Distributed resilient flocking control of multi-agent systems through event/self-triggered communication

Neda Amirian | Saeed Shamaghdari

School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

Correspondence
Saeed Shamaghdari, School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
Email: shamaghdari@iust.ac.ir

Abstract
This paper proposes a distributed control scheme for multi-agent systems to achieve resilient flocking behaviour via event-based communication. The control scheme can provide the required connectivity conditions to form a flock in the presence of cyberattacks in the network. This method is presented in a fully distributed manner to avoid the use of global data. The developed event-triggered update rules can mitigate the influence of the non-cooperative agents with the weighted mean subsequence-reduced algorithm and reduce unnecessary communication among them. It is proved that under the proposed method, convergence is guaranteed and there is no Zeno behaviour exhibited inherently. To relax the requirement of continuous monitoring of self-state, a self-triggered mechanism is proposed further. Simulation results are given to illustrate the theoretical analysis and show the advantages of the event- and self-triggered controllers.

1 INTRODUCTION

Flocking behaviour in multi-agent systems was inspired by the mass movement of animals in nature such as the aggregate motion of a flock of birds, a herd of land animals, or a school of fish. It has attracted intensive attention due to its wide applications such as distributed sensing using mobile sensor networks; self-assembly of connected mobile networks; automated parallel delivery of payloads; and performing military missions such as reconnaissance, surveillance, and combat using cooperative unmanned aerial vehicles [1]. In 1986, Reynolds proposed three rules that led to the formation of the flock in computer animation [2]. These rules are named by alignment: Attempt to velocity synchronisation in flock-members, cohesion: Attempt to stay close to flock-members and separation: Avoid collisions among flock-members.

Event-triggered control is a class of control strategy in which the controller is actuated by the occurrence of a special event [3]. The ability to reduce control costs and save resources makes event-triggered control appealing in resource-limited control systems. For example, the application of the event-triggered control strategy in a multi-agent system equipped with micro-processors can decrease the power consumption and increase the lifetime of networks [4]. Due to these advantages, growing attention has been paid to event-based control systems.

Some results related to different event-triggered control strategies have been reported in the current literature concerning multi-agent cooperative control. In [5], the average consensus problem for multi-agent systems (MASs) was addressed based on the event-based control strategy which renders both control signals and state measurement. In [6], the event-triggered average-consensus problem for multi-agent systems with direct and weighted topologies was investigated. In [7], the distributed rendezvous problem of multi-agent systems with novel event-triggered controllers was studied. Also, a combinational measurement approach to event design and developed the basic event-triggered control algorithm was used. In [8], the leader-following consensus problem was investigated for a class of switched non-linear multi-agent systems via event-triggered protocols, where the switching signal of each agent is different. Some distributed event-triggered control algorithms on the flocking problem have been proposed in [9, 10]. In these studies, a virtual leader is assumed to be tracked in the flocking motion. Generally, the flocking motion is easier to achieve when all agents are able to communicate with the leader, while for the leaderless flocking problem, limited communication and only

© 2021 The Authors. IET Control Theory & Applications published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology
local information pose a challenge for the coordination among agents.

Cybersecurity problem is a critical issue for this kind of systems [11]. Control algorithms of networked MASs are generally computed distributively without having centralised monitoring for the activity of agents. Therefore, unforeseen adverse conditions such as uncertainties or attacks can easily result in system instability and prohibit the accomplishment of the system objective [12]. There are two types of attacks on MASs, attack to connection links [13, 14] and attack to agent's dynamics [15, 16]. To mitigate the effects of attacks on agent's dynamics, a weighted mean subsequence-reduced (W-MSR) algorithm is used. This algorithm is practical and useful while its implementation is straightforward. Essential works on the W-MSR algorithm can be seen in [17, 18]. The W-MSR algorithm has been used to reach resilient consensus for second-order MASs in the presence of misbehaving agents [19] and resilient attitude alignment of leaderless multi-spacecraft systems in the presence of cyberattacks [20]. Also, in [21], the W-MSR algorithm has been employed to mitigate the effects of Byzantine nodes in directed networks in the consensus problem for switched MASs.

In [22], the authors presented a control law that allows a team of mobile robots to reach resilient flocking in the presence of non-cooperative robots. Their strategy relied on dynamic connectivity management. Also, in [23], the problem of flocking of multi-agent systems in presence of uncertainties and unknown disturbances was investigated. A biologically inspired distributed resilient controller was proposed based on a computational model of emotional learning.

In the previous studies on resilient flocking control [22], a large volume of information must be communicated in the network due to a large number of members in the flocks and high connectivity requirements of resilient control. On the other hand, its method isn’t entirely distributed and needs some global information. But in our study, these problems were addressed by event/self-triggered communications and distributed control structures, respectively. Hence, the system can be easily extended to a large scale. The contribution of our study is to show how the event-triggered as well as the self-triggered techniques can be implemented over MASs under the cyber-attacks and the Vicsek model to reach flocking behaviour and reduce unnecessary transmissions. The use of the W-MSR algorithm needs strong connectivity among agents, but these techniques can save the communication bandwidth and computation resources.

In this study, the purpose is to achieve resilient flocking behaviour in MASs with the Vicsek model [24] and dynamic network based on consensus protocol. An event-triggered three-state control system is designed to regulate relative distances in the MAS and make the agreement in direction of motions with saving energy in data transmission and reducing bandwidth requirements. Attack on the agent's dynamics causes the cooperative agents to become non-cooperative. F-locally bounded model is considered. In this model, each cooperative node has up to F non-cooperative neighbours, which threaten the group objective by preserving other agents from achieving valid states. The W-MSR algorithm is used to make attacks ineffective. The proposed control method guarantees to achieve resilient flocking for dynamic networks. Each state has a distinct event-triggering condition (ETC), and agents release their states regarding the ETC. Then, a novel self-triggering structure is suggested. Therefore, continuous monitoring of measurement suffered by the event-triggered control scheme can be avoided.

The rest of the study is organised as follows. In Section 2, notations and preliminaries on graph theory are given. The main results of flocking in MAS with distributed event-triggered control and distributed self-triggered control are presented in Sections 3 and 4. In Section 5, simulations are presented to validate the theoretical results. Conclusions are drawn in Section 6.

2 PRELIMINARIES ON GRAPHS

Some basic notions on graphs are introduced for the analysis in this section. An undirected graph $G = (\mathcal{V}, \mathcal{E})$ is used to model the communication topology of the multi-agent system, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the node-set representing the agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the undirected edge-set representing communication links. The neighbour-set of the $i$th agent is defined as $N_i = \{j \mid (i, j) \in \mathcal{E}\}$ and $n_i = |N_i|$ is the degree of the $i$th agent. The function $A_{ij}$ is determined to show the quality of the communication links between every pair of nodes $(i, j)$ based on their position in Euclidean space, $x_i \in \mathbb{R}^d$ [25].

$$A_{ij} = \begin{cases} 1 & x_i - x_j < \rho \\ 0 & x_i - x_j \geq R \\ \varepsilon & \frac{\|x_i - x_j\|}{R - \rho} \end{cases}$$

where $R$ defines a cutoff distance where the signal becomes unusable and $\rho$ defines a saturation distance where the communication between agents does not change as they get closer together. Based on function $A_{ij}$ (Equation (1)), the Laplacian matrix is defined as

$$[L]_{ij} = \begin{cases} -A_{ij} & i \neq j \\ \sum_j A_{ij} & i = j \end{cases}$$

For Laplacian matrix $L$, because we restrict the function $A_{ij}$ to be positive, all the eigenvalues are real and they satisfy $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d$ and $\lambda_2 > 0$ if and only if the graph is connected. The second smallest eigenvalue $\lambda_2$ is called algebraic connectivity of the system. It is an indicator of how much the topology graph is connected. In the remainder of this section, some important and basic definitions are provided.

Definition 1 (r-reachable). A set $S \subseteq \mathcal{V}$ is an r-reachable set if there exists a node $\exists v_i \in S$ such that $|N \setminus S| \geq r$, $r \in \mathbb{N}$.

Definition 2 (r-robust). A graph $G$ is r-robust if for any pair of non-empty, disjoint subsets $S_1, S_2 \subseteq \mathcal{V}$, at least one of them is r-reachable.
Definition 3 (flocking). A group of agents is said to flock when all agents attain the same velocity vector, distances between the agents are stabilised and no collisions among them occur.

Definition 4 (non-cooperative agent). An agent is cooperative if it applies the consensus update rule at every sample time and shares the result with its neighbours at each transmission. It is called non-cooperative otherwise.

Definition 5 (resilient-consensus). A multi-agent system is said to reach resilient consensus if the following conditions are satisfied, even in the presence of up to $F$ non-cooperative agents in the neighbours of each cooperative agent.

(a) Safety condition: There exists an interval $I$ that the cooperative agents’ value remains in the interval for all $k$.

(b) Consensus condition: The cooperative agents achieve consensus to a value that lies between the maximum and minimum initial values of the cooperative agents.

Following lemma is the main result in [22] that characterises the robustness of the communication network topology by a metric, and its validity has been proved.

Lemma 1. For a given graph $G$, $\frac{\lambda_2}{2}$ is a lower bound for robustness value and $r \geq \frac{\lambda_2}{2}$.

3 | EVENT-TRIGGERED RESILIENT FLOCKING

Considering an MAS of $N$ agents with a connected undirected graph $G$, the dynamics of each agent can be described as

$$x_i(k + 1) = x_i(k) + T.u_i(k) \quad i = 1, \ldots, n$$

(3)

where $x_i(k)$ and $u_i(k)$ are the position and the control input, respectively, of agent $i$ at $t = kT$ for $k \in \mathbb{Z}_+$.

In this study, the Vicsek model is used to model agents. In this model, particles are driven with a constant absolute velocity, and at each time step, the assumed average direction of motion of the particles in their neighbourhood is added to control relative distances in the system. Using event-triggered update rule is applied for the direction of motions.

$$\dot{\theta}_j(k + 1) = \dot{\theta}_j(k) + \sum_{j \in R_i(k)} \omega_{ij}(k) \left( \dot{\theta}_j(k) - \dot{\theta}_i(k) \right)$$

$$\dot{\theta}_j(k) = \dot{\theta}_j(k') \quad k \in [k', k_{j+1}']$$

(4)

where $\dot{\theta}_j(k)$ is the last communicated direction of node $j$ and $k_{j'}$, $k_1$, ... denotes the transmission times of node $j$ determined by an ETC. The weight $\omega_{ij}(k)$ satisfies $\omega_{ij}(k) > 0$ and $\sum_j \omega_{ij}(k) = 1$. Each agent updates its direction using its neighbours’ information and only when the ETC is satisfied, the agent sends its state to its neighbours. We assume the maximum number of the non-cooperative agents in the neighbour-set of the cooperative agents as known and determined by $F$.

In this update rule, the W-MSR algorithm is used to decrease the effects of the non-cooperative agents in the network. The W-MSR algorithm has three steps. First, every cooperative node uses the values $\dot{\theta}_j(k)$, $j \in N_i$, most recently communicated from the neighbours and creates a sorted list of them from largest to smallest. Second, comparing with $\dot{\theta}_j(k)$, node $i$ removes $F$ largest and $F$ smallest values from the list. If the number of values larger or smaller than $\dot{\theta}_j(k)$ is less than $F$, then all of them are removed. The set of values that remained in the list is denoted by $R_i(k)$. Third, each cooperative agent updates its value using Equation (4). With this algorithm, if the value of robustness is at least $2F + 1$, resilient consensus will be guaranteed. It means that for implementation of the W-MSR algorithm, the condition $r \geq 2F + 1$ must be confirmed at each sample time. So it is necessary to control the value of robustness via connectivity control during runtime. To this end, an event-triggered controller with three states is designed. The level of robustness must be computed repeatedly, and the value of $r$ must be preserved in the range of $r \geq 2F + 1$. For a given time-varying network, calculating the level of robustness is an intricate problem. For this reason, we use a lower bound of $r$ to control relative distances in the system. Using Lemma 1, $\frac{\lambda_2}{2}$ is a lower-bound for $r$ and $\lambda_2$ must be preserved in the range of $\lambda_2 > 4F$ to satisfy $r \geq 2F + 1$. The structure of the event-triggered controller can be seen in Figure 1. The system switches from one state to another according to the value of current algebraic connectivity. Also, each state has a distinct ETC.

In state 1, the amount of $\dot{\lambda}_2$ is less than $4F$; therefore, the system is not guaranteed to achieve resilient consensus. In this state, the agents apply control law as in Equation (5), which increases the algebraic connectivity of the system:

$$u_i = \dot{\lambda}_2$$

(5)

where $\dot{\lambda}_2$ is an event-triggered gradient of $\lambda_2$, which can be computed from Equation (6). Equation (6) is an event-triggered form of $\nabla \lambda_2$ in [25]. For the calculation of $\dot{\lambda}_2$, the distributed

---

**FIGURE 1** Three-state event-triggered controller
method in [26] is used in an event-triggered manner:

\[
\frac{\partial \lambda_2 (L)}{\partial X_{i\alpha}} = \text{trace} \left\{ \begin{bmatrix} h_2^T \hat{p}_2 \\ \hat{p}_2 \end{bmatrix} ^T \left[ \frac{\partial L(\xi, \hat{\xi})}{\partial X_{i\alpha}} \right] \right\}
\]

(6)

\[
\hat{s}_j \ (k) = \xi_j \ (k) - \hat{\xi}_j \ (k)
\]

where \( \alpha_{i\alpha} \) is a single element of \( \alpha_j \). ETC in state 1 is considered as follows:

\[
e_{i,1} \ (k + 1) = \xi_j \ (k + 1) - \hat{\xi}_j \ (k) \]

(7)

\[
f_{i,1} \ (k + 1) = E_{i,1} \ (k + 1) - (a_1 + b_1 e^{-c_1(k+1)})
\]

(8)

where \( a_1, b_1, c_1 > 0 \) are positive constants.

In this study, Zeno behaviours are effectively avoided. The event-triggered control is based on periodic event detections. Every two neighboring event detections are performed discontinuously and separated by a fixed time step.

In the following theorems, we show that by using the controller in Equation (5), the system goes from states 1 to 2. In other words, \( \hat{\nabla} \hat{\lambda}_2 \) can meet control objectives instead of \( \nabla \hat{\lambda}_2 \) properly (Theorem 1). Then by regulation of the parameter \( \epsilon \) in Equation (9), the system never re-enters state 1. When the agents are in states 2 and 3, it is guaranteed that they will agree on a direction of motion through the resilient consensus protocol even in the presence of up to \( F \) non-cooperative agents (Theorem 2). The switching frequency between states 2 and 3 is limited. An upper bound for it will be calculated (Theorem 3).

**Theorem 1.** For a multi-agent system with the Vicsek model, time-varying undirected topology and the controller in Figure 1, if the system is in state 1, the elements of \( \nabla \hat{\lambda}_2 \) converge to the elements of \( \nabla \hat{\lambda}_2 \) and the controller in Equation (5) drives the system to state 2:

\[
\hat{\nabla} \hat{\lambda}_2 - \nabla \hat{\lambda}_2 \leq 4nh \sqrt{d}
\]

(15)

where \( b = K(\epsilon \frac{\rho_\text{max}}{\rho_\text{min}} + \frac{R}{F}) \).

**Proof.** The elements of \( \nabla \hat{\lambda}_2 \) and \( \hat{\nabla} \hat{\lambda}_2 \) are defined as \( g_{i\alpha} \) and \( \hat{g}_{i\alpha} \), respectively.

\[
g_{i\alpha} = \text{trace} \left( \frac{\partial A_{1j}}{\partial X_{i\alpha}} \right) ^T \frac{\partial L}{\partial X_{i\alpha}}
\]

(16)

\[
\hat{g}_{i\alpha} = \text{trace} \left( \frac{\partial A_{1j}}{\partial X_{i\alpha}} \right) ^T \frac{\partial L}{\partial X_{i\alpha}}
\]

(17)

\[
v_2 = \begin{bmatrix} v_{21} \ v_{22} \ \ldots \ v_{2n} \end{bmatrix} ^T, \ v_{2j} \in [0, 1]
\]

(18)

\[
\hat{v}_2 = \begin{bmatrix} \hat{v}_{21} \ \hat{v}_{22} \ \ldots \ \hat{v}_{2n} \end{bmatrix} ^T, \ \hat{v}_{2j} \in [0, 1]
\]

(19)

\[
g_{i\alpha} = \frac{1}{v_2} \sum_{j=1}^{n} \frac{\partial A_{1j}}{\partial X_{i\alpha}} - v_{21} \frac{\partial A_{12}}{\partial X_{i\alpha}} - \ldots - v_{2n} \frac{\partial A_{1n}}{\partial X_{i\alpha}}
\]

(20)

\[
\hat{g}_{i\alpha} = \frac{1}{v_2} \sum_{j=1}^{n} \frac{\partial A_{1j}}{\partial X_{i\alpha}} - v_{21} \frac{\partial A_{12}}{\partial X_{i\alpha}} - \ldots - v_{2n} \frac{\partial A_{1n}}{\partial X_{i\alpha}}
\]

(21)
\[ g_{i\alpha} = r_{12}^2 \frac{\partial A_{1i}}{\partial x_{1\alpha}} + r_{22}^2 \frac{\partial A_{2j}}{\partial x_{2\alpha}} + \ldots + r_{n2}^2 \sum_{j=1}^{n} \frac{\partial A_{ij}}{\partial x_{j\alpha}} \]

Also, \( \hat{g}_{i\alpha} \) has a similar form of Equation (20). For the computation of the matrix \( \frac{\partial L}{\partial x_{i\alpha}} \), just the elements containing \( x_{i\alpha} \) might be non-zero. These elements are in the \( i \)-th row and column with \( j \in N_i \) and diagonal elements. Also, we write the equations for the neighbours in the interval \( \rho \leq ||x_i - x_j|| < R \); otherwise, \( \frac{\partial A_{ij}}{\partial x_{i\alpha}} \) equals zero:

\[ \hat{g}_{i\alpha} = \frac{\partial A_{1i}}{\partial x_{1\alpha}} + \frac{\partial A_{2j}}{\partial x_{2\alpha}} + \ldots + \frac{\partial A_{ij}}{\partial x_{j\alpha}} + \ldots \]

So, \( \| \hat{g}_{i\alpha} - g_{i\alpha} \| \) can be written as

\[ \| \hat{g}_{i\alpha} - g_{i\alpha} \| = \left\| \left( r_{12}^2 \frac{\partial A_{1i}}{\partial x_{1\alpha}} - r_{12}^2 \frac{\partial A_{1i}}{\partial x_{1\alpha}} \right) + \left( r_{22}^2 \frac{\partial A_{2j}}{\partial x_{2\alpha}} - r_{22}^2 \frac{\partial A_{2j}}{\partial x_{2\alpha}} \right) + \ldots + \left( r_{n2}^2 \sum_{j=1}^{n} \frac{\partial A_{ij}}{\partial x_{j\alpha}} - r_{n2}^2 \sum_{j=1}^{n} \frac{\partial A_{ij}}{\partial x_{i\alpha}} \right) \right\| \]

According to Equation (1), \( \frac{\partial A_{ij}}{\partial x_{i\alpha}} \) is as follows:

\[ \frac{\partial A_{ij}}{\partial x_{i\alpha}} = K \frac{x_{i\alpha} - x_{i\alpha}}{x_i - x_j} e^{\left( \frac{-((x_{i\alpha} - x_j)^2)}{k^2 \rho} \right)} \]

The first term of the right side of inequality in Equation (24) is examined with \( j = 1, i \neq j \).

If \( k = k_1 \), \( \epsilon_1 = e_{1\alpha} = 0 \). Also, according to the initial assumptions, \( \rho \leq ||x_i - x_j|| < R, r_{12} \in [0, 1] \) and \( r_{12} \in [0, 1] \). So,
Equation (27) converts to:
\[
\left| r_{12}^2 \frac{\partial A_i}{\partial X_{i \alpha}} - r_{12}^2 \frac{\partial A_i}{\partial X_{i \alpha}} \right| \leq b'_i \alpha \tag{28}
\]
\[
\begin{align*}
\beta' &= \frac{2KR}{\rho} \\
\end{align*}
\]

If \( k \in (k_1, k_1+1) \), \( e_1 < a_1 + b_1e^{-\gamma(k+1)} \) and \( e_{i, k} < a_i + b_i e^{-\gamma(k+1)} \). As \( a_1 = 0 \) and \( k \rightarrow \infty \), Equation (27) converts to:
\[
\left| r_{12}^2 \frac{\partial A_i}{\partial X_{i \alpha}} - r_{12}^2 \frac{\partial A_i}{\partial X_{i \alpha}} \right| \leq b' \tag{29}
\]
\[
\begin{align*}
b &= K \left( \frac{k_1}{k_1} + R + a_1 + \frac{R}{\rho} \right) \\
\end{align*}
\]

A similar process is applied to the rest terms in Equation (24), and it can be written as
\[
\hat{g}_{\alpha} - g_{\alpha} \leq \sum_{j=1}^{n} 4b = 4nb \tag{31}
\]

According to the results in Equations (28) to (31), in all conditions, the elements of \( \nabla A_2 \) converges to the elements of \( \nabla A_2 \).

By using Equation (31) for all elements of \( \nabla A_2 = \nabla A_2 \), Equation (15) is achieved.

**Theorem 2.** For a multi-agent system with the Vicsek model, time-varying undirected topology and the controller in Figure 1, if the system is in state 2, it will never re-enter state 1, and if there are no more than \( F \) non-cooperative agents, the direction of the agents will asymptotically converge to the same value.

**Proof.** If the system is in state 1, according to Theorem 1, it will move to state 2. The system never returns to state 1 if the parameter \( \varepsilon \) is chosen such that \( 0 \leq \frac{d\lambda_2}{dt} \). Therefore, it suffices to show that:
\[
0 \leq \frac{d\lambda_2}{dt} \tag{32}
\]

Using the controller from Equations (9) and (34), the following condition for \( \varepsilon \) is arrived:
\[
\frac{-\nabla A_2^T V_i}{\nabla A_2^T V \lambda_2} \leq \varepsilon \tag{35}
\]

Because there is no access to \( \nabla \lambda_2 \) at each sample time, the condition in Equation (35) can be rewritten as
\[
\left[ \hat{g}_{\alpha} \pm \max_{e_{i, a}} \hat{g}_{\beta} \pm \max_{e_{i, \beta}} \right] \tag{36}
\]
\[
\frac{T}{\nabla \lambda_2} \leq \varepsilon \\
\]
\[
\hat{g}_{\alpha} = g_{\alpha} \pm \varepsilon_{i, a} \tag{37}
\]
\[
\max e_{i, a} = 4nb \tag{38}
\]

Equation (38) is the maximum value of error in Equation (37). The sign of errors in Equation (36) should be chosen in such a way that the amount obtained for \( \varepsilon \) is more conservative.

In states 2 and 3, the system is guaranteed to be \((2F + 1)\)–robust. So according to the results in [27], each cooperative agent will converge to a common value for direction of the motion, as long as there are no more than \( F \) non-cooperative agents.

**Theorem 3.** For a multi-agent system with the Vicsek model, time-varying undirected topology and the controller in Figure 1, if there are no more than \( F \) non-cooperative agents, the switching frequency between states 2 and 3 is bounded and
\[
f_{s_{32}} \leq \frac{G}{4} \cdot F \cdot \frac{\eta_2 - \lambda_2}{(0)} \tag{39}
\]

**Proof.** By the regulation of \( \varepsilon \) such that \( 0 < \frac{d\lambda_2}{dt} \), system will inter to state 3. The minimum time of switching from states 3 to 2 must be considered. To investigate that, Equation (33) is used as
\[
\frac{d\lambda_2}{dt} = \sum_{i=1}^{n} \frac{\partial \lambda_2}{\partial X_{i \alpha}} V_{i} \tag{40}
\]
\[
\sum_{i=1}^{n} \frac{\partial \lambda_2}{\partial X_{i \alpha}} V_{i} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[ r_{12}^T T \frac{\partial L(x_{i, \alpha})}{\partial X_{i \alpha}} \right] V_{i} \tag{40}
\]

For off-diagonal entries of \( [\frac{\partial L}{\partial X_{i, \alpha}}]_{ij}, k \in \{i, j\}: \)
\[
\frac{\partial A_{ij}}{\partial X_{i, \alpha}} = -\frac{\partial A_{ij}}{\partial X_{j, \alpha}} \tag{41}
\]
\[
\begin{align*}
\sum_e \frac{\partial A_{ij}}{\partial x_i} V_{e,k} & = \frac{\partial A_{ij}}{\partial x_i} V_{i,k} - \frac{\partial A_{ij}}{\partial x_j} V_{j,k} \\
& = \frac{\partial A_{ij}}{\partial x_i} (V_{i,k} - V_{j,k}) \tag{42}
\end{align*}
\]

And for diagonal entries of \( \frac{\partial L_i}{\partial x_i} \),
\[
\sum_e \sum_j \frac{\partial A_{ij}}{\partial x_i} V_{e,k} = \sum_j \frac{\partial A_{ij}}{\partial x_i} (V_{i,k} - V_{j,k}) \tag{43}
\]

According to the first theorem of [27], the consensus error \(|\theta_i - \theta_j|\) has an upper bound that we name it by \( \delta_\theta \). So, the value of \(|\theta_i - \theta_j|\) converges to a constant small value by arriving at the consensus on the direction of the motions in state 3. As a result:
\[
\lim_{k \to \infty} |\theta_i - \theta_j| \leq \delta_\theta \tag{44}
\]
\[
\lim_{k \to \infty} \sum_{i=1}^{n} \frac{\partial L_i}{\partial x_i} V_{i,k} \leq U_K \tag{45}
\]

\[
U_K = \begin{cases} 
K_\text{rs} \sin \delta_\theta & \text{for off-diagonal entries} \\
\frac{K_\text{rs}}{\sin \delta_\theta} & \text{for diagonal entries}
\end{cases} 	ag{46}
\]

So, Equation (40) converts to the inequality in Equation (47).
\[
\text{trace} \left\{ \frac{W_2}{W_2} \sum_{i=1}^{n} \frac{\partial L_i}{\partial x_i} V_{i,k} \right\} \leq C_{\text{eq}} \tag{47}
\]

\[
C_{\text{eq}} = \text{trace} \left\{ \begin{bmatrix} 1 & \cdots & 1 \\
\vdots & \ddots & \vdots \\
1 & \cdots & 1 \\
\end{bmatrix} U_K \right\} 	ag{48}
\]

\[
\frac{\partial \lambda_2}{\partial t} \leq G \tag{49}
\]

\[
G = \sum_{d=\alpha,\beta}^d G_d \tag{50}
\]

\[
\int_0^t \frac{\partial \lambda_2}{\partial t} dt \leq \int_0^t G dt \tag{51}
\]

\[
\lambda_2 (t) \leq G t + \lambda_2 (0) \tag{52}
\]

When the system enters states 2 from 3 and \( \lambda_2 < \eta_2 (4F) \), a lower bound for the switching time is obtained.
\[
4F \eta_2 \leq GT_{i32} + \lambda_2 (0) \tag{53}
\]

So, the frequency of the switching becomes limited as
\[
f_{i32} \leq \frac{G}{4F \eta_2 - \lambda_2 (0)}
\]

### 3.1 Computation of \( \hat{v}_2 \)

For the computation of \( \hat{v}_2 \), the distributed algorithm in [26] is modified with the event-triggered mechanism. \( w_i \) is a row vector associated with each agent whose entries are the \( i \)-th component of all the eigenvectors of \( L \). Also, \( W' \) is a matrix, and its rows are \( w_i, i = 1, 2, \ldots, n \).
\[
W = \begin{bmatrix} w_1 \\
\vdots \\
w_n \end{bmatrix}
\]

We assume that \( i \)-th agent receives \( \hat{v}_i = w_i (k^{(i)}_t) \), \( j \in N_i \) from its neighbours when \( f_{i,s} > 0 \), \( s \in \{1, 2, 3\} \) and compute \( w_i \) by the following algorithm at each sample time. So, matrix \( W' \) converts to \( W' \).

1. Initialise \( w_i^{(0)} \) with a random vector;
2. \( e_i^{(i+1)} = L_i (k^{(i)}_t) \hat{v}^{(i)}_i \), where \( e_i^{(i+1)} \) is the update of the row \( w_i \) at the step \( i + 1 \) and \( L_i (k^{(i)}_t) \) is the \( i \)-th row of the Laplacian matrix. Since each element in \( L_i (k^{(i)}_t) \) is non-zero only for \( j \in N_i \), therefore the product \( L_i (k^{(i)}_t) \hat{v}^{(i)}_i \) requires only the rows of the matrix \( W' \) related with the agents connected with \( i \);
3. Compute \( e_2^{(i+1)} \), that is the average value of the second eigenvector of \( L_i \);
4. Compute \( w_i^{(i+1)} \) with ortho-normalisation of \( e_i^{(i+1)} \);
5. Repeat the loop until convergence of the vector \( w_i \);
6. For the computation of the average \( e_2^{(i+1)} \), the following algorithm is used;
7. Initialise \( e_2^{(0)} = e_2^{(i+1)} \);
\( (8) \quad \dot{\bar{e}}_j^{(r+1)} = \beta_j^{(r)} + \sum_{j \in N_j} M_{ij} \bar{e}_j^{(k)}, \) where \( \beta_j^{(k)} = \bar{e}_j (k_j), \ k \in [k_j, k_{j+1}]; \\
(9) \quad \text{Repeat the loop until convergence.} \\
M_j = \begin{cases} 
1 & \text{max} (u_j, a_j) \\
0 & \text{otherwise}
\end{cases}
\]

Also, for orthonormalisation of the row vector \( \dot{\bar{e}}_j^{(r+1)} \), we use the centralised orthonormalisation approach. The orthonormalisation is typically performed by the factorisation of the matrix \( \bar{W} \) in the form \( E = \bar{W} H \), where the rows of the matrix \( E \) are orthonormal eigenvectors \( e_j \) and \( H \) is an upper triangular matrix that must be computed by the following algorithm:

\[
S = E^T E = H^T H \rightarrow H = \text{chol}(S)
\]

(1) Initialise \( S^{(0)} = e_j^T e_j; \\
(2) S^{(r+1)} = S^{(r)} + \sum_{j \in N_j} M_{ij} S^{(k)}, \) where \( S^{(k)} = S_j (k_j), \ k \in [k_j, k_{j+1}]; \\
(3) \text{Repeat the loop until convergence;} \\
(4) \text{Compute} \ S = n.S.
\]

This distributed algorithm can estimate all eigenvectors of the Laplacian matrix. In the next step, regarding \( -L_2 = \hat{\lambda}_2 \hat{L}_2 \), each agent can estimate \( \hat{\lambda}_2 \) as

\[
\lambda_2^i = -\sum_{j \in N_i} \frac{L_{ij} \beta_j}{\bar{L}_2}
\]

where \( \lambda_2^i \) is the estimation of \( \lambda_2 \) for agent \( i \).

4 SELF-TRIGGERED RESILIENT FLOCKING

The event-triggered scheme proposed in the last section needs each agent to monitor its states continuously to check the triggering function. In this section, the results related to state 3 are extended to the self-triggered scheme (Figure 2), so the continuous self-state monitoring is relaxed. In state 3, the next triggering instant is preset at the previous triggering instant and no monitoring is required between two triggered events.

Let \( T^i \) be a trigger interval between the trigger times \( k_j^i \) and \( k_{j+1}^i \), that is, \( T^i = k_{j+1}^i - k_j^i \). Note that a trigger is not executed and agents do not update their control inputs \( u_i \) between

\[
\text{FIGURE 2} \quad \text{Three-state self-triggered controller}
\]

the trigger times \( k_j^i \) and \( k_{j+1}^i \). Then we have

\[
\theta_i (k_j^i + T^i) = \theta_i (k_j^i + T^i) - \theta_i (k_j^i) = T^i \sum_{j \in R_i(k_j^i)} \omega_j (k_j^i) \left( \hat{\theta}_j (k_j^i) - \theta_i (k_j^i) \right)
\]

(54)

\[
l^i_j = \arg \max \left\{ k_j^i, k_{j+1}^i < k \right\}
\]

(55)

Then, for each agent, \( k_j^i \) is the latest event time instant before the current sample time \( k \). Also, the error that is defined in Equation (13), in the trigger time \( k_j^i + T^i \) is

\[
\epsilon_{i,3} (k_j^i + T^i) = \theta_i (k_j^i + T^i) - \theta_i (k_j^i),
\]

(56)

\[
\Omega (k_j^i) = \sum_{j \in R_i(k_j^i)} \omega_j (k_j^i) \left( \hat{\theta}_j (k_j^i) - \theta_i (k_j^i) \right)
\]

(57)

The ETC in state 3 was \( \epsilon_{i,3} (k_j^i + T^i) = a_3 + b_3 e^{-\gamma (k_j^i + 1)} \), so it can be rewritten as

\[
T^i \Omega (k_j^i) = a_3 + b_3 e^{-\gamma (k_j^i + 1)}
\]

(58)

\[
\frac{T^i}{\Omega (k_j^i)} = \frac{a_3 + b_3 e^{-\gamma (k_j^i + 1)}}{\Omega (k_j^i)}
\]

(59)

According to Figure 2, the self-triggered communication is applied in states 2 and 3. In state 1, there is no control effort in \( \theta_i (k_j^i) \), and Equation (54) is not satisfied. So, this state remains with the event-triggered mechanism.

5 NUMERICAL EXAMPLE

In this section, some simulations were made to verify the performance of the event-triggered control system (Figure 1) and
FIGURE 3 Initial graph topology of the multi-agent system

FIGURE 4 Trajectory of $\lambda_2$

the self-triggered control system (Figure 2). An undirected MAS is considered with $n = 6$. Agents were initialised in a pattern as depicted in Figure 3. The attacks may occur in the agent located in the centre ($F = 1$). We indicated that with red colour. The system parameters are chosen as $\gamma = 0.5$, $\eta_1 = \frac{5}{6}$, $\eta_2 = \frac{7}{6}$, $\rho = 1.5$, $R = 4$, $T = 0.1$, $v = 0.03$. The initial angles are set as $\theta_1 (0) = 1$, $\theta_2 (0) = 5$, $\theta_3 (0) = 10$, $\theta_4 (0) = 3$, $\theta_5 (0) = 6$, $\theta_6 (0) = 2$.

First, the event-triggered three-state control system was applied to the MAS without the occurrence of an attack. The parameters of the event-triggering conditions are chosen as $a_1 = a_2 = 0.1$, $a_3 = 0.01$, $b_1 = b_2 = 0.1$, $b_3 = 0.01$, $c_1 = c_2 = c_3 = 2$. The trajectory of $\lambda_2$ has been plotted in Figure 4. It rises from 3.4 (in state 1) to 6 (in state 3). The amount of $\lambda_2$ in state 3 must satisfy the condition $\lambda_2 > 5.3$, but because of a jump before reaching 5.3, $\lambda_2$ arrives at 6. The cause of these jumps in the $\lambda_2$ diagrams is the entrance of the agents from the exponential interval to the constant interval $A_{ij} = 1$ in Equation (1). Figure 5 shows the behaviour of the agents in the 2D coordinate system. The agents initially close together to enhance the connectivity, then they start to align and reach a consensus on the direction of the motion. The heading value ($\theta_i$) of the agents is shown in Figure 6. According to the figure, the heading value of all agents converges to a particular value accurately. Sample times have decreased significantly for each agent compared to the time-triggered approach. Moreover, there is no data transformation after convergence.

In the next experiment, we considered that agent 6 was attacked and the attacker injected a constant value into the actuator. Figure 7 shows that cooperative agents neglect non-cooperative agent data and align with each other. The consensus value and attack value are shown obviously in Figure 7. Then, a time-varying attack was applied to agent 6. According to Figure 8, the heading value of agent 6 varies with time in a sinusoidal form. Likewise, this kind of attack was repelled well and the cooperative agents reached the consensus on the direction of motion.

To examine the performance of the self-triggered mechanism, we repeated the last experiments with the self-triggered control system (Figure 2). Consensus on the motion directions in three cases (no attack, constant attack, and the time-varying attack) was achieved, and the attacks were repelled successfully (Figures 9–11). Compared to the event-triggered control system, sample times increased but there is no requirement for continuous monitoring of self-state by agents. Each agent estimates the next sample according to the value of the neighbourhood error at the latest sample time.
Overall, these simulations explain that the three-state controller, when used with the W-MSR algorithm and the event/self-triggered mechanism, ensures that the agents achieve resilient flocking with minimum data exchange.

6 | CONCLUSION

In this study, the resilient event-triggered flocking problem of MASs with the Vicsek model and non-cooperative agents was solved. The method can manage connectivity for dynamic interaction topology with the event-triggered approach to guarantee the resilient consensus. To this end, an event-triggered three-state controller was designed. The convergence of motion directions was shown analytically. The theories represent that the suggested control structure can successfully drive the system to reach secure flocking behaviour. In the last recent studies, controllers need some global data from the whole system and
assume a lot of energy for data communications. Our study tried to resolve this issue by event-triggered mechanisms in a distributed manner. Also, a self-triggered mechanism was combined with the event-triggered structure, and a new structure that does not need for each agent to monitor its state continuously was made. In future research, the effects of other types of cyberattacks can be studied. Simulation results demonstrate that an event-triggered mechanism can reduce unnecessary data transmissions while preserving system performance.

REFERENCES

1. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Trans. Autom. Control 51(3), 401–420 (2006)
2. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: Proceedings of the 14th Annual Conference On Computer Graphics And Interactive Techniques, New York (1987). https://dl.acm.org/doi/abs/10.1145/37401.37406
3. Heng, Z., et al.: Event-triggered control in networked control systems: A survey. In: The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, (2015). https://ieeexplore.ieee.org/abstract/document/7162452
4. Ding, L., et al.: An overview of recent advances in event-triggered consensus of multiagent systems. IEEE Trans. Cybern. 48(4), 1110–1123 (2018)
5. Seyboth, G.S., et al.: Control of multi-agent systems via event-based communication. IFAC Proc. 44(1), 10086–10091 (2011)
6. Liu, Z., et al.: Event-triggered average-consensus of multi-agent systems with weighted and direct topology. J. Syst. Sci. Complexity 25(5), 845–855 (2012)
7. Fan, Y., et al.: Distributed event-triggered control of multi-agent systems with combinational measurements. Automatica 49(2), 671–675 (2013)
8. Zou, W., Xiang, Z.: Event-triggered leader-following consensus of nonlinear multi-agent systems with switched dynamics. IET Control Theory Appl. 13(9), 1222–1228 (2018)
9. Yu, P., et al.: Leader-follower flocking based on distributed Even triggered hybrid control. Int. J. Robust Nonlinear Control 26(1), 143–153 (2016)
10. Sun, F., et al.: Flocking in nonlinear multi-agent systems with time-varying delay via event-triggered control. Appl. Math. Comput. 350, 66–77 (2019)
11. Liu, J., et al.: Distributed event-triggered state estimators design for sensor networked systems with deception attacks. IET Control Theory Appl. 13, (17), 2783–2791 (2018)
12. Mustafa, A., Modares, H.: Attack analysis and resilient control design for discrete-time distributed multi-agent systems. IEEE Rob. Autom. Lett. 5(2), 369–376 (2019)
13. Lu, A.-Y., Yang, G.-H.: Distributed consensus control for multi-agent systems under denial-of-service. Inform. Sci. 439-440, 95–107 (2018)
14. Feng, Z., Hu, G.: Secure cooperative event-triggered control of linear multi-agent systems under DoS attacks. IEEE Trans. Control Syst. Technol. 28(3), 741–752 (2019)
15. Modares, H., et al.: Static output-feedback synchronisation of multi-agent systems: A secure and unified approach. IET Control Theory Appl. 12(8), 1095–1106 (2018)
16. Moghadam, R., Modares, H.: Resilient autonomous control of distributed multiagent systems in contested environments. IEEE Trans. Cybern. (99), 1–11 (2018)
17. LeBlanc, H.J., et al.: Resilient asymptotic consensus in robust networks. IEEE J. Sel. Areas Commun. 31(4), 766–781 (2013)
18. Dibaji, S.M., Ishii, H.: Consensus of second-order multi-agent systems in the presence of locally bounded faults. Systems Control Lett. 79, 23–29 (2015)
19. Dibaji, S.M., Ishii, H.: Resilient consensus of second-order agent networks: Asynchronous update rules with delays. Automatica 81, 123–132 (2017)
20. Rezaee, H., Abdollahi, F.: Resilient attitude alignment in multispacecraft systems. IEEE Trans. Aerosp. Electron. Syst. 55, 3651–3657 (2019)
21. Shang, Y.: Resilient consensus of switched multi-agent systems. Systems Control Lett. 122, 12–18 (2018)
22. Saundner, K., et al.: Resilient flocking for mobile robot teams', IEEE Robot. Autom. Lett. 2(2), 1039–1046 (2017)
23. Jafari, M., Xu, H.: A biologically-inspired distributed resilient flocking control for multi-agent system with uncertain dynamics and unknown disturbances. In: 2017 Resilience Week (RWS), Wilmington, Delaware (2017). https://ieeexplore.ieee.org/abstract/document/808651
24. Viesek, T., et al.: Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. 75(6), 1226 (1995)
25. Stump, E., et al.: Connectivity management in mobile robot teams. In: 2008 IEEE International Conference On Robotics And Automation, Pasadena, California (2008). https://ieeexplore.ieee.org/abstract/document/4177054
26. De Gennaro, M.C., Jadbabaie, A.: Decentralized control of connectivity for multi-agent systems. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, California (2006)
27. Wang, Y., Ishii, H.: Resilient consensus through asynchronous event-based communication. In: 2019 American Control Conference (ACC), Philadelphia, Pennsylvania (2019). https://ieeexplore.ieee.org/abstract/document/8815065

How to cite this article: Amirian N, Shamaghdari S. Distributed resilient flocking control of multi-agent systems through event/self-triggered communication. IET Control Theory Appl. 2021;15:559–569. https://doi.org/10.1049/cth2.12061