IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 4, 2019, accepted April 27, 2019, date of publication May 9, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916004

Swarm Intelligence-Inspired Autonomous Flocking Control in UAV Networks

FEI DAI[1], MING CHEN[2], XIANGLIN WEI[3], AND HUIBIN WANG[1]

[1]College of Command Control Engineering, Army Engineering University of PLA, Nanjing 210007, China
[2]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[3]Nanjing Telecommunication Technology Research Institute, Nanjing 210007, China

Corresponding author: Ming Chen (mingchennj@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772271 and Grant 61379149, and in part by the Research Foundation of Education Bureau of Anhui Province, China, under Grant KJ2017B15.

ABSTRACT The collaboration of multiple unmanned aerial vehicles (UAVs) has stimulated the emergence of a novel wireless network paradigm named UAV network. UAV network, compared with uncoordinated UAV systems could provide wider coverage, better monitoring, and understanding of the interested area, and smarter decision-making. However, realizing the full potential of UAV network in dynamic environments poses great challenges in topology/flocking control, energy conservation, and quality of service guarantee. In this backdrop, this paper proposes a swarm intelligence-inspired autonomous flocking control scheme for UAV networks. First, based on the concept of intelligent emergence of swarm agents, a swarm intelligence-inspired multi-layer flocking control scheme is built for the flocking control problem. Second, an integrated sensing and communication method is put forward to regulate how a UAV can calculate its distances to its neighbors and its deflection angle. Finally, a series of experiments are conducted on our simulator developed on OMNeT++ and the flocking prototype to evaluate the effectiveness of the proposed scheme. The simulation and experimental results have shown that the proposed scheme could realize efficient flocking control with low energy consumption and satisfied the quality of service.

INDEX TERMS UAV network, swarm intelligent, flocking control, energy.

I. INTRODUCTION

Compared with single-Unmanned Aerial Vehicle (UAV) system, which usually has limited energy supply, restricted computation capacity, and poor survivability, a multiple UAV system is expected to realize much wider coverage, better monitoring and understanding of interested area, smarter decision-making, and thus to better support diverse applications [1]. These benefits come from the collaboration among UAV Networks (UAVNs) [2]. However, many challenges arise in the collaboration process among UAVs in dynamic environments, like topology control, energy conservation, and Quality of Service (QoS) guarantee [3], [4]. To be specific, the communication and coordination among a group of UAVS may be affected or disturbed by many factors, like rain, wind, and electromagnetic environments [1]. Besides, supporting bandwidth-intensive or latency-sensitive mobile applications, like terrain monitoring and vehicle tracking, need a UAV maintain high-speed wireless links with its neighbors or collaborators. This means that UAV nodes can only cooperate in the air over a broadband wireless network, instead of relying on a narrow-band wireless communication system via a ground control system [2]. For Flying ad hoc networks (FANETs) [5] using broadband Ad Hoc network technology, the system cannot predict and effectively control the node to move regularly in a complex environment due to the inherent randomness of both the transmission environment and the mobility of UAVs. Thus, in this circumstance, efficiently maintaining the topology of the UAVN based on pre-defined hierarchical structure while ensuring the communication quality (i.e. QoS) of the wireless links in the network with acceptable energy consumption is very difficult.

To bridge this gap, the UAVs in a UAVN need to jointly coordinate their movement to maintain mutual high-bandwidth wireless links and the coordination process fulfills four requirements. First, the distance between two neighbor UAVs should be close enough to reduce the attenuation of wireless transmission and thus to maintain a certain

The associate editor coordinating the review of this manuscript and approving it for publication was Valentina E. Balas.

QoS level. Second, the distance between any two UAVs should be not too close to avoid potential collision. Third, each UAV should autonomously adjust its own flying status since no central control facility exists. This is due to the fact that the use of centralized optimization control methods, such as genetic algorithms, particle swarm optimization, etc., will bring extremely high computational complexity and large traffic, which will make real-time control of UAV network nodes infeasible [6]. Last but not least, the flight attitude control process should not consume much energy.

To meet these requirements, swarm intelligence [7] is a feasible way if each autonomous UAV acts as an intelligent agent. Each agent makes decisions about its own next move based on its distances to its neighbors, QoS restricts, and current transmission environment. As a whole, all UAVs can maintain a robust topology with a high degree of self-organization and self-adaptation. Ideally, a UAVN could have a flocking behavior like a swarm of birds or fishes [8]. When a UAVN has the characteristics of flocking, each UAV in it satisfies the following features: it can keep connections with other UAVs without being an isolated node; it will not collide with other UAVs; it can share common motion trends with its group; in the case of a certain signal-to-noise ratio, it autonomously perceives, calculates, and controls its distances to certain adjacent nodes, and make necessary move with the least energy consumption.

In this paper, to achieve flocking in UAVNs, a swarm intelligence-inspired multi-layer flocking control (SIMFC) scheme is put forward. Firstly, the flocking control problem in UAVN is formulated, and our flocking control scheme based on multi-layer network model is presented. Secondly, an integrated sensing and communication method is put forward to regulate how a UAV can calculate its distances to its neighbors and its deflection angle. Finally, a simulator is developed on OMNeT++ [9], and a flocking prototype is constructed using off-the-shelf DJI M100 drones to evaluate the effectiveness of the proposed scheme. Our contributions in this paper is threefold. First, the flocking control problem in UAVNs is formally stated as an iterative optimization problem, in which the QoS requirement and energy consumption factors are included. Second, a swarm intelligence-inspired multi-layer flocking control scheme is developed and introduced in detail. Third, a simulator and a prototype for UAVN are developed based on OMNeT++ and DJI drones respectively, and a series of experiments are conducted to validate the effectiveness of our proposal.

The organization of this paper is as follows. Section II summarizes related work. Section III formulates the problem and illustrates our topology control scheme. Section IV details the integrated method of communication and sensing. Section V presents the simulation and experiments and analyzes the results. Section VI concludes the paper in brief.

II. RELATED WORK

A lot of research on the issues of QoS for UAV network, such as [10], [11], focus on the transmission control of UAV networks, hoping to improve the performance of FANET [2] by ameliorating the MAC layer [12], network layer [13] or hybrid routing [14] and hierarchical routing [15] in traditional wireless networks, and the use of power control, smart antenna and other technologies. However, it still cannot achieve the expected guarantee and obtain of the QoS for UAV network.

Reynolds proposed the flocking model in his pioneering work in 1986, which has three heuristic rules that led to creation of the first computer animation of flocking [8]. Reynolds' three flocking rules include: 1) Cohesion: attempt to stay close to nearby nodes; 2) Separation: avoid collisions with nearby nodes; 3) Alignment: attempt to match velocity with nearby nodes. This has motivated and guided many flocking theoretical models. Recently, there has been a surge of interest in consensus problems due to [16], [17], and so on. Although the objectives that these theories aim at are different, the flocking, with its simplistic and effective framework, has been widely adopted as the coordination scheme in multi-agent systems [18], [19].

There has been a lot of research on the flocking control of multi UAVs. From the control structure perspective for multi UAVs, the existing flocking control approaches can be classified into the centralized method, where a single controller is used to control the whole team based on the information from the whole team [20] and the distributed/decentralized method, where each team member generates its own control based on local information from its neighbors [21]. The centralized flocking control can be a good strategy for a small team of UAVs. When considering a team with a large number of UAVs, the need for greater computational capacity and a large communication bandwidth would mandate a distributed/decentralized control. From the control mechanism perspective, flocking control approaches can be classified into consensus-based approaches [22], [23], artificial potential function-based approaches [24], [25], and leader–follower approaches [26], [27]. Consensus-based approaches convert the flocking control problem into the consensus (or stability) problem of relative positions and velocities of multi-agents. They achieve formation stability based on graph theory and consensus. However, inter-vehicle collisions are not considered. Artificial potential function-based approaches apply the negative gradient of a mixture of attractive and repulsive potential functions as control inputs to satisfy the convergence and non-collision properties, respectively. The main drawback of this type of approaches is the appearance of equilibrate, where the composite vector field vanishes and the UAVs can get trapped at undesired equilibrium points. Leader–follower approaches simplify the formation problem into individual tracking problems. The main disadvantage is that the leader is a single point of failure for the formation. Moreover, [28], [29] use visual aids to solve the flocking control problem without GPS or GPS failure, which can reduce the communication load caused by explicit communication. But it also brings higher computational complexity and cost.

Particularly, we proposed a distributed flocking model (DFM) in [30] to make the UAV network have the flocking characteristic by means of a leader-follower. However, the DFM method only used the Wi-Fi communication method based on ad hoc. In which the scalability was not strong, and the collision problem between nodes had not been studied in depth. Considering the limited computing, communication and energy capabilities of the UAV and the problem of real-time control, this paper hopes to reduce the complexity of the flocking control from the structure, and integrates the method of artificial potential function and leader-follower approaches to implement the flocking control method of UAV network, and provide performance and low cost technology for ensuring network QoS.

III. PROBLEM STATEMENT AND SCHEME DESIGN

This section states the problem first, and then presents our flocking control scheme.

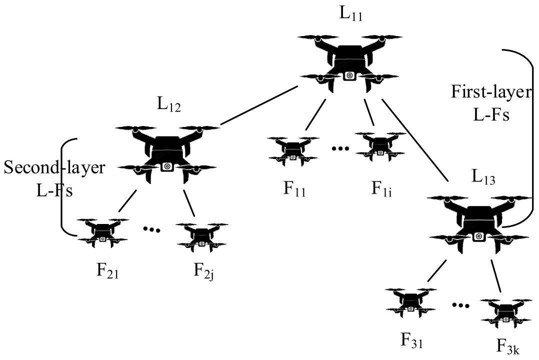

FIGURE 1. **A two-layer instance of the hierarchical UAVN architecture.**

A. PROBLEM STATEMENT

1) SYSTEM MODEL

In the application scenarios where multiple UAVs are involved and work collaboratively, a consistent hierarchical structure is usually adopted and maintained by a flocking control scheme to better regulate the communications among UAVs. A two-layer instance of this scenario is illustrated in Figure 1, where UAVs in the first layer is labeled with L_{1x}, and those in the lower layer is named F_{xy}, where x, y refers to the indices of the UAVs in the first and second layers respectively. To be specific, the ith UAV in the first layer is labeled with L_{1i}, and the jth UAV in the second layer associated with the ith UAV in the first layer is called F_{ij} in Figure 1. The UAVs in the first layer are also called leaders, and followers refer to UAVs in the second layer. In Figure 1, a leader UAV is usually connected to a few followers. For instance, L_{11}, L_{12}, L_{13} have i, j, and k followers respectively in Figure 1. The leader-follower relationship is established by wireless linking among two UAVs. Besides, wireless links are also maintained among the first-layer UAVs to ensure the connectivity of the UAVN. In this layered architecture, a flocking control scheme aims to maintain the hierarchical structure of the network in a way that some specific QoS-level could be maintained on the wireless links among UAVs without the least energy consumption.

To state the flocking control problem, we make the following assumptions about the multi-UAV systems.

a) Each UAV could calculate its own wireless communication states and maneuver, and acquire its location information $\langle lon, lat, height \rangle$ and flying status $\langle speed, angle, omega \rangle$, in real time through sensors.

b) A multi-layer hierarchical structure could be constructed through extending the layers in the architecture shown in Figure 1. In other words, a follower Figure 1 could be at the same time a leader for a few UAVs in the third layer, and so forth.

c) IEEE 802.11 standards (or Wi-Fi for short) is the Medium Access Control (MAC) protocol adopted in the UAVNs.

d) The UAVs at the same layer fly at the same altitude, and use the communication frequency, while UAVs at different layers fly at the different heights. The interference between UAVs at the first and the second layers could be neglected due to their heights differences.

e) One of the UAVs in the first layer acts as the root of the UAVN shown in Figure 1, and it decides the moving direction and trajectory of the UAVN.

f) To ensure the QoS of a wireless link, the distance between the two UAVs of this link must be less than a distance threshold R_{max}. In the case where the distance between two UAVs is smaller than R_{max}, the QoS of their mutual wireless link can be guaranteed.

g) The distance between any two UAVs should not smaller than a safe distance threshold R_{min} to avoid collisions.

For the simplicity of expression, the leaders and followers will be referred to be L-node and F-node for short in the following analysis. The network constructed by all the L-nodes is called the backbone of the UAVN, and each sub-network consists of a leader and its followers will be called a cluster for short.

2) TRACKING MODEL

Based on the idea of swarm intelligence, especially Reynolds' Boid model [8], we build a model (shown in Figure 2) to study the flocking control problem of the UAVNs. In Figure 2, r_1 (r_2) is the minimum (maximum) safe distance $R_{\min}$ ($R_{\max}$) between any two UAVs.

According to the theory of swarm intelligence, in order to guarantee the QoS of wireless links, any pair of UAVs should satisfy the three principles of Boid model (cohesion, separation, and alignment). An L-node of the first layer can determine its own flight trajectory according to the task and environmental status, and its associated F-nodes needs to adjust their own distance and flight direction according to the L-node's state information. As shown in Figure 2, for an L-node L_{11}, its associated F-nodes should be in the annular shadow region centered at L_{11}. The F-nodes need to adjust

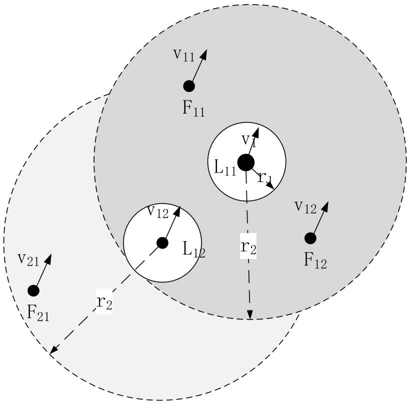

FIGURE 2. Flocking constraints between L-nodes and F-nodes in the multi-layer model.

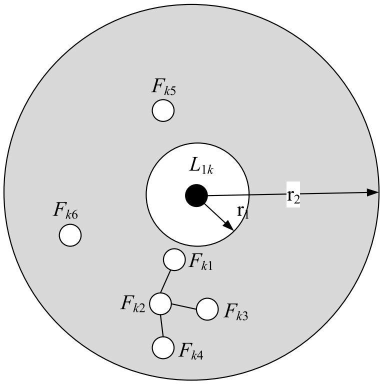

FIGURE 3. A collision avoidance scenario for multiple F-nodes.

their flying status once L_{11} changes its position, and the distance between node L and node F should satisfy:

$$r_1 \leq d_{fl} \leq r_2 \tag{1}$$

where d_{fl} is the distance between the F node and the L node.

If all UAVs follow this rule in the whole moving trajectory, the UAVN can maintain its inertia and a relatively stable state [31]. But in practice, L-nodes and F-nodes could not always keep at the desired position due to many environmental factors. For example, when the speed or course of an L-node or an F-node is changed due to external factors (such as the change of wind direction and/or speed), if the other one fails to adjust in time, it will result in the changes of overall system state. A collision or network disruption may happen in this circumstance Thus, it is necessary for all F-nodes to monitor the status change in time and make autonomous adjustments when the state of the system changes. Therefore, the QoS-guaranteed flocking control model must adopt the following control strategy:

a) During the movement, when the flying status of an L-node changes, its associated F-nodes should adjust their status accordingly.

b) When F-node falls to the region of $d_{fl} < r_1$ or $d_{fl} > r_2$, F should self-perceive and adjust to enter the appropriate range centered on L as soon as possible.

3) COLLISION AVOIDANCE

The regulation model in the above section could avoid the collision between a pair of L-node and F-node. Furthermore, the collision risk between F-nodes should also be taken into consideration. For an F-node F_{ki} following L_{1k}, its neighbor nodes set is assumed to be N_i, and all the F-nodes in N_i follow the same L-node L_{1k}. In order to avoid collision, node F_{ki} and any neighbor node F_{kj} should satisfy:

$$\forall j \in N_i, \quad d_{ij} \geq r_1 \tag{2}$$

where d_{ij} is the distance between node F_{ki} and node F_{kj}. If (2) is not satisfied, F_{ki} needs to make an autonomous adjustment to restore (2). As shown in Figure 3, there is no danger of colliding with other nodes for F_{k5} and F_{k6} as they have no neighboring nodes. As for node F_{k2}, it is adjacent to nodes F_{k1}, F_{k3}, and F_{k4}. When F_{k2} is too close to F_{k1}, F_{k3}, and F_{k4}, there is a danger of collision. So F_{k2} should move autonomously to ensure that the collision will not happen. In other words, each F-node needs to autonomously calculate the combined influence of neighboring nodes to avoid collision.

B. MULTI-LAYER FLOCKING CONTROL SCHEME

To realize the collision avoidance discussed in the above section, this section introduces the multi-layer flocking control scheme. For a F-node F_{ki}, assume its current location is q_i, and the current location of its associated L-node L_{1k} is q_L. The set of threat nodes for F_{ki}, called T_i, is defined as:

$$T_i = \{j | d_{ij} < r_1, j \in N_i\} \tag{3}$$

where $d_{ij} = \left\| q_i - q_j \right\|$ indicates the distance between node F_{ki} and F_{kj}.

The field where other nodes have a collision threat to this node is defined as the potential field, and the size of the threat is calculated by the potential function. For a node F_{ki}, the potential function that defines the influence of other nodes on this node has two parts: ϕ_{iL}^{follow} and ϕ_{ij}^{threat} $(j \in T_i)$. Here, ϕ_{iL}^{follow} shows the potential function between node F_{ki} and its associated L-node L_{1k}. When the potential function ϕ_{iL}^{follow} is 0, (1) is satisfied between F_{ki} and L_{1k}. ϕ_{ij}^{threat} is the potential function between F_{ki} and the jth threat node in T_i. The definition of the potential functions are defined as follows.

a) Potential function ϕ_{iL}^{follow}

F_{ki} and its leader L_{1k} should satisfy (1). We define:

$$\phi_{iL}^{follow} = \begin{cases} (r_1 - d_{iL})^2 & d_{iL} < r_1 \\ 0 & r_1 \leq d_{iL} \leq r_2 \\ (d_{iL} - r_2)^2 & d_{iL} > r_2 \end{cases} \tag{4}$$

Obviously, in (4), when the distance between F_{ki} and L_{1k} satisfies (1), the potential function is 0. In other words, F_{ki} does not need to adjust its relative position with respect to L_{1k}. When the distance between F_{ki} and L_{1k} is less than r_1 or greater than r_2, the potential function is greater than 0. At this time, node F_{ki} needs to adjust its position to satisfy (1). The larger the potential function is, the greater the adjustment will be.

b) Potential function ϕ_{ij}^{threat}

For the node F_{ki} and any threat neighbor node F_{kj} ($F_{kj} \in T_i$), according to the definition of T_i, it can be known that $d_{ij} < r_1$. According to (2), ϕ_{ij}^{threat} can be defined as:

$$\phi_{ij}^{threat} = (r_1 - d_{iL})^2, \quad F_{kj} \in T_i \tag{5}$$

Similar to (4), in (5), the closer the distance between F_{ki} and F_{kj} is, the larger the potential function is. At this time, F_{ki} needs to adjust its flying status with a large amplitude.

Based on (4) and (5), the goal of F_{ki} is to reduce the potential function to zero as quickly as possible. When the values of the potential functions of all nodes in the network are 0, the QoS of UAVN can be guaranteed. The velocity which can reduce the potential function ϕ is $v_i' = -k\frac{\partial \phi}{\partial q_i}$, the velocity vector for all potential functions can be added as:

$$v_i'' = -k_1 \frac{\partial \phi_{iL}^{follow}}{\partial q_i} - k_2 \sum\nolimits_{j \in T_i} \frac{\partial \phi_{ij}^{threat}}{\partial q_i} \tag{6}$$

where k_1 and k_2 are the speed reconciliation parameters, and the parameter size is related to the maximum flying speed. Since F_{ki} must satisfy the alignment principle with its associated L-node, F_{ki} needs to follow L_{1k}'s trajectory. The target speed of F_{ki} is:

$$v_i''' = \alpha v_i'' + v_L \tag{7}$$

where v_L is the speed of L_{1k}, and α is the speed limit parameter. Assuming the maximum flying speed is $v_{\max}$, then:

$$\alpha = \arg\max(\alpha v_i'' + v_L) \tag{8}$$

The use of the threat set T_i makes any node only needs to acquire the information of its neighboring nodes whose distances are within its threat range. This can effectively reduce the information sharing frequency between the node and its adjacent nodes. For the information of an L-node, if its location and status information change rapidly, the frequency of information sharing must be increased too. When the position and state change slowly or periodically, such as the case that the root node cruises at a fixed speed, that is, the linear velocity and angular velocity ω remains unchanged. Then the L-node's speed v_L in (7) can be expressed as:

$$v_L = (|v_{L0}| \cos(\theta_0 + \omega t), |v_{L0}| \sin(\theta_0 + \omega t)) \tag{9}$$

where $|v_{L0}|$ is the fixed cruising speed of the root node, θ_0 is the initial yaw angle when the node enters the flight phase with angular velocity ω, and t is the time of the flight phase from the node entering the angular velocity ω.

The process for UAVs to calculate their flight parameters and adjust their trajectories is described in Algorithm 1.

In Algorithm 1, the location and status information of $F_{ki}'s$ L-node L_{1k} and its neighbor nodes is taken as the inputs of algorithm. F_{ki} calculates the potential function between itself and L_{1k}, using the latest information (line 1-2) and the potential function with neighboring nodes (line 3-6). The speed of the L-node is calculated based on L_{1k}'s status (line 7-10). Finally, F_{ki} calculates its adjustment velocity v''' and the algorithm is performed periodically on all follower nodes in the UAVN.

Algorithm 1 Flocking Control Algorithm for F_{ki}

```
Input: location of L-node $q_L$, location of neighbors $q_j$
Output: $v''$, the target speed of $F_{ki}$
1.  $d_{iL} = \|q_i - q_L\|$
2.  calculate $\phi_{iL}^{follow}$
3.  For each $F_{kj} \in T_i$
4.    $d_{ij} = \|q_i - q_j\|$
5.    calculate $\phi_{ij}^{threat}$
6.  End
7.  If $v_L$ received
8.    update $v_L$
9.  Else
10.   calculate $v_L$ using $\omega$
11. End
12. calculate $v''$ according to (6)
13. $\alpha = \arg\max(\alpha v_i'' + v_L)$
14. $v_i''' = \alpha v_i'' + v_L$
15. adjust its flight using $v'''$
```

Since the UAV is battery powered, energy consumption issues must be considered. For a UAV with mass m, the total flying energy that it consumes to traverse distance d is [32]:

$$E = \frac{(\hat{v} + v)(mg + f_d)d}{v\eta} \tag{10}$$

where $\hat{v}$ is the induced velocity required for a given thrust T, v is the average ground speed of the UAV, g is the gravitational constant and f_d is the drag force that depends on the air speed, density of air, and the drag coefficient, η is the power efficiency of the UAV. Here we can assume that $\hat{v}$ and v are approximately constant, then the energy consumed by the UAV is proportional to the distance it travels. In order to save the energy consumption of the flocking control scheme, each UAV needs to select the moving action towards its anticipated position with the least energy need.

IV. INTEGRATED METHOD OF COMMUNICATION AND SENSING

In the flocking algorithm presented in the above section, the flying status of a follower's leader node and neighbors are assumed to be known. To obtain these status information, in this section, we need to figure out how each UAV can acquire the information and calculate its distances to other UAVs and the deflection between nodes.

In order to calculate the potential function, a UAV needs to perceive the existence of other UAVs and their relative orientation. We can obtain the spatial coordinates of L-nodes and F-nodes through the positioning system, such as the Global

Positioning System (GPS) or BeiDou [33], and calculate their distance. However, the use of high-precision satellite positioning equipment may raise the price of the UAVs, and more importantly, these devices are not available on many off-the-shelf drones. On the other hand, adopting low-precision satellite positioning equipment with large positioning errors cannot ensure the precision of distance calculation. To our best knowledge, almost no cheap sensors that can directly achieve the research goals of this paper exist for now. High-end sensors, such as expensive radars or lasers have strong directionality to achieve the desired effect. But they will further promote the price of the UAVN. Therefore, we rely on Wi-Fi signals to realize distance calculation.

This is due to the fact that each Wi-Fi device can be treated as a sensor that transmits Wi-Fi signals evenly to all directions in a 3D space. In other words, Wi-Fi has both the communication and positioning functions. In close-range situations, Wi-Fi can provide accurate signal indication for node collision avoidance.

A. WI-FI BASED DISTANCE ESTIMATION

In the case of a long distance (such as when the distance is greater than r_1), the distance and orientation information provided by the low-precision satellite positioning device can be used. When the distance between two UAVs is relatively close (for example, when the distance is less than r_2), the Wi-Fi signals can be utilized to provide more accurate distance estimation.

Considering that the wireless signal of Wi-Fi can basically cover the sphere space centered on the transmitter, there have been many researches and applications in Wi-Fi based distance estimation [34]. The Received Signal Strength (RSS) received by the receiver can reflect the distance between the transmitter and the receiver. The distance d can be expressed with respect to the received RSS value as [34]:

$$d = 10^{\frac{A - rss + \omega}{10q}} \tag{11}$$

where A is apparent transmission power, q is a parameter describing attenuation properties of the environment, and w is a zero-mean Gaussian random variable used for modeling the shadow fading. An F-node can infer its distance to the L-node by its RSS. Therefore, we use the Wi-Fi signal as a full-range distance sensor. As long as the relationship between the distance d between two UAVs and the received RSS strength of Wi-Fi can be measured in a specific environment, the value of d can be obtained from the obtained RSS value. Assuming that $rss_{\max}$ is the signal strength value corresponding to the distance r_1, then the threat nodes set T_i in section III can be expressed as:

$$T_i' = \{j | rss_{ij} \geq rss_{\max}, j \in N_i\} \tag{12}$$

In addition, due to the energy-saving technologies used by advanced Wi-Fi technology and the difference in each Wi-Fi chip, there is a big difference in the measured RSS values. We can use exponential weighted moving average (EWMA) (shown in (13)) to smooth the acquired RSS values.

$$ERSS_i = (1 - \alpha) \times ERSS_{i-1} + \alpha \times SRSS_i \tag{13}$$

where $ERSS_i$ is the ith estimated value, $SRSS_i$ is the ith measured value, and α is the weight coefficient. Therefore, the measured numerical fluctuations will not have a remarkable impact on the distance prediction.

B. OPTIMIZATION CONTROL METHODS

Each node actively sends its own location and other information through the communication protocol. To be specific, an L-node actively broadcasts its own location and other status information to its followers; each UAV actively sends its location information to the UAVs in its threat set. To control the broadcasting overhead, a few time periods are defined here. The calculation period T_0 is defined as the time interval during which a UAV performs the autonomous decision calculation; the threat information broadcast period T_1 is the time interval during which a UAV sends its position to the node in its threat matrix; the L-node broadcast period T_2 refers to the time interval during which the L-node broadcasts its location and status information to its followers.

Section III has analyzed the cruise mode of the L-node and its impact on the information sharing frequency. In order to further save the communication resources and energy consumption, we divide the broadcast period T_2 of an L-node into a short period T_{21} and a long period T_{22} ($T_{22} > T_{21}$). When the flying state (direction, speed, and etc.) of the L-node changes rapidly, the information is sent to its followers with a short period T_{21}. In contrast, when the flying state of the L-node changes slowly or periodically, the long period T_{22} is used to broadcasts the information. In particular, in the process of broadcasting in long period T_{22}, when the L-node changes from one state to another (such as a change in the angular velocity ω), information should be immediately sent to its associated F-nodes, and the cycle timer should be reset. The related process is described in the communication control algorithm (Algorithm 2) of the flocking control scheme.

In Algorithm 2, an L-node only needs to broadcast its information periodically (line 5-6) and the selection of the periodic time is based on the flying status of itself (line 7-11). For an F-node, it needs to send location messages to its threat nodes (line 15-20).

V. EXPERIMENTS AND RESULTS ANALYSIS

In order to evaluate the proposed scheme, we designed and implemented a simulator on OMNeT++ and a prototype system based on off-the-shelf DJI DMV100 drones, and carried out a series of experiments on them. To analyze the experimental data, we defined the following evaluation metrics.

Adjust Distance (AD): the AD at the ith moment between an L-F nodes pair of the UAV network is defined as:

$$AD_i = d_{t_{i+1}} - d_{t_i} \tag{14}$$

Algorithm 2 Communication Control Algorithm

```
Leader:   // when a UAV acts as an L-node
  initialize $\omega = \omega_0$
  initialize periodic time $T = T_{22}$
  While(True)
    broadcast status message to followers
    timeout($T$) //time out of the periodic time $T$
    If $\omega$ changed
      update $T = T_{21}$
    Else
      update $T = T_{22}$
    End
  End
Follower:   // when node acts as a F-node
  initialize periodic time $T = T_1$
  While(True)
    For each $j \in T_i'$
      send location message to node $j$
    End
    timeout($T$)
  End
```

where $d_{t_{i+1}}$ is the distance between the F-node and the L-node at time t_{i+1}, d_{t_i} is the distance between them at time t_i, and the unit of AD is meter (*m* for short).

Total Adjust Distance (TAD), Average Adjust Distance (AAD) and Max Adjust Distance (MAD): the TAD, AAD, and MAD between an L-F nodes pair of the UAV network are defined as:

$$TAD = \sum_{i=1}^{N_t} AD_i \tag{15}$$

$$AAD = \frac{TAD}{N_t \Delta t} \tag{16}$$

$$MAD = \max_{i \in [1, N_t]} \{AD_i\} \tag{17}$$

where Δt is the time interval between t_{i+1} and t_i, N_t is the total number of time intervals, and the unit of TAD, AAD and MAD is *m*, *m/s* and *m*, respectively. Obviously, the more frequent the distance between the two UAVs changes, the larger the value of AAD and TAD will be, and the more the network topology changes.

Network Total Adjust Distance (NTAD) and Network Average Adjust Distance (NAAD): the NTAD and NAAD of the UAV network are defined as follows:

$$NTAD = \sum_{j=1}^{N-1} TAD_j \tag{18}$$

$$NAAD = \frac{\sum_{j=1}^{N-1} TAD_j}{N_t \Delta t} \tag{19}$$

where the unit of NTAD and NAAD is m and m/s. Obviously, the more frequent the distance between the two UAVs changes, the larger the value of NTAD and NAAD will be.

Average Control Meassages (ACM): the ACM of the UAV network indicates the average control meassages used for flocking control during a unit time, and the unit of ACM is *pkts/s*.

Extra Energy Consumption Ratio (EECR): the ratio of the extra energy consumption used for flocking adjustment to the normal flight energy consumption. The EECR of the *i*th UAV is defined as:

$$EECR_i = \frac{TAD_i}{D_L} \tag{20}$$

where D_L is the total flying distance of the root leader node. Here, we assume that energy consumption is only proportional to the flying distance, and the leader does not hover during the flight. Ideally, the flight distance of a UAV should be the same as the root leader node. Therefore, the ratio of the extra energy consumption used for flocking adjustment to the normal flight energy consumption can be expressed as (20).

A. SIMULATION AND RESULTS

We built a UAV network simulation test environment on OMNeT++ [9]. The physical layer of the model mainly utilizes the Radio model and the Medium model. The MAC layer of the first layer network adopts Ad Hoc mode Wi-Fi and optimized link state routing (OLSR), and the second layer network adopts Access Point (AP) mode Wi-Fi, and the flocking control function we developed was added to each node. To reduce signal interference during communication, we use 2.4 GHz Wi-Fi channel in the first layer and 5 GHz Wi-Fi channel in the second layer. The simulated UAVN consists of 9 UAVs,L_{11} is used as the root node in the first layer; L_{12} and L_{13} also acts as the leaders in the first layer, and they need to follow L_{11} and each of them has two F-nodes. During the moving process of the UAVN, wireless communication interference and random wind interference are set. The main test parameters are shown in Table 1.

TABLE 1. Parameters of simulation tests.

Parameters	Value
platform	OMNet++
technology of MAC layer	AP (5GHz), Ad hoc (2.4GHz)
routing	OLSR
number of UAVs	9
areas	3km*3km
r_1	2m
r_2	70m
initial speed	10m/s
max speed	15m/s
length of line trajectory	1000m
length of round trajectory	3000m

In order to verify the effectiveness of our algorithm in different scenarios, the root node follows two different trajectories, one is a straight line and the other is a circle. L_{11}, L_{12} and L_{13} correspond to the leader nodes in the first layer. The remaining 6 UAVs are the second layer F-nodes, F_{11} and F_{12} follow L_{11}; F_{21} and F_{22} follow L_{12}; F_{31} and F_{32} follow L_{13}. All UAVs may be subject to random interference

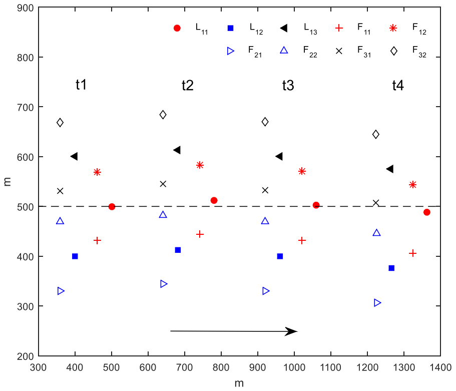

FIGURE 4. The network topology changes when the root node (i.e. L_{11}) flies along a straight trajectory.

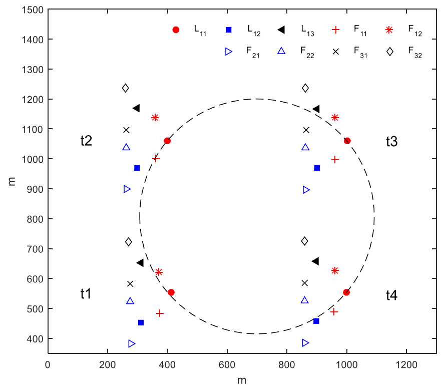

FIGURE 5. The network topology changes when the root node (i.e. L_{11}) flies along a round trajectory.

during flight. Figure 4 and Figure 5 show four snapshots of the entire network at four different times during the flying process along straight and round trajectory, respectively. It can be found from Figure 4 and Figure 5 that during the whole network flight, the actual flight trajectory and network topology of the UAV node are deviated by random interference, but they can basically fly along a predetermined trajectory and maintain the topology stability.

Table 2 shows the values of different metrics between different L-Fs pairs and the entire network under linear and round trajectory. It can be found from Table 2 that the values of the ADD index between different L-F nodes are less than 0.025 m/s, that is, the relative distance that each UAV needs to be adjusted per second during the flight is less than 0.025m. Compared with the maximum communication range between UAVs ($r_2 = 70$m), through our proposed control scheme, the UAVN can maintain a stable flight, thus ensuring the QoS of the UAV network. In terms of energy consumption, the EECR of all UAVs is below 3%, so the impact of flocking control on UAV energy consumption is within an acceptable range.

The short period T_{21} and the long period T_{22} proposed in Section IV, will affect the effectiveness of the algorithm and the performance of the network in different situations. To validate this inference, we run the PingAll module in OMNeT++ to take a test, in which each node periodically sends ping packets to all other nodes and records the number of response packets. The Number of Lost Packets (NLP) indicator is used to count the number of ping packets lost during the test. The test results are shown in Table 3.

Table 3 shows the changes in NTAD, NAAD, and NLP values during flight under different time periods T_{21} and T_{22}. It can be found from Table 3 that as time periods T_{21} and T_{22} continue to increase, the NTAD and NAAD values also gradually increase, which indicates the distance that the UAV needs to be adjusted is also increasing. Because when the time period T_{21} and T_{22} increase, the frequency of information sharing between UAVs also increases, which makes the adjustment of the relative position of the UAV network lagging behind. And that may make the adjustment cost of maintaining the flocking characteristics of the UAV network also increase, and even the network be interrupted. Different from the trend of NTAD and NAAD, when time period increases the NLP value decreases first and then increases. This is because when the time period is small, the information sharing frequency will be high. As a result, the number of control messages in the network at this time will be large, which may bring interference to other application messages that are normally transmitted. When the time period is large, although the number of control messages is reduced, the flocking characteristics of the network cannot be effectively maintained, and the QoS of the network is deteriorated.

TABLE 2. Values of different metrics under different trajectory.

		Line Trajectory				Round Trajectory			
F	L	TAD (m)	AAD (m/s)	MAD (m)	EECR	TAD (m)	AAD (m/s)	MAD (m)	EECR
F_{11}	L_{11}	20.69	0.024	4.02	2.0%	25.66	0.032	3.99	2.6%
F_{12}	L_{11}	22.03	0.025	3.81	2.2%	26.49	0.033	4.03	2.6%
L_{12}	L_{11}	21.05	0.024	3.36	2.1%	21.91	0.027	3.24	2.2%
L_{13}	L_{11}	17.42	0.020	3.11	1.7%	22.97	0.029	3.29	2.3%
F_{21}	L_{12}	8.98	0.010	0.26	0.9%	9.92	0.012	0.17	1.0%
F_{22}	L_{12}	10.06	0.012	0.20	1.0%	4.12	0.005	0.09	0.4%
F_{31}	L_{13}	5.97	0.007	0.07	0.6%	6.59	0.008	0.12	0.7%
F_{32}	L_{13}	8.10	0.009	0.15	0.8%	3.67	0.004	0.06	0.4%
NTAD (m)		114.31				121.33			
NAAD (m/s)		0.13				0.15			

TABLE 3. Values of NTAD, NAAD, and NLP in different time periods.

		Line Trajectory			Round Trajectory		
T_{21}	T_{22}	NTAD (m)	NAAD (m/s)	NLP	NTAD (m)	NAAD (m/s)	NLP
1s	1s	101.45	0.09	21	108.41	0.10	24
1s	3s	107.62	0.10	15	115.25	0.13	18
3s	3s	108.37	0.10	12	117.18	0.14	14
3s	5s	114.31	0.13	4	121.33	0.15	6
5s	5s	116.11	0.14	6	125.57	0.16	9
5s	8s	124.15	0.21	25	141.49	0.29	32

TABLE 4. Values of NTAD, and ACM under different methods.

	[24]		SIMFC	
Number of UAVs	NTAD (m)	ACM (pkts/s)	NTAD (m)	ACM (pkts/s)
5	98.79	1.5	108.24	5.5
10	107.54	4	117.96	15
15	112.66	9.5	130.18	31
20	119.25	18	141.35	52.5
25	127.28	29	149.26	76
30	136.47	41	158.47	97

Therefore, an appropriate time interval should be selected when setting the broadcast period of the L-node. In particular, both the linear trajectory test and the round trajectory test in Table 3 belong to the case where the change in motion state is small (state change due only to random interference), so the main factor affecting the NTAD, NAAD, and NLP values is the long period T_{22}.

In [24], leader-followers method and artificial potential fields are also uses to solve flocking control problems of multiple UAVs, but the size of UAVs and the overhead of control information are not considered. In order to compare it with the method in this paper, the flocking control of the UAV network is carried out by two methods under the circular trajectory. The experimental results are shown in Table 4.

Table 4 shows the changes in NTAD and ACM values during flight under different methods. In Table 4, as the number of UAVs in the network increases, the values of NTAD and ACM increase in both methods. This is because the larger the size of the UAV network, the more difficult the flocking control is, and the larger the distance that need to be adjusted and the number of control messages will be. However, in the case of the same UAV network size, the distance that needs to be adjusted and the number of control messages are relatively small under SIMFC. On the one hand, this paper uses a multi-layer flocking control scheme, which can reduce the complexity of flocking control and reduce the distance that nodes need to adjust. On the other hand, the use of integrated sensing and communication method can effectively reduce the number of control messages.

B. PROTOTYPE TEST AND RESULTS

A prototype is built using the DJI M100 drones (as shown in Figure 6), which can carry Wi-Fi modules. We have implemented the flocking control schemes on these drones, and each UAV Wi-Fi module has two wireless communication channels, i.e. 2.4 GHz and 5 GHz. Our flocking control scheme is implemented in the Raspberry Pi 3b+ embedded system [35], and the RSS values of the neighboring node are read by the Wi-Fi module. The motion status and spatial information of a UAV are read through the M100 API, and sent to the relevant UAVs through the communication protocol. Each node can use the serial port to send the calculated flight command to the UAV, to achieve self-control of its own flight, and complete the task of maintaining the QoS of the UAV network with low energy consumption.

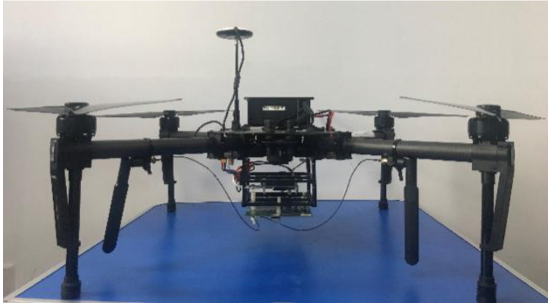

FIGURE 6. The front view of a UAV of our UAV prototype.

TABLE 5. Parameters of prototype system test.

Parameters	Value
platform	DJI M100, Raspberry Pi 3b+
technology of MAC layer	AP (5GHz), Ad hoc (2.4GHz)
routing	OLSR
number of UAVs	4
areas	100m*100m
r1	2m
r2	80m
initial speed	2m/s
max speed	5m/s
length of the trajectory	500m

The parameters of the prototype system are shown in Table 5.

In the network consisting of 4 DJI M100 UAVs, L_{11} and L_{12} are the L-nodes in the first layer, and F_{11} and F_{21} are the second layer F-nodes;F_{11} follows L_{11}, while F_{21} follows L_{12}. L_{11} also acts as the root node in this UAVN. Figure 7 shows the movement of the entire network under the situation that L_{11} is manually controlled by the remote controller, while the remaining UAVs are controlled by our flocking control scheme. It can be seen from Figure 7 that during the whole test period, the actual flight trajectory and network structure of the UAV nodes have a certain deviation due to random interference, but generally follow the trajectory of L_{11} and maintain the original network structure.

Table 6 shows the values of different metrics between different L-F nodes pairs and the entire network in our testing period. It can be found from Table 6 that the ADD values between different F-L nodes pairs are less than 0.21 m/s, that is, the relative distance that each UAV needs to be adjusted per second during the flight is less than 0.21m. Compared with the communication range between UAVs ($r_2 = 80$m),

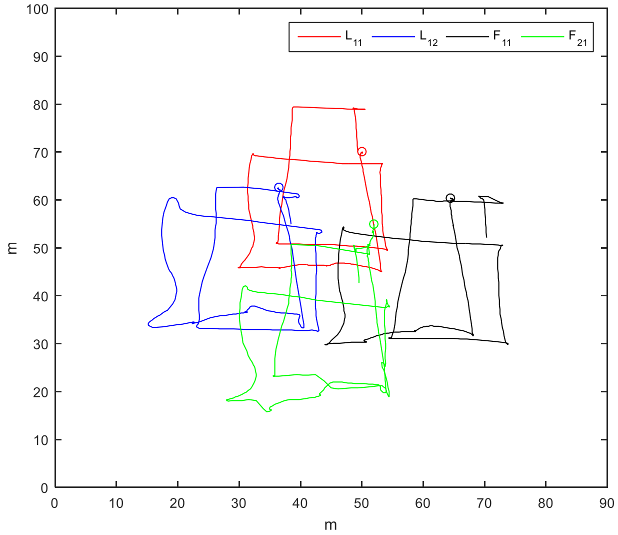

FIGURE 7. The network topology changes during the flying process of our prototype.

TABLE 6. Values of different metrics in the test.

F	L	TAD(m)	AAD(m/s)	MAD(m)	EECR
F_{11}	L_{11}	78.98	0.18	4.07	15%
L_{12}	L_{11}	87.01	0.21	3.24	17%
F_{21}	L_{12}	87.25	0.21	1.67	17%
NTAD (m)		253.24			
NAAD (m/s)		0.60			

under the control of the algorithms presented in this paper, the UAVN can maintain a stable topology, thus ensuring the QoS of the UAV network. Although the experimental results in the prototype system have been able to verify the effectiveness of the proposed method, the experimental results in the prototype system are much worse than the simulation results on OMNet++. On the one hand, it is because the various interference of flights and communications in the real environment are more complicated. On the other hand, the control of the real UAV is not as ideal as it in the simulation platform.

VI. CONCLUSION

Flocking and topology control is critical for the robust running and maintenance of multi- Unmanned Aerial Vehicle (UAV) system. This paper proposes a swarm intelligence-inspired autonomous flocking control scheme for UAV Networks (UAVNs) to help them maintain their topology during the flying process, while ensuring the Quality of Service (QoS) of them with low energy consumption. Based on the idea of swarm intelligence, a distributed multi-layer flocking control scheme (called SIMFC), which enables a follower node to autonomously follow the leader node and resolves the problems of multiple F-nodes collision avoidance, is built. In SIMFC, a multi-layer network structure is used to reduce the complexity of flocking control, and each UAV should only adjust its own flying status according to its sensed information about its leader node and neighbors UAVs to maintain a stable flying status while avoiding collision. To evaluate the effectiveness of the proposal, a simulator is built on OMNeT++ and a prototype is implemented using off-the-shelf DJI M100 drones. Results of verification test and comparison test show that the proposed SIMFC could effectively help the UAVN to maintain the anticipated topology during various scenarios.

REFERENCES

[1] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[2] E. Cruz, "A comprehensive survey in towards to future FANETs," *IEEE Latin Amer. Trans.*, vol. 16, no. 3, pp. 876–884, Mar. 2018.

[3] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[4] O. Bouachir, F. Garcia, N. Larrieu, and T. Gayraud, "Ad hoc network QoS architecture for cooperative unmanned aerial vehicles (UAVs)," in *Proc. Wireless Days*, Nov. 2013, pp. 1–4.

[5] O. K. Sahingoz, "Networking models in flying ad-hoc networks (FANETs): Concepts and challenges," *J. Intell. Robot. Syst.*, vol. 74, nos. 1–2, pp. 513–527, 2014.

[6] H. Chen and J. Fei, "UAV path planning based on particle swarm optimization with global best path competition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 6, 2018, Art. no. 1859008.

[7] E. Bonabcau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford Univ. Press, 1999.

[8] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGAPH Comput. Graph.*, vol. 21, no. 4, p. 25, 1987.

[9] T. Dietrich, S. Krug, and A. Zimmermann, "A discrete event simulation and evaluation framework for multi UAV system maintenance processes," in *Proc. Syst. Eng. Symp.*, Oct. 2017, pp. 1–6.

[10] W. P. Coutinho, M. Battarra, and J. Fliege, "The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review," *Comput. Ind. Eng.*, vol. 120, pp. 116–128, Jun. 2018.

[11] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, "Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols," in *Proc. 1st Int. Conf. Latest Trends Elect. Eng. Comput. Technol. (Intellect)*, Nov. 2017, pp. 1–9.

[12] X. Liu, Z. Wei, Z. Feng, and F. Ning, "UD-MAC: Delay tolerant multiple access control protocol for unmanned aerial vehicle networks," in *Proc. IEEE Int. Symp. Pers.*, Oct. 2018, pp. 1–6.

[13] K. Namuduri, Y. Wan, and M. Gomathisankaran, "Mobile ad hoc networks in the sky: State of the art, opportunities, and challenges," in *Proc. ACM Mobihoc Workshop Airborne Netw. Commun.*, Bangalore, India, 2013, pp. 25–28.

[14] M. A. Khan, S. M. Riaz, R. M. Asif and A. Shah, "Review of communication protocols for FANETs: (Flying ad-hoc networks)," in *Proc. Int. Conf. Eng. Emerg. Technol.*, 2015, pp. 1–9.

[15] S. Razzaq, C. Xydeas, M. E. Everett, A. Mahmood, and T. Alquthami, "Three-dimensional UAV routing with deconfliction," *IEEE Access*, vol. 6, pp. 21536–21551, 2018.

[16] J. Wei, X. Yi, H. Sandberg, and K. H. Johansson, "Nonlinear consensus protocols with applications to quantized communication and actuation," *IEEE Trans. Control Netw. Syst.*, to be published.

[17] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[18] Y. Tan and Z. Zheng, "Research advance in swarm robotics," *Defence Technol.*, vol. 9, no. 1, pp. 18–39, 2013.

[19] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 52–63, Jan. 2015.

[20] S. Keshmiri and S. Payandeh, "A centralized framework to multi-robots formation control: Theory and application," in *Collaborative Agents-Research and Development*, vol. 6066. Berlin, Germany: Springer, 2011, pp. 85–98.

[21] A. Yang, W. Naeem, G. W. Irwin, and K. Li, "Stability analysis and implementation of a decentralized formation control strategy for unmanned vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 706–720, Mar. 2014.

[22] Q. Min *et al.*, "Research on UAV cooperative formation flight based on 3D program control," *Meas. Control Technol.*, no. 3, pp. 84–87, 2017.

[23] J.-C. Trujillo, R. Munguia, E. Guerra, and A. Grau, "Cooperative monocular-based SLAM for multi-UAV systems in GPS-denied environments," *Sensors*, vol. 18, no. 5, p. 1351, 2018.

[24] J. Zhang, J. Yan, P. Zhang, and X. Kong, "Collision avoidance in fixed-wing UAV formation flight based on a consensus control algorithm," *IEEE Access*, vol. 6, pp. 43672–43682, 2018.

[25] R. Chen, N. Xu, and J. Li, "A self-organized reciprocal decision approach for sensing coverage with multi-UAV swarms," *Sensors*, vol. 18, no. 6, p. 1864, 2018.

[26] J. Ghommam, H. Mehrjerdi, and M. Saad, "Robust formation control without velocity measurement of the leader robot," *Control Eng. Pract.*, vol. 21, no. 8, pp. 1143–1156, 2013.

[27] T. Liu and Z.-P. Jiang, "Distributed formation control of nonholonomic mobile robots without global position measurements," *Automatica*, vol. 49, no. 2, pp. 592–600, Feb. 2013.

[28] Y. Tang *et al.*, "Vision-aided multi-UAV autonomous flocking in GPS-denied environment," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 616–626, Jan. 2019.

[29] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-Spatial Inf. Sci.*, vol. 21, no. 2, pp. 21–32, 2018.

[30] M. Chen, F. Dai, H. Wang, and L. Lei, "DFM: A distributed flocking model for UAV swarm networks," *IEEE Access*, vol. 6, pp. 69141–69150, 2018.

[31] M. Whitzer *et al.*, "In-flight formation control for a team of fixed-wing aerial vehicles," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, Jun. 2016, pp. 372–380.

[32] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery," *Nature Commun.*, vol. 9, no. 1, p. 409, 2018.

[33] R. Zhou *et al.*, "Elevation-dependent pseudorange variation characteristics analysis for the new-generation BeiDou satellite navigation system," *GPS Solutions*, vol. 22, no. 3, p. 60, 2018.

[34] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2nd Quart., 2017.

[35] *Raspberry Pi—Teach, Learn, and Make with Raspberry Pi*. Accessed: Jan. 30, 2019. [Online]. Available: https://www.raspberrypi.org/