Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Flocking fragmentation formulation for multi-robot system under multi-hop and lossy ad hoc networks*

Silan LI[1], Shengyu ZHANG[1], Tao JIANG[‡2]

[1]*The Research Center of 6G Mobile Communications, School of Cyber Science and Engineering, and Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China*

[2]*The Research Center of 6G Mobile Communications, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*

E-mail: lisilan@hust.edu.cn; shengyu@hust.edu.cn; taojiang@hust.edu.cn

Received Apr. 27, 2023; Revision accepted Oct. 18, 2023; Crosschecked

Abstract: In this paper, we investigate the impact of network topology characteristics on flocking fragmentation for the multi-robot system under a multi-hop and lossy ad hoc network, including the network's hop count features and information's successful transmission probability (STP). More specifically, we first propose a distributed communication-calculation-execution protocol to describe the practical interaction and control process in the ad hoc network based multi-robot system, where the flocking control is realized by a discrete-time Olfati-Saber model incorporating STP-related variables. Then, we develop a fragmentation prediction model (FPM) to formulate the impact of hop count features on fragmentation for specific flocking scenarios. This model identifies the critical system and network features that are associated with fragmentation. Further considering general flocking scenarios affected by both hop count features and STP, we formulate the flocking fragmentation probability (FFP) by a data fitting model based on the back propagation neural network, whose input is extracted from the FPM. The FFP formulation quantifies the impact of key network topology characteristics on fragmentation phenomena. Simulation results verify the effectiveness and accuracy of the proposed prediction model and FFP formulation, and several guidelines for constructing the multi-robot ad hoc network are also concluded.

Key words: Multi-robot flocking; Flocking fragmentation probability; Fragmentation prediction; Multi-robot communication networks
https://doi.org/10.1631/FITEE.2300295 **CLC number:**

1 Introduction

Multi-robot flocking refers to a phenomenon wherein multiple robots move cooperatively based on certain information interaction and coordination rules (Shen et al., 2022; Ibuki et al., 2020). This behavior allows robot swarms to perform tasks collaboratively, such as attacking targets, hunting prey, and transporting supplies. To fulfill flocking, three widely-acknowledged heuristic rules were introduced by Reynolds in 1986, i.e., swarm cohesion, velocity consensus, and collision avoidance (Reynolds, 1987). Referring to these rules, different flocking models have been proposed for a variety of applications, such as the Vicsek model (Zheng et al., 2017), the Cucker-Smale (C-S) model (Shao et al., 2021), and the Olfati-Saber (O-S) model (Olfati-Saber, 2006). In this paper, we employ the O-S model to analyze flocking phenomena due to its superior ability to guarantee the safety and stability of a multi-robot system (Huang et al., 2021; Antonelli et al., 2010; Liu and Gao, 2020).

The O-S model indicates that robots need to

[‡] Corresponding author

[*] This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1803400

first obtain the state information (i.e., position and velocity) of a virtual leader and their neighbors, and then generate control instructions to adjust their own postures, so that flocking stability can be achieved. Based on the O-S model, numerous researches have been conducted to study the flocking behavior of a multi-robot swarm (MRS). In Antonelli et al. (2010), a null-space-based behavior control algorithm based on the O-S model is proposed for the flocking behavior in a realistic MRS system. In Olfati-Saber and Iftekhar (2012), the O-S model is extended to nonholonomic mobile robots with nonlinear dynamics, and the emergence of flocking behavior is proved. Liu and Gao (2020) revise the O-S model to guarantee steady communication performance among multiple unmanned aerial vehicles. In addition, He and Li (2017) and Seoung Kyou (2022) address the obstacle avoidance problem of multi-robot flocking on the basis of the O-S model.

Despite analyzing multi-robot flocking from different perspectives, the abovementioned studies share a common assumption that the virtual leader's state information (LSI) is always globally known to all robots. However, this may not be the case in practical scenarios. In MRS, a distributed ad hoc network is generally constructed (Li et al., 2022a; Arafat et al., 2021; XIn et al., 2018; Yunpeng et al., 2020; Hafeez et al., 2013), where each robot can only communicate with the neighbors within its transmission range (Alam et al., 2022; Arafat and Moh, 2022; Li et al., 2022b; Mohamed et al., 2023). As a result, the LSI is generally broadcast in a multi-hop manner over the entire swarm. Due to the instable communication links, the transmission of LSI over each hop may face the risk of failure, which makes it uncertain for robots to successfully acquire LSI. When parts of robots cannot obtain LSI, the fragmentation phenomenon may appear (Su et al., 2007). That is, some robots get separated from the swarm, leading to the instability and failure of flocking (Wang and Chen, 2020; Yazdani and Su, 2022).

Several studies in the literature have considered the impact of lossy LSI announcements on flocking fragmentation and presented corresponding solutions. Considering a continuous-time O-S model, Su et al. (2007) indicates that only the agents that are directly or indirectly informed of the LSI will eventually reach flocking stability, whereas the others may undergo separation from the teams. The smaller the informed group, the greater the number of agents likely to be lost. Yazdani and Haeri (2017) discretize the continuous-time O-S model for realistic MRS with discrete state measurement and data interaction. Considering that partial robots are informed of the LSI, the O-S model is optimized in the research of Yazdani and Haeri (2017) to guarantee the spatial connectivity of robots. Treating the spatial connectivity as network connectivity, the uninformed robots can acquire LSI from the informed ones through the connected network, so that fragmentation is prevented. Also, assuming that partial agents can obtain LSI directly, Yazdani et al. (2019) extend the discrete O-S model into the flocking scenario with a dynamic virtual leader. Further, in Yazdani et al. (2020), the model is improved to suit the cases where the state information of the leader and neighbors is updated over different time periods. Similar to Yazdani and Haeri (2017), both Yazdani et al. (2019) and Yazdani et al. (2020) prevent fragmentation through the idea of preserving the spatial connectivity of multiple robots.

The connectivity-preserved algorithm is effective on the premise of a good communication condition, such as the ideal condition that the successful transmission probability (STP) is 1 for all communication links. Only under ideal or good communication conditions can the spatial connectivity be treated as equivalent to network connectivity. However, in practice, the successful transmission of information over each link is a random event due to the unstable communication environment. Particularly, with the decrease in STP, the proportion of robots that can obtain LSI will decrease, which aggravates the possibility of flocking fragmentation. What's worse, the overall relay hop counts of LSI can also impact the proportion of informed agents. Specifically, for the robots that are far away from the leader, they face a higher risk of missing LSI since this information has to be relayed through multiple hops before reaching them. The higher the hop count, the less likely it is for the robot to obtain LSI. When a large number of robots have to obtain LSI at high relay hops, a global informing of LSI is difficult to guarantee, which in turn raises the FFP.

To summarize, except for the spatial connectivity preservation, it is also necessary to rationally design and optimize the network topology parameters, such as the relay hop count features and STP of

LSI, to prevent multi-robot flocking fragmentation. For flocking-stability-oriented network design and optimization, a fundamental problem is to ascertain how the key network topology characteristics affect the multi-robot flocking fragmentation phenomenon, which motivates us to formulate the relationship between FFP and network parameters. An accurate fragmentation formulation can provide a beneficial tool to assist network construction and algorithm design of multi-robot systems. For example, it can indicate the critical communication requirements of preventing flocking fragmentation, which is a reference for selecting appropriate communication protocols and devices. Besides, when designing control or resource optimization algorithms, the predicated fragmentation probability can be considered as an indicator or constraint of the optimization problems.

Despite its theoretical significance, few researches have studied the FFP formulation problem. To fill in this gap, we investigate the flocking performance of multi-hop ad hoc network based multi-robot systems in terms of the FFP, so as to provide some insights for network optimization. Specifically, we first propose a distributed communication-calculation-execution (CCE) protocol to describe the practical interaction and control process of a multi-robot flocking system. Based on that, we then consider a specific flocking scenario to characterize key hop count features and develop a fragmentation prediction model (FPM). Further, considering lossy transmission of LSI, a learning based model is constructed to formulate the FFP for general flocking scenarios. The contributions of the present research are as follows:

- In the proposed CCE protocol, we characterize the unstable transmission features of LSI and neighbor state information (NSI) in a multi-hop ad hoc network as two Bernoulli variables, and incorporate them into the discrete-time O-S model, which can describe the effects of communication network on the multi-robot flocking control processes.
- For two-dimensional flocking scenarios with specific initial states and partially known LSI, we propose a method based on layer division and spatial linearization to extract the key topology features of multi-hop networks. According to that, an analytical model, referred to as the FPM, is developed to predict if the flocking fragmentation will happen.
- Further, for two-dimensional flocking scenarios with random initial states and lossy transmission of LSI and NSI, we develop a data fitting model based on a back propagation (BP) neural network to formulate the FFP. The neural network takes key network topology parameters extracted from FPM as the input features, which makes the formulation applicable to robot swarms of different sizes with changeable and random initial states.
- Extensive simulation results validate the effectiveness of our methods in formulating the FPM and FFP models. Particularly, the obtained FFP formulation has excellent accuracy and generalization capability. Besides, the impacts of network parameters on FFP are explored.

The remainder of this paper is organized as follows: Section 2 presents related definitions concerned with multi-robot flocking scenarios based on the graph theory; Section 3 introduces the CCE protocol of robots, and presents the formulation of the research problem addressed in this paper; Section 4 introduces the FPM for flocking scenarios with relatively specific initial states, based on which the FFP is formulated in Section 5 for flocking scenarios with generally random states; Section 6 conducts simulation experiments to validate the accuracy of the formulated models and discusses the network parameters' effects on FFP; and finally, Section 7 presents the conclusions.

Notations. In this paper, $\mathbb{R}^n$ represents the set of all column vectors with dimension n and $\mathbb{R}^{m*n}$ represents the set of all matrixes with dimensions $m \times n$. The symbol $\mathbb{N}$ denotes the set of all natural numbers, $\|\cdot\|$ is the 2-norm, and $\mathbf{I}_n$ represents the identity matrix with dimensions $n \times n$. The expression $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product of matrixes $\mathbf{A}$ and $\mathbf{B}$.

2 Graph-based Definitions of Flocking Scenarios

In this research, we consider a multi-robot flocking scenario where N robots move cooperatively in n-dimensional Euclidean space following a virtual leader. The virtual leader represents a group objective and can be viewed as a moving rendezvous point, whose state information (i.e., speed and position) can be acquired by robots through communication or perception. The position and speed of robot i are, respectively, represented by $\boldsymbol{q}_i(t)$ and $\boldsymbol{p}_i(t) \in \mathbb{R}^n$, whereas the leader's position and speed are, respectively, denoted by $\boldsymbol{q}_r(t)$ and $\boldsymbol{p}_r(t) \in \mathbb{R}^n$. Each robot can measure its own position, speed, and acceleration by sensors, and can exchange these data via wireless communications. The state information of the leader is desired by robots, while the robots' state information is not necessary for the virtual leader.

Let r denote the communication range of robots. Then, two robots are adjacent neighbors which can communicate with each other if their distance is smaller than r. The neighbor set of robot i is represented by $\mathcal{N}_i(t) = \{j : \|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\| < r, 1 \leq j \leq N, j \in \mathbb{N}, j \neq i\}$. Let r_l denote the virtual leader's interaction range, and $\varOmega(t)$ denote the set of robots that are within this range. Thus $\varOmega(t) = \{i : \|\boldsymbol{q}_i(t) - \boldsymbol{q}_r(t)\| < r_l, 1 \leq i \leq N, i \in \mathbb{N}\}$ and all robots in $\varOmega$ can obtain LSI directly. We assume that the set $\varOmega(t)$ is not empty.

During the flocking process, robots need to avoid colliding with each other, guarantee the swarm's cohesion, and maintain the velocity consensus. To present mathematical definitions for these requirements, we first formulate the spatial distribution of robots as a spatially induced graph (SIG). SIG is defined as $G(\boldsymbol{q}(t)) = (\mathcal{V}, \mathcal{E})$, where $\boldsymbol{q} = [\boldsymbol{q}_1^T, \boldsymbol{q}_2^T, ..., \boldsymbol{q}_N^T]^T \in \mathbb{R}^{Nn}$, $\mathcal{V} = \{1, 2, ..., N\}$, and $\mathcal{E} = \{(i, j){:}i, j \in \mathcal{V}, i \neq j, j \in \mathcal{N}_i\}$. The vector $\boldsymbol{q}$ contains the positions of all robots, while $\mathcal{V}$ and $\mathcal{E}$ represent the graph's vertex set and edge set, respectively. Obviously, if $(i, j) \in \mathcal{E}$, then $(j, i) \in \mathcal{E}$. Therefore, the SIG is an undirected graph. Based on the graph, we present some definitions about flocking as below.

Definition 1 (α-Lattice) If $\|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\| = d$ for all $(i, j) \in \mathcal{E}$, then the spatial distribution of all robots forms an α-lattice configuration (Olfati-Saber, 2006). The parameters d and $k = r/d$ are referred to as the *scale* and *ratio* of the lattice, respectively.

Definition 2 (A Quasi α-Lattice) If $-\delta \leq \|\boldsymbol{q}_i(t) - \boldsymbol{q}_j(t)\| - d \leq \delta$ for all $(i, j) \in \mathcal{E}$, then the spatial distribution of all robots forms a quasi α-lattice configuration (Olfati-Saber, 2006).

Definition 3 (A Cohesive Group) Let $(\boldsymbol{q}(\cdot), \boldsymbol{p}(\cdot)) : t \mapsto \mathbb{R}^{Nn} \times \mathbb{R}^{Nn}$ denote the state trajectory of N robots over the time interval $[t_0, t_f]$, where $\boldsymbol{p} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, ..., \boldsymbol{p}_N^T]^T \in \mathbb{R}^{Nn}$. For all $t \in [t_0, t_f]$, if there exists a sphere with radius R_s centered at $\boldsymbol{q}_c(t) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{q}_i(t)$ that could contain all robots, then the group is regarded to be cohesive within $[t_0, t_f]$ (Olfati-Saber, 2006).

Definition 4 (A Spatially Connected Graph) If there exists a path between vertices i and j for all $i, j \in \mathcal{V}$, then graph G is spatially connected. It is noted that the spatial connectivity does not necessarily mean that the network is connected.

Definition 5 (Velocity Consensus) If $\boldsymbol{p}_i(t) = \boldsymbol{p}_j(t)$ for all $i, j \in \mathcal{V}$ over time interval $[t_0, t_f]$, then the group of robots achieve velocity consensus between time t_0 and t_f.

Definition 6 (Flocking Fragmentation) Flocking fragmentation is defined as the phenomenon that the associated SIG becomes spatially disconnected before all robots achieve velocity consensus, resulting in a situation wherein robots in different connected components move further apart from each other over time.

Definition 7 (A Stable Flocking Motion) A stable flocking motion is defined as the phenomenon that a group of robots are spatially connected, form an α-lattice or a quasi α-lattice conformation, and reach velocity consensus.

3 Extended O-S Flocking Model and Problem Formulation

In this section, the CCE protocol along with the extended O-S flocking model is introduced for MRS under multi-hop ad hoc networks. The concerned problem is then formulated for multi-robot flocking.

3.1 CCE Protocol and Flocking Model

To achieve collision avoidance, velocity consensus, and spatial distribution control in flocking, robots repetitively update and execute control instructions based on their neighbors' and the leader's state information. As shown in Fig. 1, we develop a CCE protocol to describe this process. Within each period T, robot i needs to carry out three parallel sub-tasks: state information interaction (communication), control instruction updating (computation), and control instruction execution (execution), which are introduced below.

Control instruction execution. It means that robots move forward at a given speed and accel-

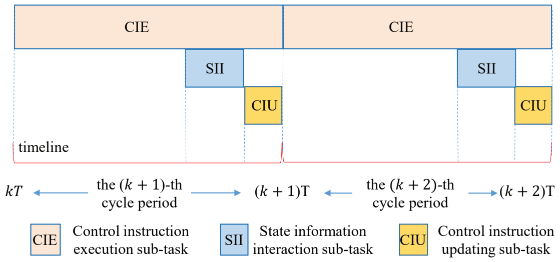
CIE
CIE
SII
SII
CIU
CIU
timeline
kT
the (k + 1)-th cycle period
(k + 1)T
the (k + 2)-th cycle period
(k + 2)T
CIE Control instruction execution sub-task
SII State information interaction sub-task
CIU Control instruction updating sub-task

Fig. 1 Illustration of communication-computation-execution cycles.

eration. At the end of the k-th cycle, i.e., $t = kT$, the position, speed, and acceleration of robot i are represented as $\boldsymbol{q}_i(k)$, $\boldsymbol{p}_i(k)$, and $\boldsymbol{u}_i(k)$, respectively. Then, within the $(k+1)$-th cycle, i.e., $t \in [kT, (k+1)T]$, robot i moves forward at the speed instruction $\boldsymbol{p}_i(k)$ and the acceleration instruction $\boldsymbol{u}_i(k)$. After executing these instructions for a cycle, robot i's position and speed are updated as

$$\begin{cases} \boldsymbol{q}_i(k+1) = \boldsymbol{q}_i(k) + \boldsymbol{p}_i(k) * T, \\ \boldsymbol{p}_i(k+1) = \boldsymbol{p}_i(k) + \boldsymbol{u}_i(k) * T. \end{cases} \tag{1}$$

Similarly, the state information of the virtual leader is updated as

$$\begin{cases} \boldsymbol{q}_r(k+1) = \boldsymbol{q}_r(k) + \boldsymbol{p}_r(k) * T, \\ \boldsymbol{p}_r(k+1) = \boldsymbol{p}_r(k) + \boldsymbol{u}_r(k) * T. \end{cases} \tag{2}$$

where $\boldsymbol{u}_r$ denotes the leader's acceleration.

State information interaction. As illustrated in Fig. 1, during the $(k+1)$-th cycle, after robot i advances for a certain duration, it is required to obtain LSI and NSI via wireless communications for control instruction updating. The NSI of robot i can be acquired through one-hop broadcast links. As for the LSI, the required relay hop count depends on the position and communication radius of robot i. If $i \in \Omega(k+1)$, then robot i can acquire LSI directly within the $(k+1)$-th cycle, and it needs to broadcast the acquired LSI to inform the other robots. If $i \notin \Omega(k+1)$, robot i is supposed to monitor the broadcast information in the wireless channel, so that it may receive the target LSI that is currently in the process of multi-hop broadcasting.

We use the variable H_i to represent the minimum relay hop count that is required for robot i to obtain LSI from the virtual leader. Let $\mathcal{V}_h$ denote the set of robots for which at least h hops are required to obtain LSI, i.e., $\mathcal{V}_h = \{i : i \in \mathcal{V}, H_i = h\}$. We denote the maximum value in set $\{H_i : i \in \mathcal{V}\}$ as H_M. Then, $h \in \{0, 1, ..., H_M\}$. As shown in Fig. 2, when $i \in \mathcal{V}_0$, it can acquire LSI directly, i.e., $\mathcal{V}_0 = \Omega$. When $i \in \mathcal{V}_1$, then $\exists j \in \mathcal{V}_0$ satisfies $(i,j) \in \mathcal{E}$. That is, robot i can communicate with some robots that can acquire LSI directly. When $i \in \mathcal{V}_2$, then $\exists j \in \mathcal{V}_1$ satisfies $(i,j) \in \mathcal{E}$.

Due to the unstable factors in the communication environment, such as electromagnetic interference, multi-path effects, and data conflicts, the transmission success over each hop becomes a random event. Since different types of information are typically transmitted with different protocols, the transmission of NSI and LSI over each hop exhibits different degrees of success probability. We use p_s and p_b to represent the STP of NSI and LSI, respectively, satisfying $0 \le p_s \le 1$ and $0 \le p_b \le 1$. Further, we use l_{ij} to indicate if transmission succeeds or not between robots i and j. Then l_{ij} is a random variable following the Bernoulli distribution, i.e., $\Pr(l_{ij}{=}1) = p_s$ and $\Pr(l_{ij} = 0) = 1 - p_s$. Let h_i indicate whether robot i successfully obtains LSI or not, which is also a Bernoulli random variable. Since the LSI must be relayed by H_i hops before it reaches robot i, robot i can only acquire the LSI if the transmission over all hops is successful. Therefore, h_i satisfies

$$\begin{cases} \Pr(h_i = 1) = p_b^{H_i}, \\ \Pr(h_i = 0) = 1 - p_b^{H_i}. \end{cases} \tag{3}$$

Control instruction updating. At the end of each cycle, robot i needs to update the control instruction $\boldsymbol{u}_i$ with the obtained LSI and NSI. The updating algorithm of control instructions is closely related to collision avoidance, velocity consensus, swarm cohesion, and the spatial distribution adjustment of robots. A well-known and efficient model is the one proposed by Olfati-Saber (2006), and is designed for the continuous-time multi-agent flocking system guided by the virtual leader. Considering the discrete-time characteristics of MRS as well as the STP of NSI and LSI, the Olfati-Sabel control model in (Olfati-Saber, 2006) is modified as

$$\begin{aligned} \boldsymbol{u}_i(k) = &\sum_{j \in \mathcal{N}_i(k)} l_{ij}(k) \cdot k_1(\|\boldsymbol{q}_i(k) - \boldsymbol{q}_j(k)\| - d_e)\boldsymbol{n}_{ij}(k) \\ &+ \sum_{j \in \mathcal{N}_i(k)} l_{ij}(k) \cdot k_2 \cdot (\boldsymbol{p}_j(k) - \boldsymbol{p}_i(k)) \\ &- h_i(k)[c_1(\boldsymbol{q}_i(k) - \boldsymbol{q}_r(k)) + c_2(\boldsymbol{p}_i(k) - \boldsymbol{p}_r(k))], \end{aligned} \tag{4}$$

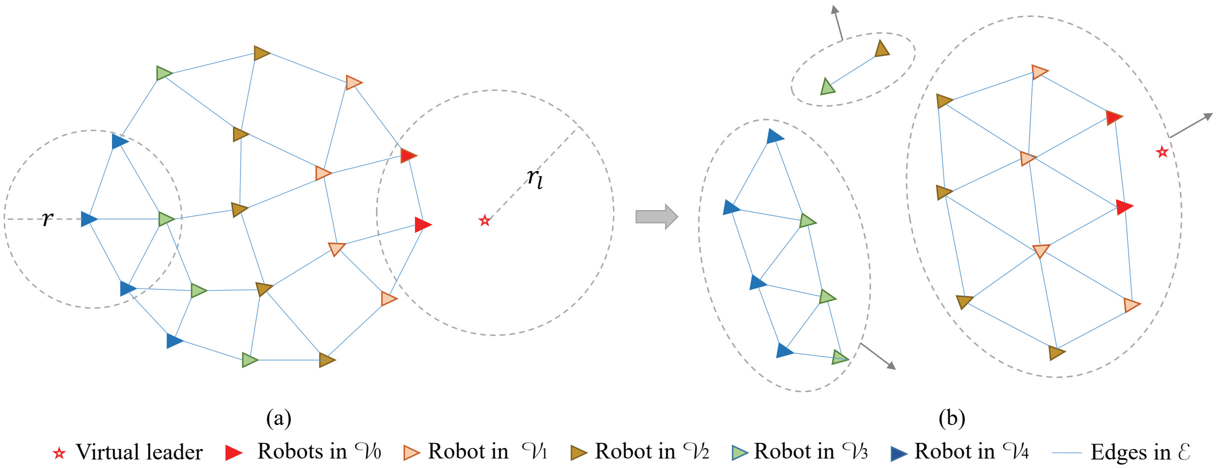
r
r_l
(a)
(b)
☆ Virtual leader
▶ Robots in $\mathcal{V}_0$
▷ Robot in $\mathcal{V}_1$
▶ Robot in $\mathcal{V}_2$
▷ Robot in $\mathcal{V}_3$
▶ Robot in $\mathcal{V}_4$
— Edges in $\mathcal{E}$

Fig. 2 (a) Illustration of the spatially induced graph, where each triangle represents a robot and its vertex angle's orientation represents the motion direction of this robot. (b) Illustration of the flocking fragmentation phenomenon.

where $i \in \mathcal{V}$, and k_1, k_2, c_1, and c_2 are the controller gains and all of them are larger than 0.

In Eq. (4), $\boldsymbol{n}_{ij}(k) = \frac{\boldsymbol{q}_j(k)-\boldsymbol{q}_i(k)}{\|\boldsymbol{q}_i(k)-\boldsymbol{q}_j(k)\|}$. The first term of Eq. (4) is intended for adjusting the distance between robot i and its neighbors to the desired value d_e. If $j \in \mathcal{N}_i(k)$ and the distance between i and j is smaller than d_e, then this term acts as a repulsive force to make i get away from j, which can realize collision avoidance. Otherwise, it acts as an attractive force to shorten the distance between i and j, which is beneficial for swarm cohesion. Let $d_e/r = 1+\epsilon$. It is noted that $\epsilon \ll 1$ and we set ϵ as 0.2 according to Olfati-Saber (2006) for the following analysis. The second term of Eq. (4) is used to realize the velocity consensus of robots. The third term drives robot i to track the virtual leader's motion state.

3.2 Problem formulation

A stable flocking is always desired when a group of movable robots perform tasks cooperatively, and the premise of stability is that no fragmentation appears during flocking. However, an unstable transmission environment makes flocking fragmentation become a possible event, especially when a considerable number of robots cannot obtain the LSI. Obviously, the acquisition of LSI for each robot is closely related to the required minimum relay hop counts and STP. With the increase of H_i or the decrease of p_b, robot i will have less success in obtaining LSI, which may increase the FFP. Understanding the effects of network topology features on FFP can help to choose appropriate network parameters for the global broadcast of LSI, so that flocking fragmentation can be prevented. In this paper, we devote to determine the relationship between network topology parameters and the flocking fragmentation probability given the system states, i.e.,

$$p_f = f(\text{system states, network topology features}). \quad (5)$$

Determining Eq. (5) is essentially a mathematical modeling problem of constructing the function $f(\cdot)$, which takes the system/network parameters as inputs and produces p_f as the output. To solve this problem, it is required to identify the input parameters and the function $f(\cdot)$. However, multi-robot flocking is a high-dimensional, discrete-time, nonlinear, and dynamic process, and an unstable environment intensifies the randomness of this process. These features make it difficult to precisely predict the final state of systems, and also make the traditional stability analysis methods inapplicable, such as the Lyapunov stability theory (Olfati-Saber, 2006; Sastry, 1999), spectral graph theory (Chung, 1997; Patterson et al., 2010), and cell mapping method (Sun et al., 2018). Hence, we take two steps to analyze this problem, progressing from special multi-robot flocking scenarios to generally complex ones.

As the first step, we consider a particular flocking scenario with specific initial states, and the LSI's and NSI's random losses are not considered. These conditions make it possible to construct the function between hop count features and fragmentation phenomena through theoretical derivation. Specifically, an FPM is proposed based on the layer division and spatial linearization of robot swarms, which can assist in extracting the key initial system/network state features that will affect the final trend of flocking swarms. Further, we focus on a general flocking scenario with random initial states, and the impacts of LSI's and NSI's STP are considered. The consideration of STP leads to random and dynamic changes in network topology and spatial distribution during multi-robot flocking. To capture these characteristics, a data fitting model based on a BP neural network is proposed, which is trained by feeding the above-extracted system/network state features as input and can formulate the function $f(\cdot)$.

Due to the complexity and randomness of fragmentation formulation, an assumption is employed in the following analysis that the acceleration of the virtual leader is 0. This assumption eliminates the impacts of the variable leader's speed on fragmentation, which can help us concentrate on analyzing the correlation between the communication parameters and the fragmentation phenomenon. Although the analysis is conducted under such an ideal assumption, its results can still act as a reference boundary of the communication network design or optimization for preventing fragmentation in the scenario where the acceleration of virtual leader is not 0.

4 Fragmentation Analysis for Specific Flocking Scenarios

We first consider a particular flocking scenario in a two-dimensional plane. As shown in Fig. 3, at the initial time, i.e., when $k = 0$, the system states contain the following features: (1) robots are arranged in the square grid points with the number of rows and columns equal to S_r and S_c, respectively, and all grids are spliced together into a rectangle; (2) all robots form an α-lattice pattern with the *scale* equal to d_0; (3) one of the rectangle's symmetric axes is set as the x-axis, and the virtual leader is on the x-axis and ahead of all robots; (4) the robots in the column closest to the leader can obtain LSI directly, while others should obtain LSI via one-hop or multi-hop communication links; (5) the initial speed of all robots is equal to $[p_x(0), 0]^T \in \mathbb{R}^2$, and the virtual leader's speed is $[p_{rx}(0), 0]^T \in \mathbb{R}^2$. Based on the above descriptions, the whole system will move forward along the x-axis. We assume a static network topology during the motion process to focus on the effect of hop count features, while the effect of random link failure is left to be analyzed in the next section. Under this assumption, the neighbor set of each robot is regarded to be unchanged during subsequent analysis.

4.1 Introduction of the FPM

Under the abovementioned initial states, we perform numerous simulation experiments and observe the following phenomenon: classifying robots into different layers according to the required minimum relay hops, the fragmentation is more likely to appear between adjacent robot layers. A similar example is provided in Fig. 2(b), where the second layer almost loses all connections with the third layer, so that fragmentation appears. This phenomenon can primarily be attributed to the fact that robots in the same layer tend to have a similar probability of obtaining the state information from the leader. Conversely, different layers of robots are affected by the leader to different degrees. This causes the robots in the same set $\mathcal{V}_h$ to have comparable motion tendency, promoting their connectivity. Meanwhile, it makes robots in different layers have larger velocity deviation, leading to an increased likelihood of fragmentation between them. Given that robots in the same set $\mathcal{V}_h$ have similar motion tendencies, they can be regarded as a whole to analyze flocking fragmentation. Building on this insight, we propose the FPM model based on the ideas of layer division and spatial linearization of multi-robot swarm.

As shown in Figs. 3(a) and 3(b), all robots are classified into different layers according to the required minimum relaying hops for obtaining LSI. The h-th layer corresponds to the robot set $\mathcal{V}_h$. Let n_h denote the size of $\mathcal{V}_h$, and a_h denote the number of robots that can obtain LSI in $\mathcal{V}_h$. Thus, $a_h \leq n_h$. Based on the static topology assumption, n_h and a_h can be regarded to be unchanged during the flocking process. As robots in each layer exhibit similar motion tendencies, we treat them as a sub-swarm and represent their overall motion features by using the speed and position of the corresponding centroid M_h, as shown in Fig. 3(b) and Fig. 3(c). Since the entire system is moving along the x-axis, multiple

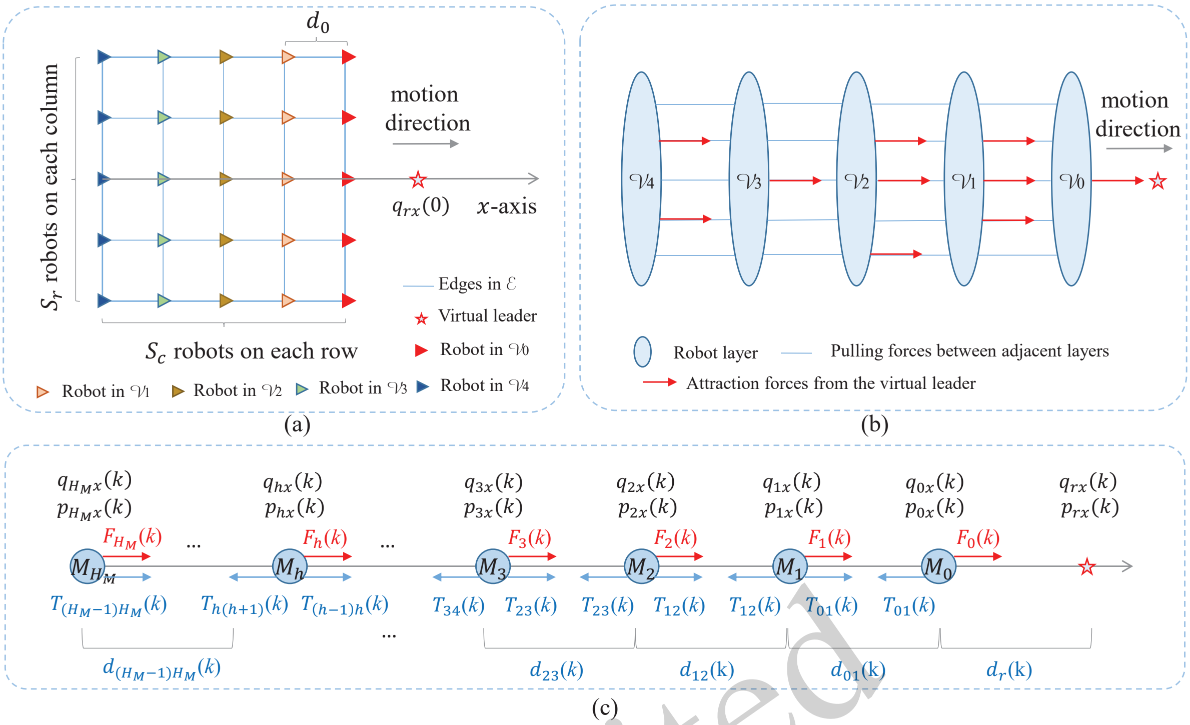

Fig. 3 (a) Initial system states for the simplified flocking scenarios. (b) Equivalent robot layers and the force illustration of each layer. (c) The force illustration for the equivalent centroid points in each layer.

sub-swarms can be simplified as multiple centroid points moving along the x-axis. The equivalence from layers in the two-dimensional plane to points in the one-dimensional line makes fragmentation more tractable to analyze. Next, we explain the other symbols in Fig. 3(c).

At time $t = kT$, the position and speed of the leader are denoted by $[q_{rx}(k), 0]^T$ and $[p_{rx}(k), 0]^T$, respectively. The position, speed, and acceleration of the centroid M_h are represented by $[q_{hx}(k), 0]^T$, $[p_{hx}(k), 0]^T$, and $[u_{hx}(k), 0]^T$, respectively. Then, according to Eqs. (1) and (2),

$$\begin{cases} q_{rx}(k+1) = q_{rx}(k) + p_{rx}(k) * T, \\ p_{rx}(k+1) = p_{rx}(k), \\ q_{hx}(k+1) = q_{hx}(k) + p_{hx}(k) * T, \\ p_{hx}(k+1) = p_{hx}(k) + u_{hx}(k) * T. \end{cases} \tag{6}$$

Let $d_{h(h+1)}(k)$ and $v_{h(h+1)}(k)$ denote the spacing and velocity difference between points M_h and M_{h+1}, respectively. Let $d_r(k)$ and $v_r(k)$ denote the spacing and velocity difference between the virtual leader and the point M_0, respectively. Thus

$$\begin{cases} d_{h(h+1)}(k) = q_{hx}(k) - q_{(h+1)x}(k), \\ v_{h(h+1)}(k) = p_{hx}(k) - p_{(h+1)x}(k), \\ d_r(k) = q_{rx}(k) - q_{0x}(k), \\ v_r(k) = p_{rx}(k) - p_{0x}(k). \end{cases} \tag{7}$$

Based on the above introduction, no fragmentation appearing in the swarm is equivalent to the condition that $d_{h(h+1)}$ keeps smaller than r for all $h \in \{0, 1, ..., H_M - 1\}$ during the whole flocking process. Then the fragmentation analysis is converted to the derivation of $d_{h(h+1)}(k)$, which requires us to determine the state updating equation of $d_{h(h+1)}(k)$.

Combining Eqs. (6) and (7), it is obtained that

$$\begin{cases} d_r(k+1) = d_r(k) + v_r(k) * T, \\ v_r(k+1) = v_r(k) - u_{0x}(k) * T, \\ d_{h(h+1)}(k+1) = d_{h(h+1)}(k) + v_{h(h+1)}(k) * T, \\ v_{h(h+1)}(k+1) = v_{h(h+1)}(k) + (u_{hx}(k) - u_{(h+1)x}(k))T. \end{cases} \tag{8}$$

Let $\boldsymbol{s}_{dv}(k) = [d_r(k), v_r(k), d_{01}(k), v_{01}(k), d_{12}(k), v_{12}(k), ..., d_{(H_M-1)H_M}(k), v_{(H_M-1)H_M}(k)]^T \in \mathbb{R}^{2(H_M+1)}$ and vector $\boldsymbol{u}_x(k) = [u_{0x}(k), u_{1x}(k), ..., u_{(H_M)x}(k)]^T \in \mathbb{R}^{H_M+1}$. Thus we have

$$\boldsymbol{s}_{dv}(k+1) = \mathbf{J} \cdot \boldsymbol{s}_{dv}(k) + \mathbf{K} \cdot \boldsymbol{u}_x(k) \cdot T, \tag{9}$$

where $\mathbf{J}{\in}\mathbb{R}^{2(H_M+1)*2(H_M+1)}$ and $\mathbf{K}{\in}\mathbb{R}^{2(H_M+1)*(H_M+1)}$.

$$\mathbf{J} = \mathbf{I}_{(H_M+1)} \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}.$$

Hence, the discrete-time state updating equation of $\boldsymbol{s}_{dv}$ can be derived after formulating $\boldsymbol{u}_x$, which is presented in the following theorem.

Theorem 1 The discrete-time state updating equation of $\boldsymbol{s}_{dv}$ is expressed by

$$\boldsymbol{s}_{dv}(k+1) = \mathbf{A} \cdot \boldsymbol{s}_{dv}(k) - \mathbf{B}, \tag{10}$$

where $\mathbf{A} \in \mathbb{R}^{2(H_M+1)*2(H_M+1)}$ and $\mathbf{B} \in \mathbb{R}^{2(H_M+1)}$. The expressions of $\mathbf{A}$ and $\mathbf{B}$ are

$$\begin{cases} \mathbf{A} = \mathbf{J} + \mathbf{K} \cdot \mathbf{F} \cdot \mathbf{D} \cdot T + \mathbf{K} \cdot \mathbf{G} \cdot \mathbf{C} \cdot T, \\ \mathbf{B} = \mathbf{K} \cdot \mathbf{G} \cdot k_1 \cdot d_e \cdot \boldsymbol{e} \cdot T, \end{cases}$$

where $\mathbf{C}$, $\mathbf{D}$, $\boldsymbol{e}$, $\mathbf{F}$, and $\mathbf{G}$ are displayed as below:

$$\mathbf{C}{=}\begin{bmatrix} 0 & 0 & k_1e_{01} & k_2e_{01} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & k_1e_{12} & k_2e_{12} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & k_1e_{(H_M-1)H_M} & k_2e_{(H_M-1)H_M} \end{bmatrix},$$

$$\mathbf{D}{=}\begin{bmatrix} c_1a_0 & c_2a_0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ c_1a_1 & c_2a_1 & c_1a_1 & c_2a_1 & 0 & 0 & \cdots & 0 \\ c_1a_2 & c_2a_2 & c_1a_2 & c_2a_2 & c_1a_2 & c_2a_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1a_{H_M} & c_2a_{H_M} & c_1a_{H_M} & c_2a_{H_M} & c_1a_{H_M} & c_2a_{H_M} & \cdots & c_2a_{H_M} \end{bmatrix},$$

$$\boldsymbol{e} = \begin{bmatrix} e_{01} & e_{12} & e_{23} & \cdots & e_{(H_M-2)(H_M-1)} & e_{(H_M-1)(H_M)} \end{bmatrix}^T,$$

$$\mathbf{G}{=}\begin{bmatrix} -\frac{1}{n_0} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{1}{n_1} & -\frac{1}{n_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{n_2} & -\frac{1}{n_2} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{n_{(H_M-1)}} & -\frac{1}{n_{(H_M-1)}} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{n_{H_M}} \end{bmatrix},$$

$$\mathbf{F} = \text{diag}([\frac{1}{n_0}, \frac{1}{n_1}, ..., \frac{1}{n_h}, ..., \frac{1}{n_{(H_M-1)}}, \frac{1}{n_{H_M}}]).$$

The expression diag($\boldsymbol{v}$) represents a matrix with diagonal elements forming the vector $\boldsymbol{v}$ while the other elements are all 0. It is noted that $\mathbf{C} \in \mathbb{R}^{H_M*2(H_M+1)}$, $\mathbf{D} \in \mathbb{R}^{(H_M+1)*2(H_M+1)}$, $\boldsymbol{e} \in \mathbb{R}^{H_M}$, $\mathbf{F} \in \mathbb{R}^{(H_M+1)*(H_M+1)}$, and $\mathbf{G} \in \mathbb{R}^{(H_M+1)*H_M}$.

Proof. From Eq. (9), $\boldsymbol{s}_{dv}$ is dependent on $\boldsymbol{u}_x$ when the initial state vector $\boldsymbol{s}_{dv}(0)$ is given. Therefore, it is required to determine each element in $\boldsymbol{u}_x$. To solve this, we first conduct the force analysis on the centroid M_h to formulate $u_{hx}(k)$, since $u_{hx}(k)$ is equal to the resultant force of the centroid divided by its mass according to Newton's Second Law (Gundlach et al., 2007). The force undertaken by M_h is supposed to be the resultant force of all external forces acting on each robot in $\mathcal{V}_h$. Each robot is influenced by the virtual leader and its adjacent neighbors during flocking, and we regard the influence from different origins as distinct forces. Then, according to Eq. (4), robot i undergoes an attractive force from the leader and a pulling force from the neighbors, which are denoted by $\boldsymbol{f}_i$ and $\boldsymbol{t}_i$, respectively. We denote the mass of each robot as m. Then the expressions of $\boldsymbol{f}_i$ and $\boldsymbol{t}_i$ are given as

$$\begin{cases} \boldsymbol{f}_i(k){=}h_i(k)[c_1(\boldsymbol{q}_r(k){-}\boldsymbol{q}_i(k)){+}c_2(\boldsymbol{p}_r(k)-\boldsymbol{p}_i(k))]m, \\ \boldsymbol{t}_i(k){=}[\sum\limits_{j\in\mathcal{N}_i(k)} l_{ij}(k)k_1(\|\boldsymbol{q}_i(k)-\boldsymbol{q}_j(k)\|{-}d_e)\boldsymbol{n}_{ij}(k) \\ \qquad + \sum\limits_{j\in\mathcal{N}_i(k)} l_{ij}(k)k_2(\boldsymbol{p}_j(k)-\boldsymbol{p}_i(k))]m. \end{cases}$$

Similarly, the centroid M_i also feels two kinds of external forces: (1) the attraction force from the leader; (2) the pulling force from adjacent centroid points.

As shown in Fig. 3(c), the centroid M_0 is attracted by the virtual leader and pulled by the centroid M_1, and the corresponding forces are represented by $F_0(k)$ and $T_{01}(k)$, respectively. Since there are a_0 individuals that can obtain LSI in $\mathcal{V}_0$ and the motion features of each one are equivalently replaced by the speed and position of M_0, the expression of $F_0(k)$ is

$$\begin{aligned} F_0(k) &{=}a_0[c_1(q_{rx}(k){-}q_{0x}(k)){+}c_2(p_{rx}(k){-}p_{0x}(k))]m \\ &= a_0[c_1d_r(k) + c_2v_r(k)]m. \end{aligned}$$

The pulling force $T_{01}(k)$ acting on the centroid M_0 is equal to the resultant force of the forces between adjacent individuals in sets $\mathcal{V}_0$ and $\mathcal{V}_1$. Specifically, for robots $i \in \mathcal{V}_0$ and $j \in \mathcal{V}_1$, if edge $(i, j) \in \mathcal{E}$, then robot i and robot j can interact with each other, and the interaction between them is a component of the resultant pulling force $T_{01}(k)$. Let $\mathcal{E}_{01}$ denote the interaction set between layers $\mathcal{V}_0$ and $\mathcal{V}_1$, so that $\mathcal{E}_{01} = \{(i, j) : (i, j) \in \mathcal{E}, \forall i \in \mathcal{V}_0, j \in \mathcal{V}_1\}$. The size of $\mathcal{E}_{01}$ is represented by e_{01}. Still employing the speed and position of the centroid to equivalently replace the motion states of robots in the corresponding layer, we obtain the expression of $T_{01}(k)$ as below:

$$\begin{aligned}T_{01}(k) &= e_{01}\cdot[k_1\cdot(q_{0x}(k)-q_{1x}(k)-d_e)\\&\quad+k_2\cdot(p_{0x}(k)-p_{1x}(k))]\cdot m\\&= e_{01}\cdot[k_1\cdot(d_{01}(k)-d_e)+k_2\cdot v_{01}(k)]\cdot m.\end{aligned}$$

The directions of $F_0(k)$ and $T_{01}(k)$ are displayed in Fig. 3(c), so that the acceleration of M_0 is

$$u_{0x}(k)=\frac{F_0(k)-T_{01}(k)}{n_0 * m},$$

where $n_0 * m$ is the mass of M_0.

Similarly, the centroid M_1 undertakes the attraction force $F_1(k)$ from the leader, the pulling force $T_{01}(k)$ from the robot layer $\mathcal{V}_0$, and the pulling force $T_{12}(k)$ from the robot layer $\mathcal{V}_2$. The directions of the three forces are illustrated in Fig. 3(c), and their expressions are shown below.

$$\begin{aligned}F_1(k)&=a_1[c_1(q_{rx}(k)-q_{1x}(k)+c_2(p_{rx}(k)-p_{1x}(k))]m\\&=a_1[c_1(d_r(k)+d_{01}(k))+c_2(v_r(k)+v_{01}(k))]m.\end{aligned}$$

$$\begin{aligned}T_{12}(k) &= e_{12}\cdot[k_1\cdot(q_{1x}(k)-q_{2x}(k)-d_e)\\&\quad+k_2\cdot(p_{1x}(k)-p_{2x}(k))]\cdot m\\&= e_{12}[k_1(d_{12}(k)-d_e)+k_2v_{12}(k)]m.\end{aligned}$$

The corresponding acceleration is

$$u_{1x}(k)=\frac{F_1(k)+T_{01}(k)-T_{12}(k)}{n_1 * m}.$$

Likewise, the acceleration of M_h is

$$u_{hx}(k)=\begin{cases}\dfrac{F_h(k)-T_{h(h+1)}(k)}{n_h\cdot m},h=0\\ \dfrac{F_h(k)+T_{(h-1)(h)}(k)-T_{h(h+1)}(k)}{n_h\cdot m},0<h<H_M\\ \dfrac{F_h(k)+T_{(h-1)(h)}(k)}{n_h\cdot m},h=H_M,\end{cases}\tag{11}$$

where the expressions of $F_h(k)$ and $T_{h(h+1)}(k)$ are

$$\begin{cases}F_h(k)=a_h[c_1\cdot(d_r(k)+\displaystyle\sum_{j=0}^{h-1}d_{j(j+1)}(k))\\ \qquad\qquad+c_2\cdot(v_r(k)+\displaystyle\sum_{j=0}^{h-1}v_{j(j+1)}(k))]m,\\ T_{h(h+1)}(k)=e_{h(h+1)}[k_1\cdot(d_{h(h+1)}(k)-d_e)\\ \qquad\qquad+k_2\cdot v_{h(h+1)}(k)]m.\end{cases}\tag{12}$$

According to Eqs. (11) and (12),

$$\begin{aligned}\boldsymbol{u}_x(k)&=[u_{0x}(k),u_{1x}(k),...,u_{(H_M)x}(k)]^T\\&=\frac{\mathbf{F}}{m}[F_0(k),F_1(k),...,F_{H_M-1}(k),F_{H_M}(k)]^T\\&\quad+\frac{\mathbf{G}}{m}[T_{01}(k),T_{12}(k),...,T_{(H_M-1)H_M}(k)]^T\\&=\frac{\mathbf{FD}\boldsymbol{s}_{dv}(k)m}{m}+\frac{\mathbf{G}(\mathbf{C}\boldsymbol{s}_{dv}(k)m-k_1d_em\boldsymbol{e})}{m}\\&=\mathbf{F}\cdot\mathbf{D}\cdot\boldsymbol{s}_{dv}(k)+\mathbf{G}(\mathbf{C}\cdot\boldsymbol{s}_{dv}(k)-k_1d_e\boldsymbol{e}).\end{aligned}\tag{13}$$

By substituting Eq. (13) into Eq. (9), the updating equation of $\boldsymbol{s}_{dv}$ in Eq. (10) can be obtained. □

Remark 1. *As demonstrated by Theorem 1, except for the cycle duration T, two types of parameters have a direct impact on the updating of vector $\mathbf{s}_{dv}$: (1) controller parameters including k_1, k_2, d_e, c_1, and c_2; and (2) network topology parameters including n_h, a_h, and $e_{h(h+1)}$ for $\forall h=\{0,1,...,H_M\}$. The controller parameter should constrain the stability conditions to guarantee the flocking stability under an ideal communication environment (i.e, $p_s=p_b=1$). Since the stability analysis is not a concern of the present research, it is not presented in this article; it is, however, derived in another study undertaken by us that is awaiting publication. The network parameters H_M, n_h and $e_{h(h+1)}$ can be determined based on the initial system states. That is, robots are arranged in square grids with size $S_r\times S_c$ and form an α-lattice as shown in Fig. 3(a) so that $H_M=S_c-1$, $n_h=S_r$ and $e_{h(h+1)}=S_r$.*

Remark 2. *According to the initial system states in Fig. 3a and the idea of spatial linearization, $\mathbf{s}_{dv}(0)$ is equal to $[q_{rx}(0)-q_{0x}(0),p_{rx}(0)-p_{0x}(0),d_0,0,d_0,0,...,d_0,0]^T$.*

Remark 3. *When $\boldsymbol{s}_{dv}(0)$ and $[a_0,a_1,...,a_{H_M}]^T$ are presented, the state vector $\boldsymbol{s}_{dv}(k)$ can be calculated for each time step $k\in\{0,1,2,...\}$. Then the spacing distance $d_{h(h+1)}(k)$ between any two adjacent robot layers $\mathcal{V}_h$ and $\mathcal{V}_{h+1}$ in each cycle can be obtained, which can be employed to judge if flocking fragmentation appears. The detailed judgement model, referred to as FPM, is displayed in Table 1.*

4.2 Validation of the FPM

We take a concrete example to validate the accuracy of the proposed FPM model. Let $S_r=5$, $S_c=5$. The parameter d_0 is set to a value larger than

Table 1 Illustration of FPM based on Theorem 1

```
Fragmentation Prediction Model
Input: $S_r$, $S_c$, $r$, $d_0$, $q_{rx}(0)$, $q_{0x}(0)$, $p_{rx}(0)$, $p_{0x}(0)$,
       $[a_0, a_1, ..., a_{H_M}]^T$.
Output: fragmentation indicator $f_r$.
Initialize: $k = 0$ and $f_r = 0$.
Process:
  1. determine $n_h$ and $e_{h(h+1)}$ based on Remark 1;
  2. calculate $\boldsymbol{s}_{dv}(0)$ based on Remark 2;
  3. while $k \leq 500$ :
       calculate $\boldsymbol{s}_{dv}(i+1)$ based on Theorem 1;
       for j=1:1:$H_M$
           if ($\boldsymbol{s}_{dv}(2j+1) > r$):
             $f_r$=1;
             return: $f_r$ .
           endif
       end
       $k = k + 1$;
     end
return: $f_r$ ($f_r = 1$ means fragmentation happens,
        and otherwise, it means that fragmenta-
        -tion does not happen.)
```

$\frac{\sqrt{2}r}{2}$ but smaller than r. From Fig. 3(a), the maximum hop count H_M can be determined as 4, while n_h and $e_{h(h+1)}$ for $h = \{0, 1, 2, 3, 4\}$ are $[5, 5, 5, 5, 5]$ and $[5, 5, 5, 5]$, respectively. Based on the initial state parameters $q_{rx}(0)$, $q_{0x}(0)$, $p_{rx}(0)$ and $p_{0x}(0)$, $\boldsymbol{s}_{dv}(0)$ can be obtained as $[d_r(0), v_r(0), d_0, 0, d_0, 0, d_0, 0, d_0, 0]^T$, where $d_r(0) = q_{rx}(0) - q_{0x}(0)$ and $v_r(0) = p_{rx}(0) - p_{0x}(0)$. Since robots in the column closest to the leader can obtain LSI directly, a_0 is equal to 5, so that the fragmentation result can be predicted after a_1, a_2, a_3, and a_4 are presented. Variables a_1, a_2, a_3, and a_4 all satisfy that $a_h \in \{0, 1, 2, ..., n_h\}$, so that there are a total of $\prod_{h=1}^{H_M}(n_h + 1)$ value situations for the set $[a_0, a_1, a_2, a_3, a_4]$. For each value situation, we can obtain the model-based and simulation-based fragmentation prediction results. If these two results are the same, then the prediction is correct. The comparison can be conducted for all value situations of $[a_0, a_1, a_2, a_3, a_4]$, and the total number of correct predictions can be obtained, which is denoted by *cp*. Then, the prediction accuracy of model is defined as

$$a_c = \frac{cp}{\prod_{h=1}^{H_M}(n_h + 1)}.$$

For a given $[a_0, a_1, a_2, a_3, a_4]$, the simulation process is displayed in Table 2. Related parameters are $k_1 = 0.4, k_2 = 0.2, c_1 = 0.04, c_2 = 0.4, T = 0.3, p_x(0) = 4$, $d_e = 10$ and $r = 1.2 * d_e$.

Table 2 Illustration of simulation process

```
Simulation of Predicting Fragmentation
Input: $k_1$, $k_2$, $c_1$, $c_2$, $T$, $d_e$, $S_r$, $S_c$, $d_0$, $q_{rx}(0)$,
       $p_{rx}(0)$, $p_x(0)$, $[a_0, a_1, ..., a_{H_M}]^T$.
Output: fragmentation indicator $f_r$.
Initialize: $k = 0$ and $f_r = 0$.
Process:
  1. generate the initial system states based
     on $S_r$, $S_c$, $d_0$, $q_{rx}(0)$, $p_{rx}(0)$, $p_x(0)$;
  2. determine the robot sets $\mathcal{V}_h$
     for $h = \{0, 1, ..., H_M\}$ ;
  3. while $k \leq 500$ :
       1) each robot acquires the NSI;
       2) robots in $\mathcal{V}_0$ acquire the LSI;
       for $h$=1:1:$H_M$
           randomly choose $a_h$ robots from $\mathcal{V}_h$
           and these robots can obtain LSI;
       endfor
       3) each robot calculates the acceleration
          instruction based on Eq. (4);
       4) robots and leader move forward for
          a duration $T$ and then update its
          state based on Eqs. (1) and (2);
       5) identify if fragmentation appears
          based on Definition 6;
       if (fragmentation appears):
           $f_r$=1;
           return: $f_r$
       endif
       6) $k = k + 1$;
     end
return: $f_r$ ($f_r = 1$ means fragmentation happens.
        And otherwise, it means that fragmenta-
        -tion does not happen) .
```

Fig. 4 displays the model's prediction accuracy under different sets of $[d_0, d_r(0), v_r(0)]$. In Fig. 4, the color bar in each subgraph indicates the range of the accuracy values. The grid with lighter color means that the FPM has higher accuracy under the corresponding set of $[d_0, d_r(0), v_r(0)]$. From the four graphs presented in Figs. 4(a)-4(d), we ascertain that a large fraction of the prediction accuracy values keeps within the interval $[0.7, 0.8]$ under a wide range of parameter settings. This implies that our FPM analysis has a certain degree of precision and the extracted key initial system or network state features do affect the final trend of flocking swarms. However, it should also be noted that there is still room to improve the accuracy, since the proposed method neglects the dynamic change of network topology and spatial distribution. Thus, the following sections will further involve these factors to construct a more accurate model for fragmentation formulation.

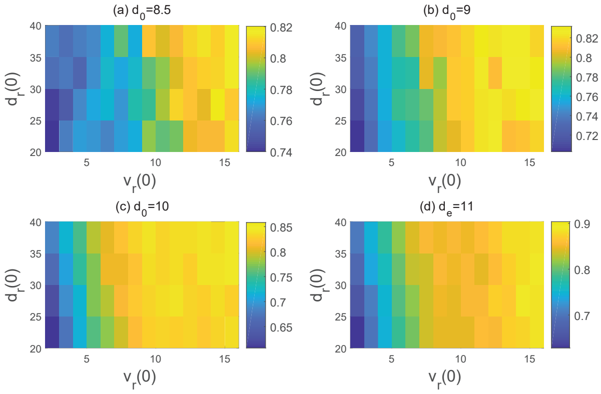

Fig. 4 Prediction accuracy of FPM under different sets of $[d_0, d_r(0), v_r(0)]$.

5 Fragmentation Probability Formulation for General Flocking Scenario

In this section, we focus on a more general flocking scenario where the random and dynamic nature of communication networks is considered. To formulate the relationship between FFP and network states, a data fitting method based on a BP neural network is proposed. We first extract the input features for the BP neural network by analyzing the proposed FPM, and then introduce the network architecture as well as the training method.

5.1 The General Flocking Scenario

Different from the specified scenario where robots are regularly arranged in square grids, the positions of robots in a general flocking scenario are not strictly restricted. At the initial time (i.e., $k = 0$), N robots are uniformly distributed on the two-dimensional plane and form an α-lattice with the *scale* equal to d_0. The speed of robot i is expressed as $\boldsymbol{p}_i(0) = [p_{ix}(0), 0] \in \mathbb{R}^2$, where $p_{ix}(0)$ is a random variable following an uniform distribution over the interval $[\overline{v} - \delta_v, \overline{v} + \delta_v]$. The parameter $\overline{v}$ represents the average speed along the x-axis, while δ_v indicates the speed deviation, which reflects the group's velocity discrepancy. The position of the virtual leader is represented as $\boldsymbol{q}_r(0) = [\max(q_{ix}(0)) + dr, \frac{1}{N}\sum_{i=1}^{N} q_{iy}(0)]$, which means the leader is positioned ahead of the robot with the maximum x-axis coordinate by a distance of dr, while in the y-axis direction, it is located at the center of all robots. The speed of the leader is represented by $\boldsymbol{p}_r(0) = [\overline{v} + v_r, 0]$, where v_r denotes the speed deviation between the leader and the whole robot swarm. All robots proceed according to the CCE protocol, and the control algorithm is based on Eq. (4).

In this scenario, the STP of LSI and NSI (i.e., p_b and p_s) is considered, which means that the assumption of a static network topology is excluded. For such a flocking scenario, the final state is impossible to analytically predict even if the initial states of all robots and the virtual leader (i.e., $\boldsymbol{q}_r(0)$, $\boldsymbol{p}_r(0)$, $\boldsymbol{q}(0)$, and $\boldsymbol{p}(0)$) are known, since the uncertainty and randomness of information interaction lead to a large number of possibilities for the SIG of the final swarm state. At time $t = kT$, all possible SIGs construct the SIG space, denoted by $\varPhi(k)$, where the possible fragmented SIGs form the set $\varUpsilon(k)$. Then the FFP at kT is $|\varUpsilon(k)|/|\varPhi(k)|$. Since the sets $\varPhi(k)$ and $\varUpsilon(k)$ are dependent on the initial system states as well as the network parameters, there is bound to be a function relationship between FFP and system/network state parameters. Due to the high-dimensional, nonlinear, and random features of flocking, we choose the BP neural network to formulate the function, which is introduced below.

5.2 Input Feature Extraction for BP-based FFP Formulation

To formulate the FFP calculation function $f(\cdot)$, the first step is to determine the input vector, denoted as $\boldsymbol{x}$. The input data should be selected based on two aspects: (1) parameters about the initial system state; and (2) parameters about the network features. For the former aspect, a common way is to directly take the initial position and speed of robots and leader to indicate the system state. However, this incurs two problems. On the one hand, the dimension of vector $\boldsymbol{x}$ relies on the total number of robots (i.e., N), so that if we want to determine the dimension of $\boldsymbol{x}$ we have to set N to a fixed value, which renders function $f(\cdot)$ unable to generalize to different sizes of robot swarms. On the other hand, it may increase the complexity of neural network architecture and decrease the training efficiency, since it is the velocity discrepancy and SIG's structure that affect FFP rather than the absolute position or speed of the robots and leader. Considering these factors, it is expected to extract key state features for the vector $\boldsymbol{x}$ from the original position and speed data. Since the FPM has extracted and analyzed key state features that impact fragmentation, we utilize them to construct the input vector as

$$\boldsymbol{x}=[\boldsymbol{x}_{HCD}^T,p_b,N,\boldsymbol{x}_{mx}^T,\boldsymbol{x}_{mv}^T,\boldsymbol{x}_{mD}^T,\boldsymbol{x}_{minD}^T,p_s,\boldsymbol{x}_{sys}^T]^T.$$

The components are illustrated as below.

(1) $\boldsymbol{x}_{HCD}$ is equal to $[\frac{n_0(0)}{N},\frac{n_1(0)}{N},...,\frac{n_{H_M}(0)}{N},0,...,0]^T\in\mathbb{R}^{\widehat{H}}$. The i-th element represents the proportion of robots which require $i-1$ relay hop counts at least to get LSI at the initial time. $\boldsymbol{x}_{HCD}$ represents the discrete probability distribution of hop count, called the hop count distribution (HCD) here. The parameter $\widehat{H}$ is a constant and represents the maximum relay hop count that an ad hoc network can tolerate, which can prevent the network performance degradation due to excessive relaying (Wenxing et al., 2017; Toh et al., 2002). The terms $\boldsymbol{x}_{HCD}$, p_b and N determine the number of robots that can get LSI in set $\mathcal{V}_h$, i.e., a_h for $h=\{0,1,2,...,H_M\}$. Since $[a_0,a_1,...,a_h]$ has direct impact on flocking fragmentation according to Eq. (10), we choose $\boldsymbol{x}_{HCD}$, p_b, and N as input features.

(2) $\boldsymbol{x}_{mx}$ equals $[\frac{1}{n_0}\sum\limits_{i\in\mathcal{V}_0}\boldsymbol{q}_i(0)^T,\frac{1}{n_1}\sum\limits_{i\in\mathcal{V}_1}\boldsymbol{q}_i(0)^T,...,\frac{1}{n_{H_M}}\sum\limits_{i\in\mathcal{V}_{H_M}}\boldsymbol{q}_i(0)^T,0,0,...,0]^T\in\mathbb{R}^{2\widehat{H}}$, the i-th element of which denotes the average position of robots in the set $\mathcal{V}_{i-1}$. $\boldsymbol{x}_{mv}$ is equal to $[\frac{1}{n_0}\sum\limits_{i\in\mathcal{V}_0}\boldsymbol{p}_i(0)^T,\frac{1}{n_1}\sum\limits_{i\in\mathcal{V}_1}\boldsymbol{p}_i(0)^T,...,\frac{1}{n_{H_M}}\sum\limits_{i\in\mathcal{V}_{H_M}}\boldsymbol{p}_i(0)^T,0,...,0]\in\mathbb{R}^{2\widehat{H}}$, the i-th element of which denotes the average speed of robots in the set $\mathcal{V}_{i-1}$. These two features are selected due to the idea that the state of robots in layer $\mathcal{V}_h$ can be equivalently replaced by the motion state of the corresponding centroid M_h, whose effectiveness has been verified in the previous section.

(3) $\boldsymbol{x}_{mD}$ equals $[\frac{1}{n_0}\sum\limits_{i\in\mathcal{V}_0}dg_i(0),\frac{1}{n_1}\sum\limits_{i\in\mathcal{V}_1}dg_i(0),...,\frac{1}{n_{H_M}}\sum\limits_{i\in\mathcal{V}_{H_M}}dg_i(0),0,0,...,0]^T\in\mathbb{R}^{\widehat{H}}$, where $dg_i(0)$ represents the number of neighbors of robot i at $t=0$, called the degree of robot i in graph theory. The element $\frac{1}{n_h}\sum\limits_{i\in\mathcal{V}_h}dg_i(0)$ represents the average degree of robots in the set $\mathcal{V}_h$. Components $\boldsymbol{x}_{mD}$ and p_s are closely related to the interaction intensity between robot layers. Specifically, larger degrees or p_s indicate stronger connection between layers, which is beneficial to preventing fragmentation between layers referring to FPM.

(4) $\boldsymbol{x}_{minD}$ equals $[\min\limits_{i\in\mathcal{V}_0}(dg_i(0)),\min\limits_{i\in\mathcal{V}_1}(dg_i(0)),...,\min\limits_{i\in\mathcal{V}_{H_M}}(dg_i(0)),0,...,0]^T\in\mathbb{R}^{\widehat{H}}$, where the term $\min\limits_{i\in\mathcal{V}_h}(dg_i(0))$ represents the minimum degree of robots in $\mathcal{V}_h$ at $t=0$. The reason to choose $\boldsymbol{x}_{minD}$ as input feature is that the robot with the minimum degree in a layer is more likely to lose connections with swarms, which may easily result in the flocking fragmentation of robots.

(5) $\boldsymbol{x}_{sys}$ is equal to $[d_e,\overline{v},\delta_v,d_0,d_r,v_r]^T$. Parameter d_e represents the feature of the expected SIG, and d_0 represents the feature of the initial SIG. The greater the similarity of the initial SIG to the expected SIG, the greater the possibilities of stability being achieved rather than fragmentation. The velocity discrepancy as well as the leader's state also have direct impacts on the system's stability so that parameters $\overline{v}$, δ_v, d_r, and v_r are considered as the input features.

Based on the above introductions of all components, the vector $\boldsymbol{x}$ can be generalized to different sizes of swarms, having a dimension of $7\hat{H}+9$.

5.3 Neural Network Architecture

The BP neural network has been widely applied in function approximation due to its strong ability to imitate nonlinear relations between input and output. Therefore, we choose the BP neural network to formulate the function relation between the high-dimensional input $\boldsymbol{x}$ and the output p_f. The end-to-end network architecture is shown in Fig. 5, from which, it can be found that the BP neural network is made up of the input, hidden, and output layers. Let b denote the number of hidden layers, and the number of neurons in the $i-$th hidden layer is o_i. The activation function in the $i-$th hidden layer is $g_i(\cdot)$. Let $W_I\in\mathbb{R}^{(7\hat{H}+9)*o_1}$ denote the weight matrix between the input layer and the first hidden layer, and the corresponding *bias* vector is θ_I. Let $W_O\in\mathbb{R}^{o_b*1}$ denote the weight matrix between the last hidden layer and the output layer, and the corresponding *bias* vector is θ_O. Let $W_i\in\mathbb{R}^{o_i*o_{(i+1)}}$ denote the

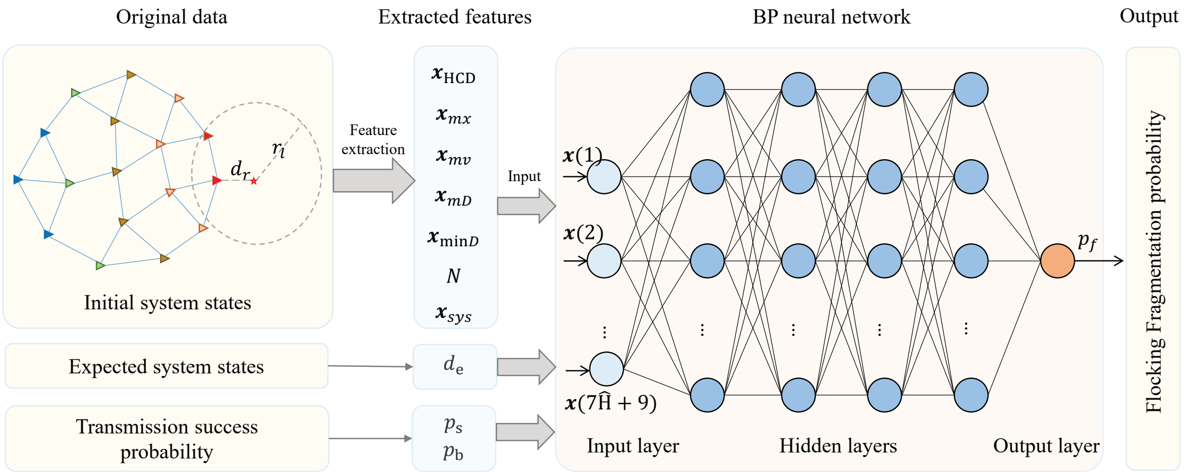

Fig. 5 The neural network architecture for fragmentation problem formulation.

weight matrix between the $i-$th hidden layer and the $(i+1)-$th hidden layer, and the corresponding *bias* vector is θ_i. The input and output vectors of the $i-$th hidden layer are denoted as H_i^I and H_i^O, respectively. The activation function of the output layer is $g_o(\cdot)$.

Based on the above symbols, the input and output of the first hidden layer are $H_1^I = {W_I}^T\boldsymbol{x} + \theta_I$ and $H_1^O = g_1({W_I}^T\boldsymbol{x} + \theta_I)$, respectively. The input and output of the $i-$th hidden layer are $H_i^I = {W_{i-1}}^T H_{i-1}^O + \theta_{i-1}$ and $H_i^O = g_i({W_{i-1}}^T H_{i-1}^O + \theta_{i-1})$, respectively. Therefore, the final output is $p_f = g_O({W_O}^T \cdot H_b^O + \theta_O)$, and its complete expression is

$$p_f = g_O({W_O}^T g_b(\ldots g_2({W_1}^T g_1({W_I}^T \boldsymbol{x} + \theta_I) + \theta_1)\ldots) + \theta_O).$$

Let $\widehat{p}_f$ denote the expected output. Then, the network's mean square error function is

$$E_r = \frac{1}{P}\sum_{i=1}^{P}(p_f^i - \widehat{p}_f^i)^2, \tag{14}$$

where P denotes the number of the input examples.

5.4 Training Method and Sample Generation of BP Network

The activation functions of neurons in hidden and output layers are all selected as *tansig* function, i.e, $tansig(x) = \frac{1+e^{-2x}}{1-e^{-2x}}$. Then, the formulation of the function $f(\cdot)$ requires to further determine the neural network parameters (i.e., W_I, θ_i,...,W_O and θ_O). We use the back propagation based on the scaled conjugate gradient (SCG) method (Møller, 1993; Andrei, 2007) to train the network, and the mean square error function is employed as the loss function. The training and validation samples are generated by simulation for the BP neural network. The generation process of each $(\boldsymbol{x}, p_f)$ sample is demonstrated in Table 3. Through changing parameter configurations, a large amount of training and validation samples can be obtained.

6 Simulation Results and Analysis

In this section, the BP-based flocking fragmentation probability formulation is evaluated by simulation with MATLAB. First, the training data set and architecture configuration are introduced for the BP neural network. Then, the error analysis is conducted by comparing the BP-based and simulation-based FFP results. Finally, the impacts of network parameters on FFP are explored and demonstrated by numerical analysis.

6.1 Dataset Construction and Architecture Configuration

We first introduce the parameter configuration for constructing the training and validation data sets. Since the present research focuses on the impact of network topology parameters (especially the transmission performance of LSI) on the flocking phenomenon, there are some parameters that are excluded from the scope of this study and they are accordingly set as constants throughout all simulations: $d_e = 20$, $\bar{v} = 4$, $r_l = 50$, $p_s = 0.8$, $r = 1.2d_e$,

Table 3 Illustration of sample generation

```
Training Sample Generation Process
For BP Neural Network
Input: $N$, $d_e$, $d_0$, $\bar{v}$, $\delta_v$, $r_l$, $d_r$, $d_v$, $p_s$, $p_b$, $k_1$, $k_2$,
       $c_1$, $c_2$, $T$, $l_n$.
Output: feature vector $\boldsymbol{x}$, the FFP $p_f$.
Initialize: $k = 0$ and $f_n = 0$.
Process:
  1. generate scenario based on parameters $N$,
     $d_0$, $\bar{v}$, $\delta_v$, $d_r$ and $d_v$. The initial system sta
     -tes are generated based on the description
     provided in sub-section 5.1, and are saved
     in the vector $\boldsymbol{q}(0)$, $\boldsymbol{p}(0)$, $\boldsymbol{q}_r(0)$, and $\boldsymbol{p}_r(0)$.
  2. extract input feature vector $\boldsymbol{x}$ from the
     initial system states and parameters;
  3. for i=1:1:$l_n$
       while $k \le 500$ :
         (1) each robot acquires the NSI at the
             successful probability $p_s$;
         (2) each robot acquires the LSI at the
             success probability $p_b$;
         (3) each robot calculates the acceleration
             instruction based on Eq. (4);
         (4) robots and leader move forward for a
             duration $T$ and then update its state
             based on Eq. (1) and (2);
         (5) identify if fragmentation appears
         if (fragmentation appears):
             $f_n = f_n + 1$;
         endif
         (6) identify if flocking has reached stabi-
         -lity based on Definition 7;
         if (reaches stability):
             break;
         endif
         (7) $k = k + 1$;
       end
     end
  4. $p_f = f_n / l_n$.
return: a set of training sample ($\boldsymbol{x}$, $p_f$).
```

and $\widehat{H} = 16$. Besides, the controller related parameters also remain fixed: $k_1 = 0.4$, $k_2 = 0.2$, $c_1 = 0.04$, $c_2 = 0.4$ and $T = 0.3$. It is noted that all the samples employed to construct the data set must achieve flocking stability under the conditions that $p_s = 1$ and $p_b = 1$. This is to ensure that the fragmentation phenomenon solely results from the configurations of network topology parameters but is unrelated to the controller parameters.

To construct the data set, the range of values for different parameters is defined as: (1) $N \in [30:5:90] \bigcup [33:5:93]$, which embodies different sizes of robot swarms; (2) $\delta_v \in \{2, 4, 6\}$, which corresponds to different degrees of velocity discrepancy; (3) $d_0 \in [0.5:0.1:1.1] * d_e$, which represents different degrees of gap between the initial SIG and the expected SIG; (4) $d_r \in [0.4r_l : 0.1r_l : 0.8r_l]$; (5) $v_r \in \{2, 4, 6, 8\}$, with the result that the smaller the values of d_r or v_r is, the easier it would be for robots to catch up with the leader; (6) $p_b \in [0.3 : 0.05 : 1]$. There are a total of $26 * 3 * 7 * 5 * 4 * 15 = 163800$ value situations for the set $[N, \delta_v, d_0, d_r, v_r, p_b]$. For each value situation, we obtain $n_s = 4$ samples according to Table 3.

Thus, there are a total of 655200 samples in the data set. The whole data set is randomly divided into an 80% training set and a 20% validation set. The learning rate is set to 0.01 during training. The training process will halt when the loss $E_r < 0.0001$ or remains unchanged for 15 consecutive iterations. The obtained BP neural network has four hidden layers in total, and the numbers of neurons in each layer variously amount to 85, 70, 45 and 25.

6.2 Error Analysis for the BP-based FFP

To validate the accuracy and generalization capability of the FFP formulation, the error analysis is conducted by comparing the FFP formulation with its simulational counterparts. Specifically, for a set of parameters $[N, \delta_v, d_0, d_r, v_r, p_b]$, we randomly generate the initial flocking scenario according to Table 3. Then, for the same initial states of robots and leaders, both the simulational and theoretical function curves of p_f with respect to p_b are obtained and compared. Furthermore, the mean square error E_c is employed to quantify the error between formulation-based and simulation-based p_b-p_f curves by calculating $E_c = \frac{1}{n_p}\sum_{i=1}^{n_p}(p_f^i - \widehat{p}_f^i)^2$, where n_p is the number of points in the p_b-p_f curves. The symbols p_f^i and $\widehat{p}_f^i$ correspond to the i-th pair of formulation-based and simulation-based FFP values in the p_b-p_f curves.

Fig. 6 depicts the model errors under different values of N, where $N \in \{31, 37, 42, 48, 54, 60, 66, 72, 78, 84, 90\}$ and δ_v, d_0, d_r, and v_r are set to 2, $0.9d_e$, $0.5r_l$, and 2, respectively. It is noted that a large proportion of the parameter values employed for testing are not present in the training data set, and the intention behind is the need to test the accuracy and generalization capability of BP-based formulation. According to Fig. 6(a), the formulation-based p_b-p_f

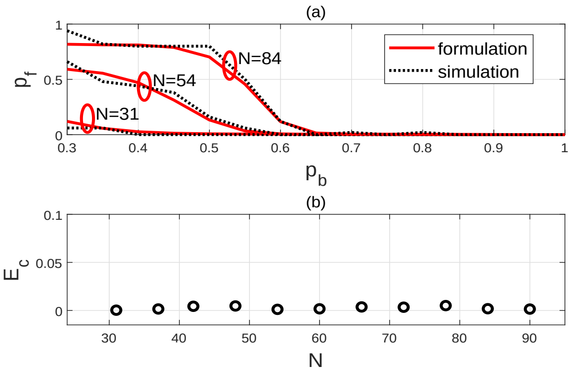

Fig. 6 (a) Comparisons between simulational and theoretical p_b-p_f curves when N is set to different values. (b) The mean square error E_c between simulational and theoretical p_f-p_b curves with N set to different values.

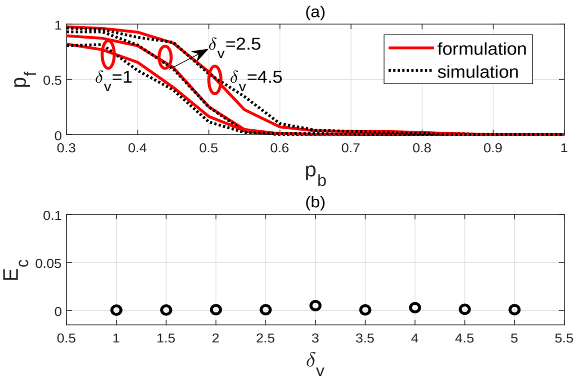

Fig. 7 (a) Comparisons between simulational and theoretical p_b-p_f curves when δ_v is set to different values. (b) The mean square error E_c between simulational and theoretical p_f-p_b curves with δ_v set to different values.

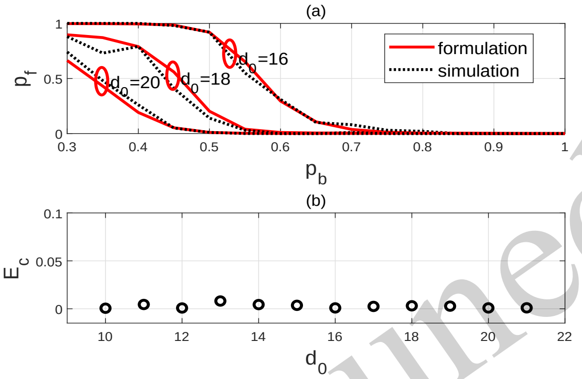

Fig. 8 (a) Comparisons between simulational and theoretical p_b-p_f curves when d_0 is set to different values. (b) The mean square error E_c between simulational and theoretical p_f-p_b curves when d_0 is set to different values.

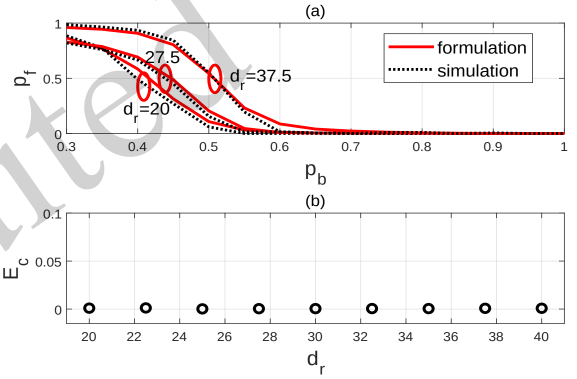

Fig. 9 (a) Comparisons between simulational and theoretical p_b-p_f curves when d_r is set to different values. (b) The mean square error E_c between simulational and theoretical p_f-p_b curves when d_r is set to different values.

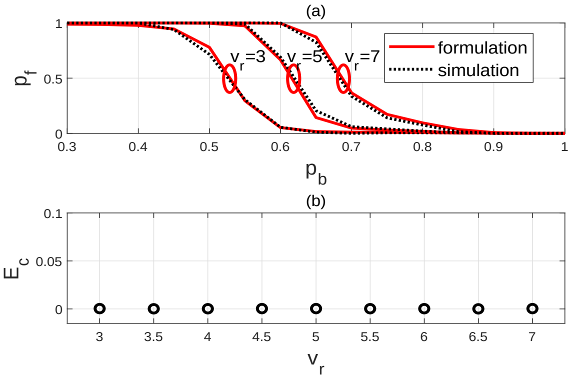

Fig. 10 (a) Comparisons between simulational and theoretical p_b-p_f curves when v_r is set to different values. (b) The error E_c between simulational and theoretical p_f-p_b curves under different v_r.

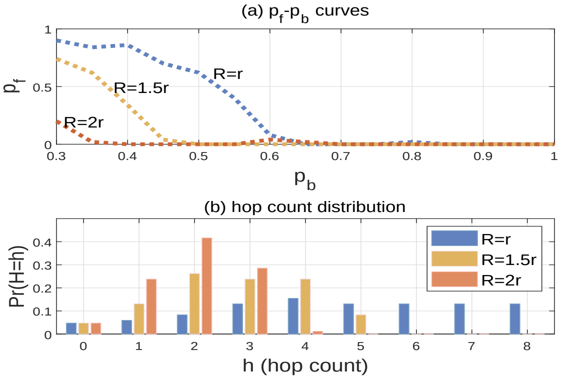

Fig. 11 (a) Illustration diagraphs of p_b-p_f curves when LSI's transmission range R is set to r, $1.5r$, and $2r$. (b) Illustration diagraphs of hop count distribution when R is set to r, $1.5r$, and $2r$.

Table 4 Simulation parameters of Figs. 7-10

	Fig. 7	Fig. 8	Fig. 9	Fig. 10
N	50	50	50	50
d_0	$0.9d_e$	$[0.5{:}0.05{:}1.05]{*}d_e$	$0.9d_e$	$0.9d_e$
d_r	$0.5r_l$	$0.5r_l$	$[0.4{:}0.05{:}0.8]{*}r_l$	$0.5r_l$
v_r	2	2	2	3:0.5:7
δ_v	1:0.05:5	2	2	2

curves match well with the simulation-based ones when $N = 31$, $N = 54$ and $N = 84$, which demonstrates the accuracy of our formulation. The results in Fig. 6(b) further validate the formulation's accuracy, since the mean square error E_c for various N are all ≤ 0.05, a value that is close to 0. The coincidence of curve pairs and the tiny values of E_c both reflect the accuracy and generalization capability of the BP-based formulation for different swarm sizes.

Similar findings can also be found in Figs. 7-10. The corresponding simulation parameters for Figs. 7-10 are listed in Table 4, where a considerable proportion of parameter configurations are not present in the training data set, either. Figs. 7-10 demonstrate the errors of FFP formulation with respect to the parameters δ_v, d_0, d_r, and v_r. In each figure, both the p_b-p_f curves and the mean square error of curve pairs are demonstrated, and it is clear that the BP-based FFP formulation shows good coincidence with simulation results. Therefore, it can be concluded that the FFP formulation has excellent accuracy and generalization capabilities, thus demonstrating its applicability in relation to robot swarms of different sizes with changeable and random initial states.

6.3 Effect Analysis of key feaures on FFP

In this subsection, we analyze the impact of the key input features on FFP based on Figs. 6(a)-10(a). From Figs. 6(a)-10(a), it is observed that regardless of the parameter configurations, p_f decreases with the increase in p_b. This observation is in line with expectations, since the increase in p_b is beneficial for the global announcement of LSI. Besides, it is found that for a given set of $[N, \delta_v, d_0, d_r, v_r]$, when p_b exceeds a critical threshold, denoted by $p_b^* \in (0, 1)$, the value of p_f will remain equal to 0, which means that the fragmentation would not occur. Therefore, it is not necessary to realize the absolute transmission success of LSI over each hop. Nevertheless, when designing the communication network for a multi-robot flocking system, appropriate technologies, such as signal modulation (Jiang and Wu, 2008; Jiang et al., 2005) and channel coding (Wang et al., 2016; Jiang et al., 2023) technologies, should be selected and improved to satisfy the critical performance requirements (e.g., p_b^*) of links.

We next analyze the effect of parameters N, δ_v, d_0, d_r, and v_r successively. According to Fig. 6(a), the size of the swarm has a great impact on FFP. When robot swarms have a small size, such as $N = 31$, the FFP is almost 0 over the interval $p_b \in (0.3, 1)$. However, when N raises to 54 or 84, the fragmentation probability increases sharply over the interval $p_b \in [0.3, 0.5]$. The corresponding thresholds p_b^* also grows to about 0.6 and 0.65, respectively. These results mean that robot swarms with larger sizes have a higher probability of demonstrating the fragmentation phenomenon. This is attributable to the fact that in a larger swarm, more robots have to get LSI at higher relay hop counts with less success possibility. Intended to solve this problem, one solution is to improve the STP of communication links through physical-layer technologies, and another is to optimize the network topology features such as the hop count distribution by adjusting the transmission range of LSI.

To visually demonstrate the impact of hop count features on FFP, we conduct a comparison experiment by changing the HCD of multi-robot communication networks. Specifically, the transmission range of LSI is assumed to be adjustable, and is denoted by R, while the transmission range of NSI is still set to r. We consider three scenarios where R is set to r, $2r$, and $3r$, respectively, and the other parameters N, δ_v, d_0, d_r, and v_r are set to 84, 2, $0.9d_e$, $0.5r_l$, and 2, respectively. By displaying the p_s-p_f curves under different values of R, we can better understand the impact of hop count on FFP. As shown in Fig. 11(a), with the increase in R, the fragmentation probability falls. Particularly, when R is set to r, $1.5r$, and $2r$, respectively, the corresponding thresholds p_b^* are about 0.62, 0.45, and 0.35, respectively. The reason for the observation is demonstrated in Fig. 11(b). As can be seen, the increase in R has greatly changed the HCD of robots. When $R = r$, the farthest robots require 8 hop-count relays to obtain LSI, and most robots acquire LSI at large hop counts. When R is equal to $1.5r$ and $2r$, the maximum relay hop count becomes 5 and 4, respectively, and more robots acquire LSI at a smaller hop count. An overall small hop count is beneficial for the global informing of LSI, and thus the FFP decreases.

The impact of δ_v is displayed in Fig. 7(a). The variable δ_v represents the random velocity discrepancy of all robots at initial time. From Fig. 7(a), the increase in speed difference among robots will raise the fragmentation probability, and higher critical values of p_b^* are required to prevent fragmentation. This phenomenon arises for two reasons. Firstly, a wider speed difference means that the distance between neighbors changes sharply, so that they are more likely to lose connection. Secondly, wider velocity discrepancies require more time to achieve velocity consensus, before which, there always exists the possibility for robots with different speeds to get separated from each other. Therefore, it can be inferred that there is a tendency for a higher risk of fragmentation to prevail in pursuance of more random and chaotic initial states of MRS. To deal with it, more reliable and robust communication networks and links are desired to ensure the information interaction among robot.

Fig. 8(a) demonstrates the effect of d_0 on FFP. d_0 represents the initial distance between neighbors, which is a key feature of the initial SIG. Contrastingly with the situations prevailing corresponding to the d_0 values of 18 and 16, the FFP is lowest when d_0 is equal to the expected distance d_e, i.e., $d_0 = 20$. As the value of d_0 gets further away from d_e, the FFP increases, which demonstrates that a value closer to d_e is beneficial for preventing fragmentation. The reason is that when d_0 approaches the expected distance, the initial SIG becomes closer to the expected SIG, which helps the robot swarm to reach stability faster, so that the corresponding FFP decreases. As a result, the required threshold p_b^* for preventing fragmentation also decreases. Therefore, when the initial state of robot swarms is far from the stable state, the communication performance should be more strictly guaranteed to ensure interaction.

The impact of d_r and v_r are displayed in Figs. 9(a) and 10(a), respectively. We are able to infer from Figs. 9(a) and 10(a) that the increase in either d_r or v_r raises the fragmentation probability. Larger d_r means that the robot swarms are further away from the leader, while higher v_r means that the average speed of the swarm is far less than that of the leader. Both these conditions are unfavorable for robot swarms to track the leader and cost the whole swarm more time to reach flocking stability. Therefore, the fragmentation probability is increased. Intended for the increase of d_r and v_r, communication links and networks with higher reliability are required to ensure the transmission of robot information to prevent fragmentation.

7 Conclusions and Future Researches

The transmission of LSI and NSI in communication networks is crucial to guaranteeing multi-robot flocking stability. Especially, when the global delivery of LSI cannot be guaranteed, the flocking is likely to demonstrate fragmentation phenomena and cannot reach stability. Since LSI's transmission performance depends on the topology structure and parameter configuration of multi-robot communication networks, we analyze the impact of network topology parameters (including hop count features and STP) on flocking fragmentation by formulating their relationship. More specifically, we first present a CCE protocol to describe the interaction and control process of multi-robot flocking, where an extended discrete-time O-S model is employed to achieve flocking control in multi-hop and lossy ad hoc networks. Then, we put forward the FPM model for specific flocking scenarios, and furthermore develop the FFP formulation for generalized flocking scenarios. Simulation validates the effectiveness of our proposed method. The developed FFP formulation has excellent accuracy and generalization capability, which is applicable to robot swarms of different sizes with changeable and random initial states. The results indicate that a higher STP and an overall small hop count can prevent flocking fragmentation and ensure final stability. Future researches may leverage more advanced neural network technologies or propose methods to guarantee flocking stability under multi-hop and lossy ad hoc networks.

Contributors

Silan LI designed the research. Silan LI processed the data. Silan LI drafted the manuscript. Shengyu ZHANG and Tao JIANG helped organize the manuscript. All the authors revised and finalized the paper.

Compliance with ethics guidelines

Silan LI, Shengyu ZHANG, and Tao JIANG declare

that they have no conflict of interest.

References

Alam MM, Arafat MY, Moh S, et al., 2022. Topology control algorithms in multi-unmanned aerial vehicle networks: An extensive survey. *Journal of Network and Computer Applications*, 207:103495.
https://doi.org/10.1016/j.jnca.2022.103495

Andrei N, 2007. Scaled conjugate gradient algorithms for unconstrained optimization. *Computational Optimization and Applications*, 38:401-416.
https://doi.org/10.1007/s10589-007-9055-7

Antonelli G, Arrichiello F, Chiaverini S, 2010. Flocking for multi-robot systems via the null-space-based behavioral control. *Swarm Intelligence*, 4:37-56.
https://doi.org/10.1007/s11721-009-0036-6

Arafat MY, Moh S, 2022. A Q-learning-based topology-aware routing protocol for flying ad hoc networks. *IEEE Internet of Things Journal*, 9(3):1985-2000.
https://doi.org/10.1109/JIOT.2021.3089759

Arafat MY, Poudel S, Moh S, 2021. Medium access control protocols for flying ad hoc networks: A review. *IEEE Sensors Journal*, 21(4):4097-4121.
https://doi.org/10.1109/JSEN.2020.3034600

Chung FR, 1997. Spectral graph theory. American Mathematical Soc.

Gundlach JH, Schlamminger S, Spitzer CD, et al., 2007. Laboratory test of newton's second law for small accelerations. *Phys Rev Lett*, 98:150801.
https://doi.org/10.1103/PhysRevLett.98.150801

Hafeez KA, Zhao L, Mark JW, et al., 2013. Distributed multichannel and mobility-aware cluster-based MAC protocol for vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62(8):3886-3902.
https://doi.org/10.1109/TVT.2013.2258361

He G, Li H, 2017. Distributed control for multirobot systems with collision-free motion coordination. 2017 10th International Symposium on Computational Intelligence and Design (ISCID), 2:72-76.
https://doi.org/10.1109/ISCID.2017.87

Huang X, Liu A, Zhou H, et al., 2021. FMAC: A self-adaptive MAC protocol for flocking of flying ad hoc network. *IEEE Internet of Things Journal*, 8(1):610-625.
https://doi.org/10.1109/JIOT.2020.3007071

Ibuki T, Wilson S, Yamauchi J, et al., 2020. Optimization-based distributed flocking control for multiple rigid bodies. *IEEE Robotics and Automation Letters*, 5(2):1891-1898.
https://doi.org/10.1109/LRA.2020.2969950

Jiang T, Wu Y, 2008. An overview: Peak-to-average power ratio reduction techniques for OFDM signals. *IEEE Transactions on Broadcasting*, 54(2):257-268.
https://doi.org/10.1109/TBC.2008.915770

Jiang T, Yang Y, Song YH, 2005. Exponential companding technique for papr reduction in OFDM systems. *IEEE Transactions on Broadcasting*, 51(2):244-248.
https://doi.org/10.1109/TBC.2005.847626

Jiang T, Liu Y, Xiao L, et al., 2023. PCC polar codes for future wireless communications: Potential applications and design guidelines. *IEEE Wireless Communications*, :1-7.
https://doi.org/10.1109/MWC.017.2200586

Li S, He S, Zhang Y, et al., 2022a. Edge intelligence enabled heterogeneous multi-robot networks: Hybrid framework, communication issues, and potential solutions. *IEEE Network*, 36(6):108-115.
https://doi.org/10.1109/MNET.106.2100465

Li S, Hu X, Jiang T, et al., 2022b. Hop count distribution for minimum hop-count routing in finite ad hoc networks. *IEEE Transactions on Wireless Communications*, 21(7):5317-5332.
https://doi.org/10.1109/TWC.2021.3139350

Liu W, Gao Z, 2020. A distributed flocking control strategy for UAV groups. *Computer Communications*, 153:95-101.
https://doi.org/10.1016/j.comcom.2020.01.076

Mohamed RE, Hunjet R, Elsayed S, et al., 2023. Connectivity-aware particle swarm optimisation for swarm shepherding. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):661-683.
https://doi.org/10.1109/TETCI.2022.3195178

Møller MF, 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525-533.
https://doi.org/10.1016/S0893-6080(05)80056-5

Olfati-Saber R, 2006. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401-420.
https://doi.org/10.1109/TAC.2005.864190

Olfati-Saber R, Iftekhar L, 2012. Flocking for networks of mobile robots with nonlinear dynamics. 9th International Conference on Informatics in Control, Automation and Robotics, 4:353-359.
https://doi.org/10.5220/0004048403530359

Patterson S, Bamieh B, Abbadi AE, 2010. Convergence rates of distributed average consensus with stochastic link failures. *IEEE Transactions on Automatic Control*, 55(4):880-892.
https://doi.org/10.1109/TAC.2010.2041998

Reynolds CW, 1987. Flocks, herds and schools: A distributed behavioral model. Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, New York, USA, p.25-34.
https://doi.org/10.1145/37401.37406

Sastry S, 1999. Lyapunov Stability Theory. In: Nonlinear Systems: Analysis, Stability, and Control. Springer New York, New York, NY.

Seoung Kyou L, 2022. Distributed deformable configuration control for multi-robot systems with low-cost platforms. *Swarm Intelligence*, 3(16):169-209.
https://doi.org/10.1007/s11721-022-00211-2

Shao J, Zheng WX, Shi L, et al., 2021. Leader-follower flocking for discrete-time Cucker-Smale models with lossy links and general weight functions. *IEEE Transactions on Automatic Control*, 66(10):4945-4951.
https://doi.org/10.1109/TAC.2020.3046695

Shen G, Lei L, Li Z, et al., 2022. Deep reinforcement learning for flocking motion of multi-UAV systems: Learn from a digital twin. *IEEE Internet of Things Journal*, 9(13):11141-11153.
https://doi.org/10.1109/JIOT.2021.3127873

Su H, Wang X, Lin Z, 2007. Flocking of multi-agents with a virtual leader part I: With a minority of informed agents. 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, p.2937-2942.
https://doi.org/doi.org/10.1109/CDC.2007.4434066

Sun JQ, Xiong FR, Schütze O, et al., 2018. Cell mapping methods. Springer.

Toh C, Delwar M, Allen D, 2002. Evaluating the communication performance of an ad hoc wireless network. *IEEE Transactions on Wireless Communications*, 1(3):402-414.
https://doi.org/10.1109/TWC.2002.800539

Wang F, Chen Y, 2020. Fast convergent flocking control of multi-agent systems with switching communication topology. 2020 American Control Conference (ACC), Denver, CO, USA, p.695-700.
https://doi.org/10.23919/ACC45564.2020.9147719

Wang T, Qu D, Jiang T, 2016. Parity-check-concatenated polar codes. *IEEE Communications Letters*, 20(12):2342-2345.
https://doi.org/10.1109/LCOMM.2016.2607169

Wenxing L, Muqing W, Min Z, et al., 2017. Hop count limitation analysis in wireless multi-hop networks. *International Journal of Distributed Sensor Networks*, 13(1):1550-1329.
https://doi.org/10.1177/1550147716683606

XIn Y, Zhiyong F, Wenjun X, et al., 2018. Secure connectivity analysis in unmanned aerial vehicle networks. *Frontiers of Information Technology & Electronic Engineering*, 3(19):409-422.
https://doi.org/10.1631/FITEE.1700032

Yazdani S, Haeri M, Su H, 2019. A sampled-data leader-follower algorithm for flocking of multi-agent systems. *IET Control Theory and Applications*, 13(5):609-619.
https://doi.org/10.1049/iet-cta.2018.5533

Yazdani S, Haeri M, Su H, 2020. Multi-rate sampled-data algorithm for leader-follower flocking. *IET Control Theory and Applications*, 14(4):3038-3046.
https://doi.org/10.1049/iet-cta.2020.0637

Yazdani S, Haeri M, 2017. A sampled-data algorithm for flocking of multi-agent systems. 2017 Artificial Intelligence and Robotics, Qazvin, Iran, p.147-152.
https://doi.org/10.1109/RIOS.2017.7956458

Yazdani S, Su H, 2022. A fully distributed protocol for flocking of time-varying linear systems with dynamic leader and external disturbance. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(2):1234-1242.
https://doi.org/10.1109/TSMC.2020.3019469

Yunpeng W, Kunxian Z, Daxin T, et al., 2020. Cooperative channel assignment for vanets based on multiagent reinforcement learning. *Frontiers of Information Technology & Electronic Engineering*, 7(21):1047-1058.
https://doi.org/10.1631/FITEE.1900308

Zheng J, Dong JG, Xie L, 2017. Synchronization of the delayed vicsek model. *IEEE Transactions on Automatic Control*, 62(11):5866-5872.
https://doi.org/10.1109/TAC.2017.2648505