k.\end{aligned}$$

From (5.26) and (5.27), we obtain

$$V(k+1) \leq \Phi V(k) + \Gamma \gamma^k.$$

Using linear iterations, we have

$$\begin{aligned}V(k) &\leq \Phi^k V_{\max} + \sum_{l=0}^{k-1} \Phi^{k-l-1} \Gamma \gamma^l \\ &= \Phi^k V_{\max} + \frac{\Phi^k - \gamma^k}{\Phi - \gamma}\Gamma.\end{aligned}$$

We next proceed to find an upper bound on $|x_i(k) - x_j(k)|$ using (5.28) as

$$\begin{aligned}
|x_i(k) - x_j(k)| &\leq |x_i(k) - \overline{x}(0)| + |x_j(k) - \overline{x}(0)| \\
&\leq \sqrt{2(|x_i(k) - \overline{x}(0)|^2 + |x_j(k) - \overline{x}(0)|^2)} \\
&\leq \left[2\Big(\Phi^k V_{\max} + \frac{\Phi^k - \gamma^k}{\Phi - \gamma}\Gamma\Big)\right]^{1/2} \\
&= f(k).
\end{aligned}$$

An upper bound on $|u_{ij}(k)|$ can be found as

$$\begin{aligned}
|u_{ij}(k)| &= |\hat{x}_i(k) - \hat{x}_j(k)| \\
&\leq |\hat{x}_i(k) - x_i(k)| + |x_i(k) - x_j(k)| + |x_j(k) - \hat{x}_j(k)| \\
&\leq \left(\frac{\eta}{\sqrt{d_i}} + \frac{\eta}{\sqrt{d_j}}\right)\gamma^{\frac{k}{2}} + f(k). \qquad (5.30)
\end{aligned}$$

We are now going to find an upper bound on $|e_i(k)|$. At time t_l^i, agent i has information on t_l^j, $x_j(t_l^j)$, and t_{l+1}^j for $j \in \mathcal{N}_i$. It follows that $u_{ij}(k)$ is constant for $k \in [t_l^i, t_{ij}^1]$ and for $k > t_{ij}^1$, we can upper bound it by (5.30). Thus, we have

$$|e_i(k)| = \epsilon\left|\sum_{l=t_l^i}^{k-1}\sum_{j\in\mathcal{N}_i} a_{ij}u_{ij}(k)\right| \leq g_i(k), k \in [t_l^i, t_{l+1}^i),$$

that is, $|e_i(k)|$ is bounded by $g_i(k)$. Therefore, using self-triggered protocol (5.29), we arrive at

$$|e_i(k)| \leq \frac{\eta}{\sqrt{d_i}}\gamma^{\frac{k}{2}}, \ \ \forall k \in [t_l^i, t_{l+1}^i).$$

Hence, the agent network asymptotically achieves weighted consensus under the self-triggered protocol. □

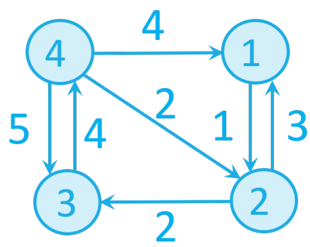

Figure 5.1: Network topology

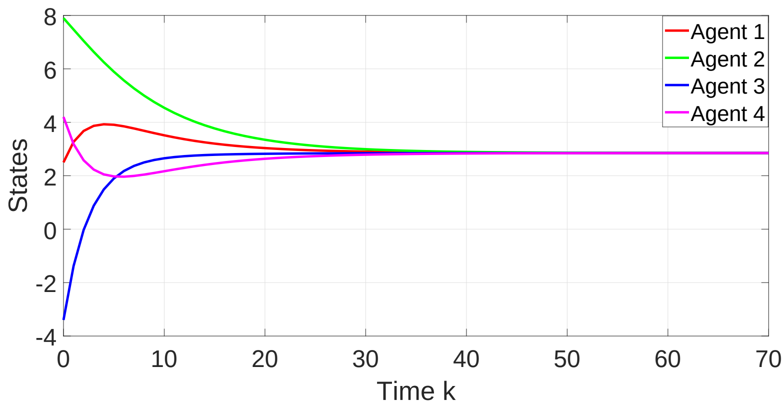

Figure 5.2: State evolutions under the original protocol (5.2)

5.5 Numerical Example

We consider the strongly connected digraph comprising four agents shown in Fig. 5.1. The left eigenvector corresponding to the zero eigenvalue of its Laplacian is $\xi^T = [0.03\ 0.21\ 0.28\ 0.48]$, which satisfies the conditions $\xi^T L = 0$ and $\sum_{i \in \mathcal{V}} \xi_i = 1$ as given in Lemma 5.1.1. The initial states for agents are set as $x(0) = \begin{bmatrix} 2.5 & 7.9 & -3.4 & 4.2 \end{bmatrix}$. The objective is to achieve consensus on the weighted average of these initial states, which is $\overline{x}(k) = 2.85$. The state evolutions of agents using original control protocol (5.2) with communication at every step k are shown in Fig. 5.2. We observe that the agents reach weighted consensus. We next examine the performances of the proposed protocols in terms of convergence rates and numbers of triggering times for agents. Their parameters are chosen so that the protocols have similar convergence rates.

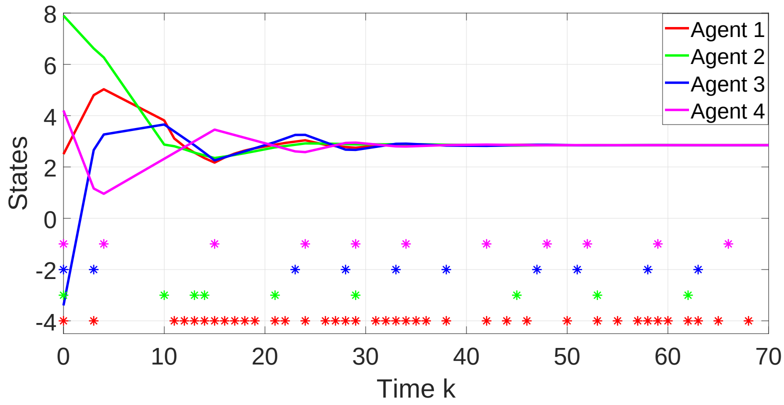

Figure 5.3: State evolutions and triggering instants under the static event-triggering law (5.5)

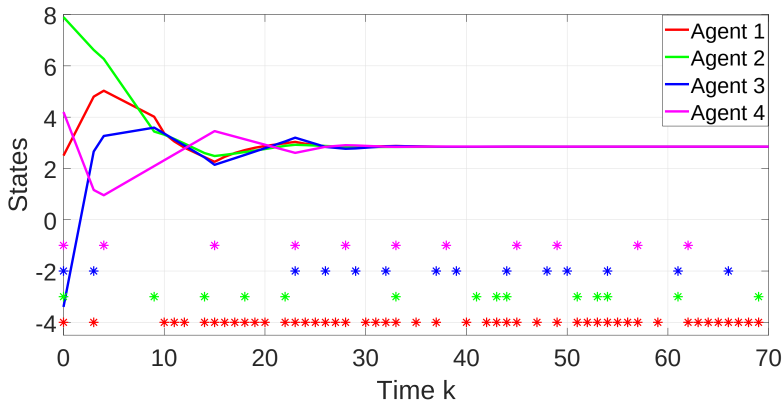

Figure 5.4: State evolutions and triggering instants under the state-independent triggering law

First, we look at the results of the static event-triggered protocol (5.5). The parameters are set as $\epsilon = .03$ and $\sigma_i = 0.1$. The state evolutions of agents and their triggering instants are shown (with their corresponding colors) in Fig. 5.3. It can be seen that the network achieves weighted consensus. Each agent triggers 17.25 times in 70 steps on average. Note that agent 1 is triggered most frequently under this protocol.

We now make comparison with the conventional event-triggering protocol from [81] employing the threshold $0.1\mathrm{e}^{-0.01k}$, which is independent of the state. The state evolutions for individual agents and their triggering times are shown in Fig. 5.4. We observe similar convergence rates as the static protocol however with about 25% more triggering times. Each agent triggers 21.75 times in 70 steps on average.

Next, we examine network performances with the dynamic event-triggered protocol (5.17). We used the same values for the parameters ϵ and σ_i as in the static event-triggered protocol. We set the other parameters as $\delta_i = .60$, $\beta_i = .35$, $\chi_i(0) = 10$ and $\theta_i = 10$. The results are presented in Fig. 5.5 in a similar fashion as above. It can be noticed from Figs. 5.3 and 5.5 that the triggering instants are less with the dynamic protocol when compared with the static version. In the dynamic case, the average number of triggering times per agent is 13 in 70 steps, about 30% less than the static triggered case. We point out that the agents demonstrate similar convergence rates as in the other protocols though the level of oscillations in their states in the initial phase seem more rough. Also, there is no particular period where events occur more often, and for agent 1, the number of transmissions has especially decreased. This seems to indicate that the event times are more carefully chosen by the dynamic event-triggered protocol.

We finally discuss the performances with the self-triggered protocol (5.29). The parameters are set as $\eta = 10$ and $\gamma = .99$. The simulation results are shown in Fig. 5.6. It can be observed that the convergence rates of the network are similar for all the proposed protocols. The number of triggering times are higher with self-triggered protocol. Agents 1 and 3 have the maximum weighted degree, which may be the reason of more events at these nodes.

We highlight that the proposed protocols provide different options for achieving consensus in agent networks. They exhibit different characteristics in terms of both convergence and communication/control update requirements.

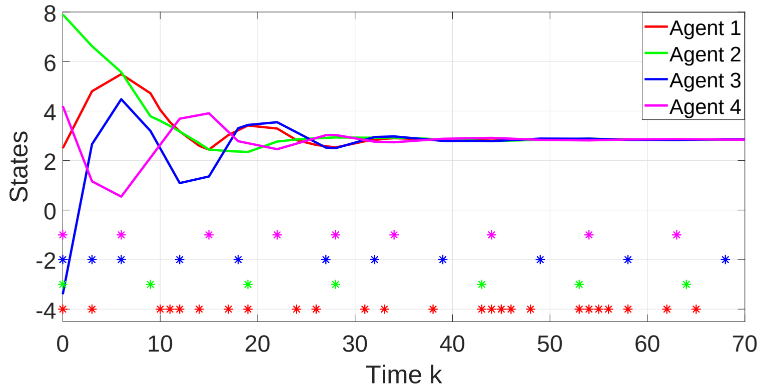

Figure 5.5: State evolutions and triggering instants under the dynamic event-triggering law (5.17)

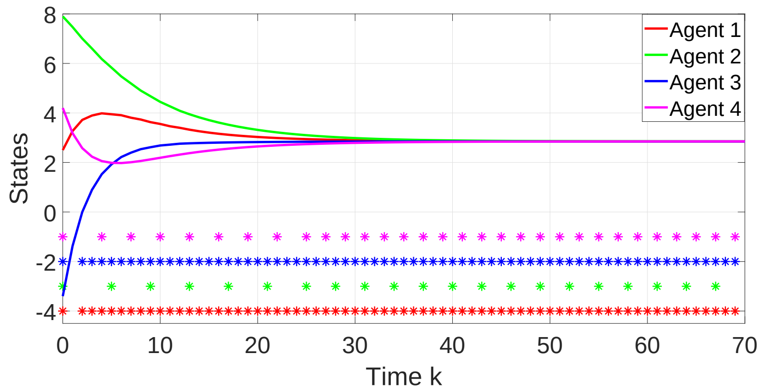

Figure 5.6: State evolutions and triggering instants under the self-triggering law (5.29)

5.6 Resilient Consensus Problem

Cyber physical systems are synergistic integration of communication, computation and control combining both physical devices and cyber layer. The communication network in CPS is prone to malicious attacks, making security and reliability the prime design concern. The two common category of attacks are denial-of-service (DoS) attacks affecting the timeliness of information exchange and deception attacks deteriorating the reliability and integrity of data [21]. We focus our attention on DoS attacks [15] that can considered as partial or total interruption of communications.

We consider event-triggered resilient consensus problem of MAS in discrete time under DoS attacks. In continuous-time setting there are few works which address this issue [28], [92]. The objective of the attackers is to destroy consensus by denying the communications between agents. We assume that the DoS attacks simultaneously affect both the measurement (sensor to controller) and the control (controller to actuator) channels.

We recall the fundamental case of MAS with single-integrator dynamics described as

$$x_i(k+1) = x_i(k) + u_i(k), \quad i \in \mathcal{V},$$

The distributed event-triggered implementation of controller for agent i under Dos attacks is given by

$$u_i(k) = \begin{cases} 0, & \text{under DoS} \\ -\epsilon \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i(k) - \hat{x}_j(k)), & \text{otherwise} \end{cases} \tag{5.31}$$

We now introduce the class of DoS attacks considered in our study. The network may not reach consensus if the DoS attacks are active at all times. We assume that there exist $\Pi_d \geq 0$ and $\nu_d \in [0, 1]$ such that for $k \in \mathbb{Z}_+$, the duration of DoS attacks are confined by

$$\Phi_d(k) \leq \Pi_d + \nu_d(k). \tag{5.32}$$

Here ν_d is the fraction of sampling times under attacks. The value $\Pi_d > 0$ enables the attack to be active at time 0. This type of model for DoS attacks are considered in continuous time setting [21], [79] as well as in discrete time framework [26], [46].

In the presence of DoS attacks, the Lyapunov function evolves in two modes.

In nominal mode when there are no attacks, the network evolves under the static triggering protocol 5.5 as discussed in section 5.2 while it evolves as constant under disruptions. We next find bounds on Lyapunov functions in both modes as

$$V(k+1) \leq \begin{cases} \Lambda V(k), & \Lambda = 1, \text{under DoS} \\ (1-\Omega)V(k) + \Delta_1(k), & \text{otherwise} \end{cases} \tag{5.33}$$

Here, the design parameter Ψ is chosen as $\Psi \in (0,1)$, where $\Psi = 1 - \Omega$. The Lyapunov function in multiplicative form can be given as

$$\begin{aligned} V(k) &\leq \Big(\Psi^{k-\Phi_d(k)} + \sum_{l=0}^{k-\Phi_d(k)-1} \Psi^{k-\Phi_d(k)-l-1}\Delta_1(l)\Big)\Lambda^{\Phi_d(k)}V(0) \\ &\leq \Psi^{k-\Pi_d-\nu_d(k)}\Lambda^{\Phi_d(k)}V(0) + \sum_{l=0}^{k-\Phi_d(k)-1} \Psi^{k-\Phi_d(k)-l-1}\Delta_1(l)\Lambda^{\Phi_d(k)}V(0) \\ &\leq \Psi^{k-\Pi_d-\nu_d(k)}V(0) + \sum_{l=0}^{k-\Phi_d(k)-1} \Psi^{k-\Phi_d(k)-l-1}\Delta_1(l)\Lambda^{\Phi_d(k)}V(0). \end{aligned} \tag{5.34}$$

Note that $\Psi \in (0,1)$ and the last term in the inequality 5.34 is exponentially decreasing to zero. Thus, it follows that $V(k) \to 0$ as $k \to \infty$. This implies that average consensus is obtained asymptotically under DoS attacks.

We consider the MAS network shown in Fig. 5.7 under DoS attacks. We assume that all communication links fail simultaneously during the disruption period. We examined the static-triggering protocol (5.5) to solve consensus problem in presence of DoS attacks. The state responses of agents are presented in Fig. 5.8 and the corresponding triggering instants for each agent are shown in Fig. 5.9. The shaded areas represent the DoS attack intervals. The agents achieve weighted consensus with slower convergence compared with the case without DoS attacks as shown in Fig. 5.3.

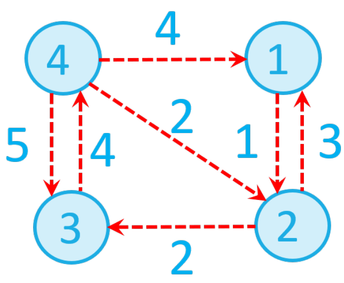

Figure 5.7: Network topology under DoS attacks

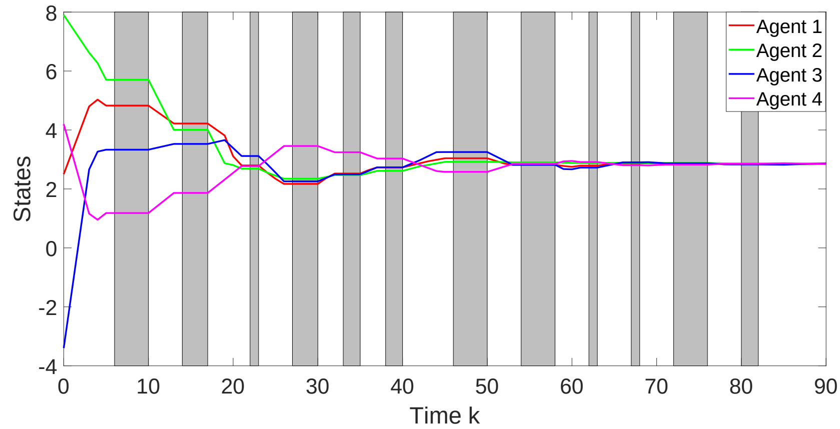

Figure 5.8: State evolutions under the static event-triggering law (5.5) with DoS attacks

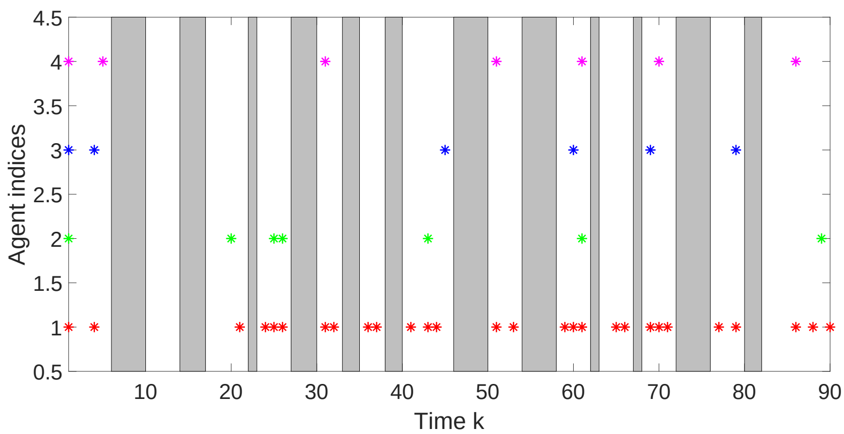

Figure 5.9: Triggering instants under the static event-triggering law (5.5) with DoS attacks

Chapter 6

Conclusion

6.1 Summary of Achievements

The main contribution of the thesis can be summarized as follows:

1. **Static event-triggered protocol**

   We proposed a novel static event-triggered protocol for the state consensus problem in discrete time MAS. The inter agent communications occur opportunistically to alleviate communication and computational burden. We have employed event-triggering protocol with threshold mechanism based on the state values received from neighboring agents. We characterized its advantages in terms of reducing transmission and update frequencies for the agents.

2. **Dynamic event-triggered protocol**

   We generalized the static-triggering protocol such that the threshold involved an auxiliary dynamic variable that allowed larger inter event times. We observed that with more complex protocols in triggering events, the number of triggering instants became less. The static triggering protocol is a particular case of the dynamic triggering protocol.

3. **Self-triggered protocol**

   We developed self-triggered protocols to avoid continuous monitoring and listening requirements. Under the approach, each agent only needs to make updates/broadcasts at its own triggering instants and listen to its neighbors at their announced triggering instants.

We developed a unified framework for these triggering based protocols to solve state consensus problem for MAS with single integrator dynamics as well as with general linear dynamics. We further extend our proposed framework to solve the weighted consensus problems in MAS having integrator dynamics with directed topologies.

6.2 Future Directions

We have focused our attention on distributed triggering based strategies to solve multi-agent consensus problem. Even though state control problems in MAS have been well addressed in literature, there is still much space for improvement over the existing methodologies. There are many potential research directions worthy of further exploration. We suggest some of the interesting yet challenging research topics below.

1. **Resilient consensus with event-triggered control**

   The cyber security issue is prevailing in the context of networked systems as more and more devices are getting connected. The communication network may easily fall victim to malicious attacks. The security problem in CPS are more challenging compared with the general cyber security as the impact of cyber attacks can transcend to the physical world [85]. For e.g., if a nuclear power plant is open-loop unstable then failure of communication between controller and actuator can severely damage the environment.

We are interested in developing event-triggered protocols to solve resilient consensus problems for agent systems having linear dynamics.

2. **Finite-time event-triggered consensus**

   Most studies on distributed event-triggered protocol guarantee asymptotic consensus of MAS. In many practical agent systems, it is expected to reach event-triggered consensus in finite time [55]. It is an interesting yet challenging area of research to ensure fast convergence while reducing communication and computational burden. Furthermore, it is more desirable to develop fixed time event-triggered protocols [52] to achieve consensus in preset time while saving energy expenses.

3. **Event-triggered consensus in directed network**

   In the undirected network topology, the Laplacian matrix is symmetrical and average consensus can be achieved. The average consensus problem for the directed graph with continuous communication is addressed in the work [12]. In order to reach consensus, a rooted spanning tree must be present in the digraph. The event-triggered consensus problem in directed MAS having integrator dynamics [17], [49] and linear dynamics [42], [54] are studied.

4. **Distributed optimization**

   The event-triggered protocols may be used to reduce information exchanges between agents in order to solve the distributed optimization problem. The objective here is to find an optimal control strategy subjected to given constraints or specific cost functions [53], [93]. The application area includes wireless sensor networks and source localization. This particular problem of interest is a worthy avenue for future exploration.

5. **Event-triggered protocols under stochasticity**

   Most studies on consensus problem consider the deterministic agent dynamics and deterministic event-triggering protocols. In practice, the stochastic measurement and communication noise may be present in the MAS network. There are only few works [10], [33] where events are triggered stochastically. This area is worthwhile for future investigations.

References

[1] Adaldo, A., Liuzza, D., Dimarogonas, D. V., and Johansson, K. H. Cloud-supported formation control of second-order multiagent systems. *IEEE Transactions on Control of Network Systems 5*, 4 (2018), 1563–1574.

[2] Akashi, S., Ishii, H., and Cetinkaya, A. Self-triggered control with tradeoffs in communication and computation. *Automatica 94* (2018), 373–380.

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

[4] Antsaklis, P. Goals and challenges in cyber-physical systems research editorial of the editor in chief. *IEEE Transactions on Automatic Control 59*, 12 (2014), 3117–3119.

[5] Åström, K. J., and Wittenmark, B. *Computer-Controlled Systems: Theory and Design.* Courier Corporation, 2013.

[6] Baskar, L., De Schutter, B., Hellendoorn, J., and Papp, Z. Traffic control and intelligent vehicle highway systems: A survey. *Intelligent Transport Systems, IET 5* (2011), 38–52.

[7] Bemporad, A., Heemels, W., and Johansson, M., Eds. *Networked Control Systems*. Lecture notes in control and information sciences. Springer, 2011.

[8] Bertsekas, D., and Tsitsiklis, J. *Parallel and Distributed Computation: Numerical Methods*. Athena scientific optimization and computation series. Prentice Hall, 1989.

[9] Borkar, V., and Varaiya, P. Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control 27*, 3 (1982), 650–655.

[10] Brunner, F. D., Antunes, D., and Allgower, F. Stochastic thresholds in event-triggered control: A consistent policy for quadratic control. *Automatica 89* (2018), 376–381.

[11] Bullo, F. *Lectures on Network Systems*. Kindle Direct Publishing, 2019.

[12] Cai, K., and Ishii, H. Average consensus on general strongly connected digraphs. *Automatica 48*, 11 (2012), 2750–2761.

[13] Carli, R., and Zampieri, S. Network clock synchronization based on the second-order linear consensus algorithm. *IEEE Transactions on Automatic Control 59*, 2 (2014), 409–422.

[14] Cetinkaya, A., Ishii, H., and Hayakawa, T. Networked control under random and malicious packet losses. *IEEE Transactions on Automatic Control 62*, 5 (2017), 2434–2449.

[15] Cetinkaya, A., Ishii, H., and Hayakawa, T. An overview on denial-of-service attacks in control systems: Attack models and security analyses. *Entropy 21*, 2 (2019).

[16] Chen, X., and Hao, F. Event-triggered average consensus control for discrete-time multi-agent systems. *IET Control Theory and Applications 6* (2012), 2493–2498.

[17] Chen, X., Hao, F., and Rahmani, A. Asynchronous decentralised event-triggered control of multi-agent systems. *International Journal of Control 87*, 10 (2014), 2130–2139.

[18] Chu, X., Peng, Z., Wen, G., and Rahmani, A. Distributed formation tracking of multi-robot systems with nonholonomic constraint via event-triggered approach. *Neurocomputing 275* (2018), 121–131.

[19] Cruz, D., McClintock, J., Perteet, B., Orqueda, O. A., Cao, Y., and Fierro, R. Decentralized cooperative control - a multivehicle platform for research in networked embedded systems. *IEEE Control Systems Magazine 27*, 3 (2007), 58–78.

[20] De Persis, C., and Frasca, P. Robust self-triggered coordination with ternary controllers. *IEEE Transactions on Automatic Control 58*, 12 (2013), 3024–3038.

[21] De Persis, C., and Tesi, P. Input-to-state stabilizing control under denial-of-service. *IEEE Transactions on Automatic Control 60*, 11 (2015), 2930–2944.

[22] Degroot, M. H. Reaching a consensus. *Journal of the American Statistical Association 69*, 345 (1974), 118–121.

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

[24] Dou, C.-X., and Liu, B. Multi-agent based hierarchical hybrid control for smart microgrid. *IEEE Transactions on Smart Grid 4*, 2 (2013), 771–778.

[25] Fan, Y., Hu, G., and Egerstedt, M. Distributed reactive power sharing control for microgrids with event-triggered communication. *IEEE Transactions on Control Systems Technology 25*, 1 (2017), 118–128.

[26] Feng, S., Cetinkaya, A., Ishii, H., Tesi, P., and Persis, C. D. Networked control under dos attacks: Tradeoffs between resilience and data rate. *IEEE Transactions on Automatic Control 66*, 1 (2021), 460–467.

[27] Feng, S., and Ishii, H. Dynamic quantized consensus of general linear multi-agent systems under denial-of-service attacks. *IFAC-PapersOnLine 53*, 2 (2020), 3533–3538. 21st IFAC World Congress.

[28] Feng, Z., and Hu, G. Secure cooperative event-triggered control of linear multiagent systems under dos attacks. *IEEE Transactions on Control Systems Technology 28*, 3 (2020), 741–752.

[29] Garcia, E., Cao, Y., and Casbeer, D. W. An event-triggered control approach for the leader-tracking problem with heterogeneous agents. *International Journal of Control 91*, 5 (2018), 1209–1221.

[30] Garcia, E., Mou, S., Cao, Y., and Casbeer, D. W. An event-triggered consensus approach for distributed clock synchronization. In *Proc. of the American Control Conference* (2017), pp. 279–284.

[31] Ge, X., and Han, Q.-L. Distributed formation control of networked multi-agent systems using a dynamic event-triggered communication mechanism. *IEEE Transactions on Industrial Electronics 64*, 10 (2017), 8118–8127.

[32] Girard, A. Dynamic triggering mechanisms for event-triggered control. *IEEE Transactions on Automatic Control 60*, 7 (2015), 1992–1997.

[33] Han, D., Mo, Y., Wu, J., Weerakkody, S., Sinopoli, B., and Shi, L. Stochastic event-triggered sensor schedule for remote state estimation. *IEEE Transactions on Automatic Control 60*, 10 (2015), 2661–2675.

[34] He, W., and Mo, Z. Secure event-triggered consensus control of linear multiagent systems subject to sequential scaling attacks. *IEEE Transactions on Cybernetics* (2021), 1–14.

[35] Heemels, W., and De Schutter, B. Special issue on analysis and design of hybrid and networked control systems. *Nonlinear Analysis: Hybrid Systems 10* (2013), 1–3. Special Issue related to IFAC Conference on Analysis and Design of Hybrid Systems (ADHS 12).

[36] Heemels, W., Johansson, K., and Tabuada, P. An introduction to event-triggered and self-triggered control. In *Proc. of the 51st IEEE Conference on Decision and Control (CDC)* (2012), pp. 3270–3285.

[37] Heemels, W. P. M. H., Donkers, M. C. F., and Teel, A. R. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control 58*, 4 (2013), 847–861.

[38] Heemels, W. P. M. H., Sandee, J. H., and Bosch, P. P. J. V. D. Analysis of event-driven controllers for linear systems. *International Journal of Control 81*, 4 (2008), 571–590.

[39] Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. A survey of recent results in networked control systems. *Proceedings of the IEEE 95*, 1 (2007), 138–162.

[40] Hetel, L., Fiter, C., Omran, H., Seuret, A., Fridman, E., Richard, J.-P., and Niculescu, S. I. Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica 76* (2017), 309–335.

[41] Hu, W., Liu, L., and Feng, G. Consensus of linear multi-agent systems by distributed event-triggered strategy. *IEEE Transactions on Cybernetics 46*, 1 (2016), 148–157.

[42] Hu, W., Yang, C., Huang, T., and Gui, W. A distributed dynamic event-triggered control approach to consensus of linear multiagent systems with directed networks. *IEEE Transactions on Cybernetics 50*, 2 (2020), 869–874.

[43] Jadbabaie, A., Lin, J., and Morse, A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control 48*, 6 (2003), 988–1001.

[44] Johan Astrom, K., and Bernhardsson, B. Comparison of periodic and event based sampling for first-order stochastic systems. *IFAC Proceedings Volumes 32*, 2 (1999), 5006–5011.

[45] Kadowaki, Y., and Ishii, H. Event-based distributed clock synchronization for wireless sensor networks. *IEEE Transactions on Automatic Control 60*, 8 (2015), 2266–2271.

[46] Kang, X., and Ishii, H. Resilient control of uncertain networked systems under dos attacks. In *Proc. of the 60th IEEE Conference on Decision and Control (CDC)* (2021).

[47] Lamnabhi-Lagarrigue, F., Annaswamy, A., Engell, S., Isaksson, A., Khargonekar, P., Murray, R. M., Nijmeijer, H., Samad, T.,

Tilbury, D., and Van den Hof, P. Systems and control for the future of humanity, research agenda: Current and future roles, impact and grand challenges. *Annual Reviews in Control 43* (2017), 1–64.

[48] Lehmann, D., Henriksson, E., and Johansson, K. H. Event-triggered model predictive control of discrete-time linear systems subject to disturbances. In *Proc. of the European Control Conference (ECC)* (2013), pp. 1156–1161.

[49] Li, H., Liao, X., Huang, T., and Zhu, W. Event-triggering sampling based leader-following consensus in second-order multi-agent systems. *IEEE Transactions on Automatic Control 60*, 7 (2015), 1998–2003.

[50] Li, Z., Duan, Z., and Chen, G. Consensus of discrete-time linear multi-agent systems with observer-type protocols. *Discrete and Continuous Dynamical Systems - B 16*, 2 (2011), 489–505.

[51] Li, Z., Duan, Z., Chen, G., and Huang, L. Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers 57*, 1 (2010), 213–224.

[52] Liu, J., Zhang, Y., Yu, Y., and Sun, C. Fixed-time event-triggered consensus for nonlinear multiagent systems without continuous communications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 49*, 11 (2019), 2221–2229.

[53] Liu, Q., Yang, S., and Hong, Y. Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks. *IEEE Transactions on Automatic Control 62*, 8 (2017), 4259–4265.

[54] Liu, X., Du, C., Liu, H., and Lu, P. Distributed event-triggered consensus control with fully continuous communication free for general linear multi-agent systems under directed graph. *International Journal of Robust and Nonlinear Control 28*, 1 (2018), 132–143.

[55] Lu, Q., Han, Q.-L., Zhang, B., Liu, D., and Liu, S. Cooperative control of mobile sensor networks for environmental monitoring: An event-triggered finite-time control scheme. *IEEE Transactions on Cybernetics 47*, 12 (2017), 4134–4147.

[56] Lynch, N. A. *Distributed Algorithms.* Morgan Kaufmann, 1996.

[57] Ma, C.-Q., and Zhang, J.-F. Necessary and sufficient conditions for consensusability of linear multi-agent systems. *IEEE Transactions on Automatic Control 55*, 5 (2010), 1263–1268.

[58] Marescaux, J., Leroy, J., and Gagner, M. Transatlantic robot-assisted telesurgery. *Nature 413* (2001), 379–380.

[59] Matsume, H., Wang, Y., and Ishii, H. Resilient self/event-triggered consensus based on ternary control. *Nonlinear Analysis: Hybrid Systems 42* (2021), 101091.

[60] Mazo, M., and Tabuada, P. On event-triggered and self-triggered control over sensor/actuator networks. In *Proc. of the 47th IEEE Conference on Decision and Control* (2008), pp. 435–440.

[61] Meng, W., Wang, X., and Liu, S. Distributed load sharing of an inverter-based microgrid with reduced communication. *IEEE Transactions on Smart Grid 9*, 2 (2018), 1354–1364.

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

[63] Mesbahi, M., and Egerstedt, M. *Graph Theoretic Methods in Multi-agent Networks*. Princeton University Press, 2010.

[64] Mishra, R. K., and Ishii, H. Average consensus in discrete-time multi-agent systems with distributed event-triggered control. In *Proc. of the European Control Conference (ECC)* (2021), pp. 608–613.

[65] Mishra, R. K., and Ishii, H. Dynamic event-triggered consensus control of discrete-time linear multi-agent systems. *IFAC-PapersOnLine 54*, 17 (2021), 123–128.

[66] Mishra, R. K., and Ishii, H. Event-triggered control for discrete-time multi-agent average consensus. *International Journal of Robust and Nonlinear Control* (2021), 1–18.

[67] Nowzari, C., and Cortes, J. Zeno-free, distributed event-triggered communication and control for multi-agent average consensus. In *Proc. American Control Conference* (2014), pp. 2148–2153.

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

[69] Nowzari, C., Garcia, E., and Cortes, J. Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica 105* (2019), 1–27.

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

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

[72] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials 16*, 1 (2014), 414–454.

[73] Postoyan, R., Tabuada, P., Nesic, D., and Anta, A. A framework for the event-triggered stabilization of nonlinear systems. *IEEE Transactions on Automatic Control 60*, 4 (2015), 982–996.

[74] Ren, W., and Atkins, E. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control 17* (2007), 1002–1033.

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

[76] Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 25–34.

[77] Scardovi, L., and Sepulchre, R. Synchronization in networks of identical linear systems. *Automatica 45*, 11 (2009), 2557–2562.

[78] Schenato, L., and Fiorentin, F. Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica 47*, 9 (2011), 1878–1886.

[79] Senejohnny, D., Tesi, P., and De Persis, C. A jamming-resilient algorithm for self-triggered network coordination. *IEEE Transactions on Control of Network Systems 5*, 3 (2018), 981–990.

[80] Seo, J. H., Shim, H., and Back, J. Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach. *Automatica 45*, 11 (2009), 2659–2664.

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

[82] Su, Y., and Huang, J. Stability of a class of linear switching systems with applications to two consensus problems. *IEEE Transactions on Automatic Control 57*, 6 (2012), 1420–1430.

[83] Sun, Z., Huang, N., Anderson, B. D. O., and Duan, Z. A new distributed zeno-free event-triggered algorithm for multi-agent consensus. In *Proc. 55th IEEE Conference on Decision and Control (CDC)* (2016), pp. 3444–3449.

[84] Tarbouriech, S., Seuret, A., Moreira, L., and da Silva Jr, J. G. Observer-based event-triggered control for linear systems subject to cone-bounded nonlinearities. *IFAC-PapersOnLine 50*, 1 (2017), 7893–7898.

[85] Teixeira, A., Shames, I., Sandberg, H., and Johansson, K. H. A secure control framework for resource-limited adversaries. *Automatica 51* (2015), 135–148.

[86] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters 75*, 6 (1995), 1226–1229.

[87] Wang, Y., and Ishii, H. Resilient consensus through event-based communication. *IEEE Transactions on Control of Network Systems 7*, 1 (2020), 471–482.

[88] Wu, C. W., and Chua, L. Application of graph theory to the synchronization in an array of coupled nonlinear oscillators. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 42*, 8 (1995), 494–497.

[89] Xiao, F., Chen, T., and Gao, H. Synchronous hybrid event- and time-driven consensus in multiagent networks with time delays. *IEEE Transactions on Cybernetics 46*, 5 (2016), 1165–1174.

[90] Xie, G., Liu, H., Wang, L., and Jia, Y. Consensus in networked multi-agent systems via sampled control: Fixed topology case. In *Proc. of the American Control Conference* (2009), pp. 3902–3907.

[91] Xing, L., Wen, C., Guo, F., Liu, Z., and Su, H. Event-based consensus for linear multiagent systems without continuous communication. *IEEE Transactions on Cybernetics 47*, 8 (2017), 2132–2142.

[92] Xu, W., Hu, G., Ho, D. W. C., and Feng, Z. Distributed secure cooperative control under denial-of-service attacks from multiple adversaries. *IEEE Transactions on Cybernetics 50*, 8 (2020), 3458–3467.

[93] Yang, S., Liu, Q., and Wang, J. A multi-agent system with a proportional-integral protocol for distributed constrained optimization. *IEEE Transactions on Automatic Control 62*, 7 (2017), 3461–3467.

[94] Yi, X., Liu, K., Dimarogonas, D. V., and Johansson, K. H. Dynamic event-triggered and self-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control 64*, 8 (2019), 3300–3307.

[95] Yi, X., Lu, W., and Chen, T. Pull-based distributed event-triggered consensus for multiagent systems with directed topologies. *IEEE Transactions on Neural Networks and Learning Systems 28*, 1 (2017), 71–79.

[96] You, K., and Xie, L. Network topology and communication data rate for consensusability of discrete-time multi-agent systems. *IEEE Transactions on Automatic Control 56*, 10 (2011), 2262–2275.

[97] Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. Internet of things for smart cities. *IEEE Internet of Things Journal 1*, 1 (2014), 22–32.

[98] Zhang, X., Liu, L., and Feng, G. Leaderfollower consensus of time-varying nonlinear multi-agent systems. *Automatica 52* (2015), 8–14.

[99] Zhang, X.-M., Han, Q.-L., and Yu, X. Survey on recent advances in networked control systems. *IEEE Transactions on Industrial Informatics 12*, 5 (2016), 1740–1752.

[100] Zhou, B., Liao, X., Huang, T., Li, H., and Chen, G. Event-based semiglobal consensus of homogenous linear multi-agent systems subject to input saturation. *Asian Journal of Control 19*, 2 (2017), 564–574.

[101] Zhou, F., Huang, Z., Yang, Y., Wang, J., Li, L., and Peng, J. Decentralized event-triggered cooperative control for multi-agent systems with uncertain dynamics using local estimators. *Neurocomputing 237* (2017), 388–396.

[102] Zhu, W., Jiang, Z.-P., and Feng, G. Event-based consensus of multi-agent systems with general linear models. *Automatica 50*, 2 (2014), 552–558.