Article

Speeding Up Velocity Consensus Control with Small World Communication Topology for Unmanned Aerial Vehicle Swarms

Xiang Ji, Wanpeng Zhang, Shaofei Chen, Junren Luo, Lina Lu, Weilin Yuan, Zhenzhen Hu and Jing Chen *

College of Intelligence Science and Technology, National University of Defense and Technology, Changsha 410073, China; jixiang14@nudt.edu.cn (X.J.); wpzhang@nudt.edu.cn (W.Z.); chenshaofei01@nudt.edu.cn (S.C.); luojunren17@nudt.edu.cn (J.L.); lulina16@nudt.edu.cn (L.L.); yuanweilin12@nudt.edu.cn (W.Y.); hzzmail@163.com (Z.H.)

* Correspondence: chenjing001@vip.sina.com

Abstract: This study addressed a problem of rapid velocity consensus within a swarm of unmanned aerial vehicles. Our analytical framework was based on tools using matrix theory and algebraic graph theory. We established connections between algebraic connectivity and the speed of converging on a velocity. The relationship between algebraic connectivity and communication cost was established. To deal with the trade-off among algebraic connectivity, convergence speed and communication cost, we propose a distributed small world network construction method. The small world network characteristics expedite the convergence speed toward consensus in the unmanned aerial vehicle swarm. Eventually, our method greatly sped up the consensus velocities in the unmanned aerial vehicle swarms at a lower communication cost than other methods required.

Keywords: UAV swarms; swarm intelligence; graph Laplacians; small world network; consensus convergence

Citation: Ji, X.; Zhang, W.; Chen, S.; Luo, J.; Lu, L.; Yuan, W.; Hu, Z.; Chen, J. Speeding Up Velocity Consensus Control with Small World Communication Topology for Unmanned Aerial Vehicle Swarms. *Electronics* **2021**, *10*, 2547. https://doi.org/10.3390/electronics10202547

Academic Editor: Jose Eugenio Naranjo

Received: 6 August 2021
Accepted: 12 October 2021
Published: 18 October 2021

1. Introduction

Unmanned aerial vehicle swarms are very promising for counter-terrorism surveillance and post-disaster rescue. However, in general, the communication conditions of the environment are poor and the communication capacity of unmanned aerial vehicle swarms is limited. In addition, the communication topology of the swarm changes dynamically during its motion. Both of these things make it more difficult to design a local interaction topology for the velocity consensus of swarms. In this paper, we focus on speeding up the velocity consensus in swarm decision making, which can be applied to disaster relief. For example, in a confrontation scenario, UAVs in the swarms can quickly reach a consensus on velocity under the condition of limited communication capabilities and highly dynamical changes in the topology. Through the improvement of the overall response speed of the swarms, the confrontation ability and survivability of the swarms can be greatly improved.

Consensus-based approaches [1] realize the consensus of swarm decision making based on dynamic functions. In [2], Hao proved the convergence of a connected graph of the multiagent cooperative decision problem and designed the invulnerability topology generation algorithm to guarantee the connectivity when k edges or k nodes are removed from the graph. However, the problem of rapid consensus in decision making is not solved. In [3], though Olfati-Saber introduced and verified the advantages of small world networks in consensus algorithms, how to construct a small world network was not shown, especially in the case of decentralized control.

Research on swarm synchronization [4] revealed that bird flocking in reality is not fully linked, as each bird synchronizes with the surrounding six birds to achieve swarm behavior. Therefore, we propose the method of only interacting with six neighbors, which would greatly reduce the communication cost. We took and heavily modified the Vicsek model to improve the interaction method, mainly from the perspective of interactive topology, the individual speed and constraints added to the single agent. We solved the dilemma by creating a decentralized method for constructing an approximative small world network. The small world network has high algebraic connectivity, which accelerates the convergence speed toward consensus [5] in the decision making of the UAV swarm and reduces communication costs at the same time. To the best of our knowledge, other researchers have not used this method to improve the response speed in a swarm. The contributions are listed as follows:

- We propose a swarm control architecture inspired by swarm intelligence where each UAV communicates with a certain number of its neighbors. This method greatly reduces the communication costs.
- We propose a distributed algorithm for constructing an approximative small world network to control the UAV swarm, which reduces the communication cost and speeds up the consensus on velocity.
- Among the four algorithms we proposed, the fourth algorithm combining the two interaction strategies above performs the best, which reveals the relationship between convergence speed and the communication topology. It is the small world characteristic that improves the speed of swarm consensus convergence. In addition, using neighbors with large differences in velocity can further improve the speed of convergence.

The organization of this paper is as follows. Section 2 summarizes related work. In Section 3, we formulate the problem, illustrate the topology control scheme and define the norms, illustrating the convergence speed and the consistency of the system. Finally, we propose the methods for decision making based on characteristics of swarm intelligence and small world networks. Section 4 presents the simulations and experimental evaluation. Section 5 presents the discussion. Section 6 concludes the paper briefly.

2. Related Work

In this section, we introduce research on biological swarm intelligence applied to the control of UAV swarms. Moreover, we summarize the shortcomings that cause the slow convergence speed toward consensus in swarms control. Finally, we demonstrate the potential of applying the characteristics of a small world network to improve convergence speed toward consensus in the control of swarms.

In decentralized control, the agent-based control method was inspired by the intelligence of biological swarms and has been extensively studied. Different control models have been proposed, including the Boids model, the Vicsek model, the biological self-organization model, etc. Each team member conducts autonomous control with local information from its neighbors [6–9]. The research on swarm intelligence started by mimicking biological swarms, such as Reynold's Boids, ant colony modelling, particle swarm optimization (PSO) and pheromone-based algorithms [10]. A flocking algorithm imitates the behavior of a flock of birds in flight and compels each agent to steer itself based on three simple rules: separation, alignment and cohesion. In 1986, Reynolds [11] introduced three heuristic rules: (1) flock centering—attempt to stay close to nearby flock mates; (2) collision avoidance—avoid collisions with nearby flock mates; (3) velocity matching—attempt to match velocity with nearby flock mates. The Vicsek model contains many simplifications of parts of the Boids model [11]. The work of Vicsek [12] is mainly focused on the emergence of alignment in self-driven particle systems. Collision avoidance is not considered between neighbors and there is no corresponding centripetality. Olfati-Saber et al. embodied and improved the Boids model, and proposed three algorithms which can realize swarm behavior under obstacle-free conditions in [13].

However, the current research on swarm intelligence also has corresponding shortcomings. The Kilobot robot [14] designed by the team of Radhika Nagpal verified collaborative behaviors, such as foraging and formation control. A swarm system with thousands of units can automatically form patterns, such as a "K" shape, a five-pointed star, or a wrench. However, the speed of the swarm behavior is too slow; it takes more than 12 h to form such shapes. It is infeasible to apply it to highly dynamical scenarios.

It was proven that distributed formation control of a swarm is a consensus problem in [3]. Olfati-saber introduced algebraic connectivity and spectral properties of graphs, and further performed in-depth convergence analysis for directed networks. It was announced that networks with bigger k values or small world characteristics lead to faster consensuses in [5]. The design of the weights of a network was considered by Xiao and Boyd [15] to increase the algebraic connectivity of said network and the convergence speed toward consensus algorithms, but the results are not particularly efficient. The design of the topology is a promising method to achieve relatively high algebraic connectivity, and an algorithm proposed by Olfati-Saber which is based on the idea of Watts and Strogatz in [16] performs well. The rewiring of existed links gives rise to considerably faster consensus algorithms. However, the rewiring method for constructing a small world network based on global information is not feasible for the decentralized control of UAV swarms, and that is where we have to improve in this paper.

3. Problem Formulation and Scheme Design

In this section, we introduce our main work in three parts. Firstly, we put forward the problem of velocity consensus of UAV swarms with high dynamical topology and limited communication capabilities, and establish a corresponding model. Secondly, we analyze the relationship between the communication cost and the convergence speed from the perspective of topology. Thirdly, we present our swarm control method for fast consensus within dynamical swarms.

3.1. Problem Formulation

Just like the sardines in a school of sardines, UAVs can quickly synchronize with the velocities of other UAVs in a swarm. We intend to achieve consensus among the swarms' velocities and maximize the speed of velocity convergence. The velocity decision methods in swarm control based on neighbors' positions and velocity information are inspired by biological intelligence. Technical realization involves specific parameters, such as communication distance, communication cost, degree of consistency and convergence speed. The communication distance is the maximum distance over which two UAVs can communicate to each other with a certain error rate. The communication distance of a UAV is relatively small, which means that each UAV can only communicate with a few close neighbors, and the topology of UAV swarms is established on the basis of limited communication distance. Different topologies correspond to different communication costs. The greater the number of communication links, the greater the communication burden of the swarm system.

3.1.1. Velocity Consensus Model

For convenience when explaining the problem, we describe the swarm control model on a two-dimensional plane. A UAV can communicate with certain number of neighbors which fall into the region of $d < r$ ($r > 0$), which is limited by its capability, where d is the distance between the UAVs and r is the perception radius. The schematic diagram is shown in Figure 1. The UAV swarm topology changes dynamically during the motion of the swarm, and we adopted graph topology to illustrate the swarm system.

A graph $G(V, E)$ consists of a set of vertices $V = \{1, 2, \ldots, n\}$ and a set of edges $E \subseteq V \times V$. The quantity $|V|$ is called the order of the graph G and $|E|$ represents the size of the graph. The matrix $A = [a_{ij}]$ satisfying the property $a_{ij} \neq 0 \Leftrightarrow (i, j) \in E$ is called the adjacency matrix of graph G.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \tag{1}$$

Throughout the paper, we assume $a_{ii} = 0$ for all i. The graph G is undirected when the matrix A is symmetric.

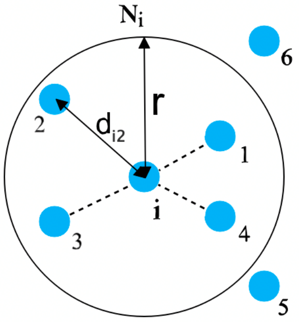


Figure 1. A UAV and its neighbors which fall into the region of d < r, where r is the perception radius and d is the distance between the UAVs.

Let $q_i \in \mathbb{R}^m$ denote the position of node i, and vector $q = \text{col}(q_1, \dots, q_n) \in Q = \mathbb{R}^{mn}$ denotes the positions of all nodes. Neighbors of node i are defined by

$$N_i = \{j \in V : \|q_j - q_i\| < r\}, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^m$. A proximity $G(q) = (V, E(q))$ net can be defined by V and the edges $E(q) = \{(i,j) \in V \times V : \|q_j - q_i\| < r, i \neq j\}$. In this paper, the perception radius of all UAVs is the same, so the dynamic proximity net $G(q)$ is undirected.

UAV $i's$ velocity is v_i, and it only communicates with certain neighbors in N_i. Each UAV's speed can be computed by multiplying coefficient v_{co} and maximum speed $V_{\max}$. v_{co} is a coefficient corresponding to difference in velocities among neighbors. The velocities v_i are determined at each time step, and the positions of each node are updated as follows:

$$x_i(k+1) = x_i(k) + v_i(k) * \Delta t, \tag{3}$$

where $\Delta t > 0$ is the time interval between two time steps. For the purpose of imitating the realistic environment of the limited communication, we suppose each UAV has randomly distributed directions θ_i. The velocity $v_i(k+1)$ of a UAV corresponds to a speed $V_i(k+1)$ and a direction $\theta_i(k+1)$—which is updated by Equation (4).

$$\theta_i(k+1) = f\{\theta_i(k), \theta_j(k)\} + \eta \cdot \Delta\theta, \quad j \in N_i, \tag{4}$$

where $f(\cdot)$ computes the direction depending on the velocities of the neighbors surrounding the focal UAV. $\Delta\theta$ denotes the noise and is randomly chosen with a uniform probability from the interval $[-\pi, \pi]$. η is the intensity of the noise. In the field of consensus algorithms, the dynamic function of discrete model can be denoted as:

$$\theta_i(k+1) = \theta_i(k) + \epsilon \sum_{j \in N_i} a_{ij}\big(\theta_j(k) - \theta_i(k)\big) + \eta \cdot \Delta\theta, \tag{5}$$

where $0 < \epsilon < 1/\Delta$, and Δ is the maximum degree of the network. Let G be a connected undirected graph. It was proven in [3] that a consensus will be asymptotically reached with the average dynamic function for all initial states. When the dynamic function is an average consensus function, a consensus will be reached in the form $\alpha = (\sum_i \theta_i(0))/n$. In our framework, the $f(\cdot)$ function gets the average direction of certain neighbors. Similarly, in the absence of external interference and under the premise that the topology is connected, the dynamic function based on direction averaging can also make multi-agents converge to a consistent direction.

Constraints such as random fluctuations and maximum turning angle are attached to individual UAVs. In the UAV swarm model, a random fluctuation is added to the direction at each time step and the intensity of the random perturbation is defined by η.

Taking into account the limited maneuverability of the UAV, the turning angle that can be achieved within a time step is limited. The maximum turning angle is referred to as φ. Every UAV in the model is initialized with a random angle θ between $[-\pi, \pi]$, and the UAVs are randomly or evenly distributed in a two-dimensional plane.

3.1.2. Velocity Consistency Measurement

The following order measurement $\Phi_\theta(k)$ is used to measure the consistency of the system.

$$\Phi_\theta(k) = \frac{1}{N}\left|\sum_{i=1}^{N} e^{j\theta_i(k)}\right|, \tag{6}$$

where N is the number of UAVs and $\theta_i(k)$ is the direction of UAV i at time step k. $\Phi_\theta(k)$ has the property of $0 \leq \Phi_\theta(k) \leq 1$. $\Phi_\theta = 1$ means the isotropy state of direction, and emergent behavior can be observed if $\Phi_\theta(k) \gg 0$. $\Phi_\theta(k)$ is based on only the directions of neighbors, so the consistency will not be affected by the variable speed.

Furthermore, the computational complexity of $\theta_i(k)$ is $O(n)$. Thus, it is suitable for our model with varying speed.

3.1.3. Communication Cost

An important aspect of performing coordinated tasks in a distributed multi-agent system is to maintain communication when the inter-agent communication cost is limited. The communication cost of an individual is the number of neighbors that a UAV refers to during velocity synchronization, and it is the same as the cost of an individual computing the motions of certain surrounding neighbors. We define the communication cost of the topology G as M. In [17], M is called the "communication complexity" of executing a task. For weighted undirected graphs, M can be denoted as a function of the adjacency metrix by

$$M = \sum_{i,j=1}^{n} \operatorname{sgn}(a_{ij}), \tag{7}$$

where $\operatorname{sgn}(\cdot)$ is the sign function. However, in our paper, the graphs are unweighted and undirected. The communication cost is equal to the number of communication links among all the UAVs.

3.2. The Trade-Off between Communication Cost and the Convergence Speed

The velocity decision method based on neighbors' position and velocity information can reach the goal of convergence in general, but the communication cost and convergence speed vary with the changes in the interactive topology. Higher communication cost guarantees that the swarm converges quickly, even in the presence of noise. With lower communication cost values, it is hard to reach high $\Phi_\theta(k)$. However, utilizing the characteristics of a small world network with relatively shorter average lengths of connected paths, the convergence speed of the system could be improved while the communication cost is relatively low.

3.2.1. Characteristics of a Small World Network

The diagonal matrix of vertex degrees of graph G is denoted by D.

$$D = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n \end{bmatrix}, \tag{8}$$

where d_i denotes the degree of the corresponding node, and $L(G) = D - A$ is called the Laplacian matrix [5] of graph G.

The Laplacian of an undirected graph is a symmetric matrix with all real eigenvalues, and the eigenvalues can be located sequentially in an ascending order as follows:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2\Delta, \tag{9}$$

where Δ is the maximum degree of a graph.

The eigenvalues of the Laplacian matrix are important characteristics of graph topology. The second smallest eigenvalue [18,19] of the Laplacian is called the algebraic connectivity of a graph, and it denotes the convergence speed toward consensus algorithms. When the topological structure is undirected, the eigenvalues of the Laplacian matrix are located at the zero point and positive on the real axis. The eigenvalue which equals zero denotes that the graph contains corresponding number of sub-connected graphs. The maximum eigenvalue describes the degree of tolerance or robustness of the graph topology to the communication time delay.

A small world network [16] with a shorter average path length has a larger clustering coefficient than a random network of the same scale. In the past, people thought that networks were divided into completely regular networks and completely random networks. The regular network has a large average path length and clustering coefficient, and the random network has a small average path length and clustering coefficient. The clustering coefficient C is a local feature quantity which illustrates the ratio of the number of edges of a node to the maximum number of edges that the node can construct. Due to the short average path and large clustering coefficient of the small world network, the decision topology with small world characteristics always has a lager convergence speed, which is mentioned below.

In [16], normalized average shortest path lengths and clustering coefficients corresponding to different rewiring probabilities p were gathered via a regular network to construct the small world network drawn in Figure 2.

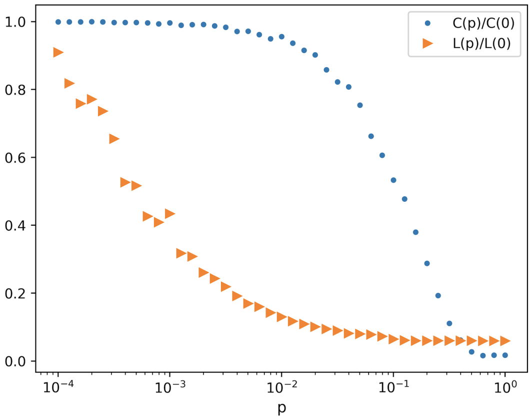


Figure 2. The normalized average shortest path lengths and clustering coefficients corresponding to different rewiring probabilities p, which were gathered via a regular network to construct the small world network.

When the communication cost M is the same among networks (small world and regular), the convergence speed of the topologies with small world characteristics is faster than that of the regular networks because the second-smallest eigenvalue of the small world network is larger. In [3], it was concluded that a dramatic increase in λ_2 entails both a high probability of random rewiring and high complexity of the network. Here in our research, the large number of nodes and the dynamical changes in the topology in the UAV swarms result in high complexity of the topology, which ensures that we can construct a small world network with limited links to achieve an increase in convergence speed.

3.2.2. Constructing Small World Network in a Decentralized Manner

In [16], a method of constructing a small world network was introduced. Each node of a regular network is connected to K nearest neighbor nodes (K is an even number). The regular network in Figure 3a has 10 nodes, and each node is connected to four adjacent nodes. Then, one should randomly reconnect each edge with probability $p(0 <= p <= 1)$. When $p = 0$, it is a regular network; and when $p = 1$, it is a completely random network (Figure 3c). By adjusting the value of p to around 0.1, a small world network with better characteristics, such as a shorter average shortest path length and lager clustering coefficient, can be obtained, as in Figure 3b.

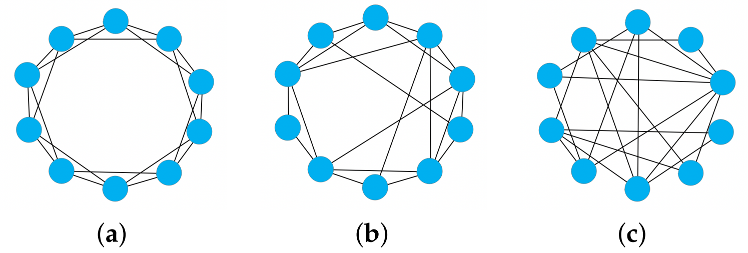


Figure 3. Three kind of networks composed of a set of vertices $V = \{1, 2, \ldots, n\}$ and edges $E \subseteq V \times V$. (**a**) Regular network, (**b**) Small world network, (**c**) Random network.

The key to constructing a small world network is to generate long connecting edges. The difficulty in practice is that individuals can only grasp the information of some neighbors, and cannot obtain the global information, so the classic small world network construction method is not applicable. We utilize the definition of second-order neighbors [3] and establish connections between second-order neighbors, so that we get the long connecting edges in the decentralized swarm.

Definition 1. *$\forall j \in N, l(i,j) = y$, the node j is called the yth-order neighbor of node i, where $l(i,j)$ is the shortest path length between node i and j.*

Definition 2. *$\forall k \in N, l(i,k) = 2$, the node k is the second-order neighbor of i, where $l(i,k)$ is the shortest path length between node i and k.*

UAVs in the swarm establish connections with second-order neighbors with a certain probability, so as to obtain small world characteristics. Details of the establishment of connections with second-order neighbors are introduced in the next subsection. Additionally, our proposed method for constructing a small world network based on second-order neighbors is feasible.

3.3. Algorithm Design

In this subsection, we introduce the basic algorithm based on the classic Vicsek model and our promising algorithms. The Vicsek model can simulate the flight behavior of birds in the biological world. It has been cited and studied in depth by many researchers, and it is a practical model for swarm intelligence research. The basic algorithm we introduce here has some improvements on the Vicsek model, such as variable speed and maximum turning angle. We use the basic algorithm which has been verified in the biological world as a reference to highlight the superiority of the new algorithm.

3.3.1. Basic Consensus Convergence Algorithm

We first came up with a basic method based on the Vicsek model which has been verified in the biological world; for example, the coordination between bird flocks probably adopts this approach. In the basic swarm interaction method in Figure 4, the UAV i will find all of its neighbors N_i in yellow with Equation (2). At each discrete time step, the direction of velocity is updated by

$$\theta_i(k+1) = \frac{1}{g}\sum_{j \in N_i} \theta_j(k), \tag{10}$$

where g is the number of neighbors of UAV i. As the maneuverability of the UAV is limited, we set the maximum turning angle as φ. When $|\theta_i(k) - \theta_i(k+1)| > \varphi$, we have to correct the $\theta_i(k+1)$ with

$$\theta_i'(k+1) = \theta_i(k) + [(\theta_i(k) - \theta_i(k+1))/|\theta_i(k) - \theta_i(k+1)|] \cdot \varphi, \tag{11}$$

which ensures that the turning angle does not exceed the maneuverability of the UAV. Then we apply the value of $\theta_i'(k+1)$ to $\theta_i(k+1)$. Considering some interference in reality, we introduce uniformly distributed interference $\Delta\theta$ and add it to the velocity direction $\theta_i(k+1)$. $\Delta\theta$, which can be calculated by $\Delta\theta = \eta \cdot$ uniform $(-\pi, \pi)$. For updating of the speed of UAV i, we calculate the consistency v_{co} of neighbors of UAV i with Equation (6). Then we update the speed of UAV i as

$$V_i(k+1) = v_{coki}^2 \cdot (V_{\max} - V_{\min}) + V_{\min}, \tag{12}$$

The UAV i travels fast when the neighbors are in order. Furthermore, the speed is limited between the maximum speed $V_{\max}$ and minimum speed $V_{\min}$. After that, we calculate the velocity of UAV i with

$$v_i(k+1) = [\cos\theta_i(k+1), \sin\theta_i(k+1)] \cdot V_i(k+1), \tag{13}$$

Finally, the location is updated by Equation (3). The details of the basic swarm algorithm are illustrated in Algorithm 1 and we abbreviate the basic Vicsek algorithm as BV-A. The computational complexity of this method is $O(n^2)$.

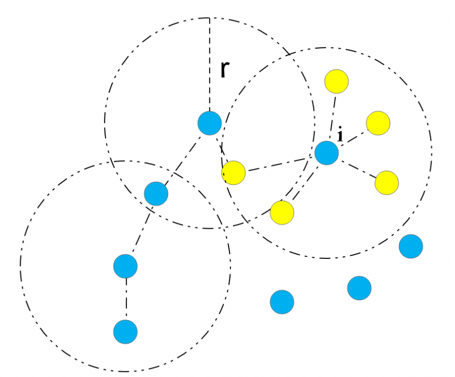


Figure 4. The basic swarm interaction method where UAV i synchronizes with all neighbors within r sphere.

In this model, the numbers of neighbors of the individual UAVs are not the same. If each individual has to communicate with all neighbors to compute the motions of all neighbors around, then the communication cost will be very high.

3.3.2. Methods Improving Consensus Convergence

In this part, four synchronizing methods created from the basic swarm algorithm are proposed. The method with the most potential is the fourth method, which combines the advantages of the law discovered by other researchers before, and adopts the method of obtaining small world characteristics by connecting with second-order neighbors proposed in this article. Other methods were used as controls to demonstrate the fourth method's superiority. Through the comparison between the performance of each group in the experiment, various specific operations that are effective for performance can be screened out.

Algorithm 1 A basic swarm algorithm for consensus convergence (BV-A).

Input: a set of UAVs U with positions x and speed $V_{\max}$,$V_{\min}$, heading angle θ, perception radius r, Maximum turning angle φ, intensity η of the random perturbation.

Output: a set of UAVs U with adjusted positions and velocities, corresponding communication topologies.

```
1:  for each timestep $k$ do
2:    for each UAV $i$ do
3:      $N_i(k) \leftarrow$ Equation (2);
4:      $\theta_i(k+1) \leftarrow$ Equation (10);
5:      if $|\theta_i(k) - \theta_i(k+1)| > \varphi$ then
6:        $\theta_i(k+1) \leftarrow$ Equation (11);
7:      end if
8:      $\Delta\theta = \eta\cdot$ uniform $(-\pi, \pi)$;
9:      $\theta_i(k+1) \leftarrow \theta_i(k+1) + \Delta\theta$;
10:     $v_{\text{co}} \leftarrow$ Equation (6);
11:     $V_i(k+1) \leftarrow$ Equation (12);
12:     $v_i(k+1) \leftarrow$ Equation (13);
13:     $x_i(k+1) \leftarrow$ Equation (3);
14:     $i \leftarrow i+1$;
15:   end for
16:   $k \leftarrow k+1$;
17: end for
```

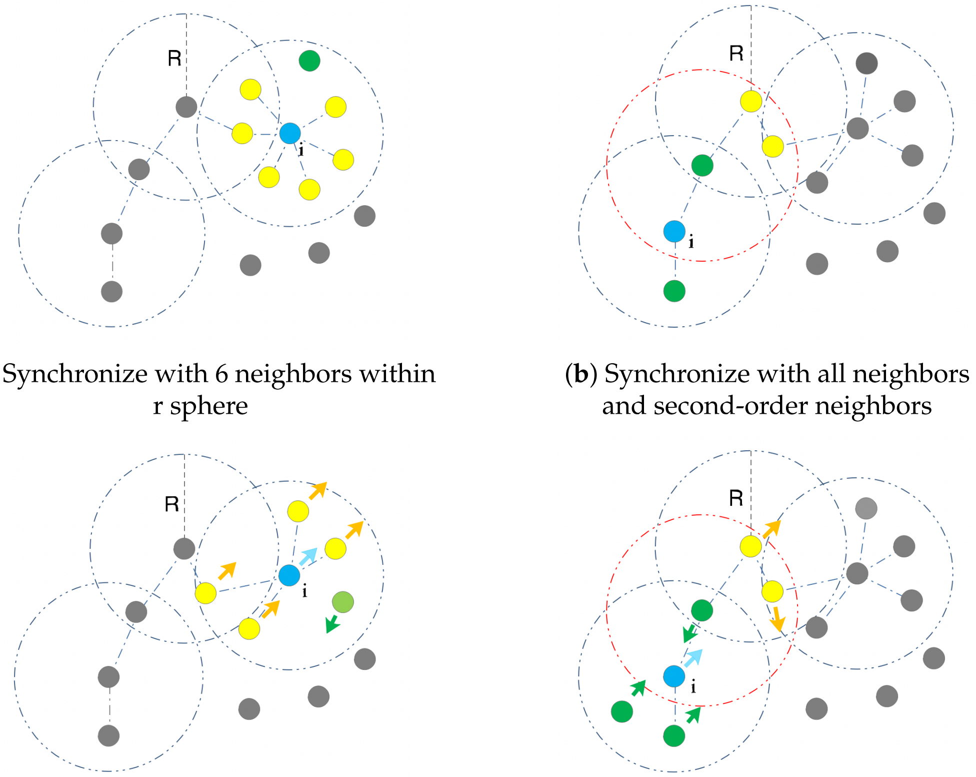
In [20], it was verified that the individuals in the bird flocks in nature actually only synchronize with a few neighbors around them. Hence, it is worth trying to synchronize with a fixed number of individuals. Firstly, we introduce a method based on six neighbors' velocity information and abbreviate the swarm intelligence inspired method communicating with six closest neighbors as SI-CS. In this swarm interaction method, shown in Figure 5a, the UAV i will choose the six closest neighbors from N_i denoted as N_i' to update the positions and velocities.

Secondly, we propose a method of synchronizing with all neighbors and second-order neighbors in Figure 5b.We abbreviate it as S-A (the method that communicates with all second-order neighbors). Differently from SI-CS, S-A extends the range of neighbors to all neighbors and second-order neighbors, which increases the information perception of a single UAV and also increases the burden of the system. The algorithm is different from Algorithm 1 in line 3 where the N_i in S-A contains neighbors of UAV i and neighbors' neighbors, and we denote it as N_i^2.

Thirdly, we came up with a method based on neighbors with specific velocity attributes. In other methods, no consideration is given to selecting specific neighbors for synchronization based on the velocity information. Among all the neighbors, neighbors in N_i' $(N_i' \subset N_i)$ could be more important for the decision of UAV i. Since the UAVs in the swarm have to quickly reach a final agreement, neighbors with bigger differences from i may have a better influence on UAV i, as shown in Figure 5c. Each UAV chooses six neighbors with the most different directions of velocities, denoted as N_i'. ξ indicates the size of the difference, and it can be calculated by Equation (14). $\xi_{ij}(k) = 1$ when $\theta_i = \theta_j$. It means the smallest difference between UAV i and UAV j. $\xi_{ij}(k) = -1$ when $\theta_i = \theta_j + \pi$ or $\theta_i = \theta_j - \pi$, which means the largest difference. We abbreviate this method as SI-SDP (swarm intelligence inspired method with speed difference preference). This algorithm is different from Algorithm 1 in line 3 where the N_i in SI-SDP contains six most different neighbors at most, and we denote it as N_i'.

$$\xi_{ij}(k) = \left[\cos\theta_i(k), \sin\theta_i(k)\right] \cdot \left[\cos\theta_j(k), \sin\theta_j(k)\right], \tag{14}$$

where $\theta_i(k)$ and $\theta_j(k)$ are the headings of UAV i and UAV j.



(**a**) Synchronize with 6 neighbors within r sphere

(**b**) Synchronize with all neighbors and second-order neighbors

(**c**) Synchronize with several specific neighbors with bigger ξ

(**d**) Synchronize with several certain neighbors and second-order neighbors with bigger ξ

Figure 5. Four synchronization scenarios generated by the four algorithms evolved from the basic swarm algorithm. (**a**) UAV i only synchronizes with six yellow neighbors and ignores the presence of the green neighbor. (**b**) UAV i synchronizes with all the green neighbors and the yellow second-order neighbors, which puts high demands on the communication link of the system. (**c**) UAV i pays more attention to green neighbors who are quite different from itself in velocity and ignore the yellow neighbors. (**d**) UAV i pays more attention to the six neighbors who are most different from itself in velocity among all the neighbors and second-order neighbors.

At last, we propose the most promising method. This method is based on certain neighbors and second-order neighbors with specific velocity attributes, and we abbreviate it as SI-WS (swarm intelligence inspired method with small world characteristics). We use the concept of second-order neighbors proposed in Definition 2. We denote the neighbors and second-order neighbors of UAV i as N_i^2. It chooses six neighbors the with greatest differences in velocity among N_i^2, as SI-SDP does through reference neighbors $N_i^{2'}$ (see Figure 5d). Compared to other methods, SI-WS chooses neighbors from a larger range, so it can get neighbors with bigger differences in velocity. In this way, not only can the neighbors with bigger differences be selected, but also the construction of long connecting edges can be realized. Compared with the previous methods from the perspective of a complex network, it is equivalent to adding long edges to the regular network so that it can improve the algebraic connectivity of the system. Furthermore, it is expected that the communication cost would be reduced by communicating with fewer neighbors in $N_i^{2\prime}$ while ensuring the same convergence performance.

4. Performance Evaluation

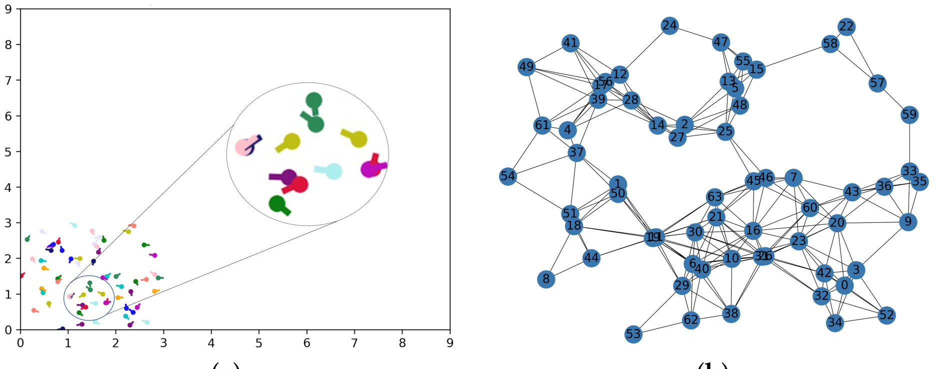
Numerical simulations were performed to verify the effectiveness of the proposed algorithms. Simulations were performed on a platform with 2.20 GHz Intel Core i7 CPU, 16 GB RAM. The algorithms we proposed were implemented in python.

4.1. Experimental Settings

UAVs flying in the 2D plane follow the rules described in Section 3.3. In order to make the experimental conclusions less affected by the initial conditions, a variety of initial position layouts and angle directions were chosen for the experiments. Multiple sets of experiments were carried out to ensure the reliability of the experimental conclusions. In the experiments, 64 UAVs were engaged in the swarm, the whole experiments were divided into five groups of experiments under different initial conditions corresponding to five different methods. The flight space of the swarm was set to a square area with a periodic boundary with a side length of 9 km. A periodic boundary is the boundary where UAVs come out of the area and re-enter the area from another side. Further, the initial positions of the swarm were distributed in a square area with a side length of 3 km in the lower left corner of the flight space. The reason why all UAVs were only distributed in a small area and not in the full plane was to avoid being affected by the periodic boundary, so that UAVs on the edge of the boundary which were not neighbors would not be misjudged as neighbors. Inspired by research in [4], we knew that bird flocking in reality is not fully linked, where each bird synchronizes with the surrounding six birds to achieve swarm behavior, so we adjusted the perception radius r of the UAVs to 0.6 km, which can ensure that the topology graph of the initial position is close to reality. The fastest speed $V_{\max}$ and slowest speed $V_{\min}$ were set to 0.05 and 0.01 km/s, respectively, so that the setting of speed and communication distance were reasonable. In order to conduct a powerful comparison in experiments, we took the presence of interference into consideration, and the intensity η of the random perturbation was set to 0.05.

4.2. Performance with Randomly Distributed Initial Positions and Directions

In this part of the experiments, the initial positions and directions of the UAVs were randomly distributed as shown in Figure 6a. The initial topology of the swarm with perception radius r is shown in Figure 6b.



(**a**) (**b**)

Figure 6. Experimental settings of the initial distribution diagram. (**a**) Flight space and initial positions, (**b**) Initial communication topology.

One hundred sets of distributions were randomly initialized, and 50 sets of experiments for five different methods were conducted.

Is SI-WS an algorithm with large algebraic connectivity which guarantees the fast consensus in velocities? SI-WS gets a large λ_2, as shown in Figure 7, though S-A has the largest λ_2. The λ_2 of SI-WS is the second largest and is about 60 percent of λ_2 of the S-A. SI-WS with a small number of links between UAVs, just like SI-CS and SI-SDP, got good performance in algebraic connectivity, mainly benefiting from the links connected to second-order neighbors.

Does SI-WS get a larger speed of consensus? Under the preset parameters, SI-CS performed the worst because it only communicated with the six nearest neighbors. Hence, its performance is much worse than BV-A, which communicates with all neighbors. Although SI-SDP does not have the advantage of BV-A in terms of the number of reference neighbors, it refers to the six neighbors with the largest velocity differences, and the directions of UAVs are much more consistent than those of BV-A. The swarm orders of S-A and SI-WS performed best, as shown in Figure 8. Both S-A and SI-WS for swarm consensus get large speeds of consensus and converged to high degrees of consistency. SI-WS was the second fastest method to achieve a consensus in the swarm. A high degree of consensus can be achieved within 10 iterations. Methods other than S-A did not get high degrees of consensus.

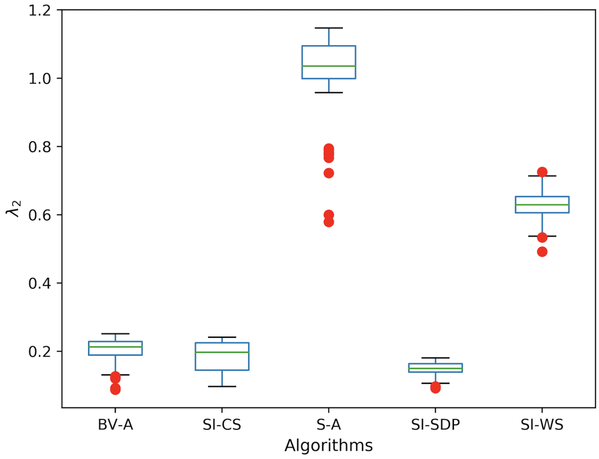


Figure 7. The algebraic connectivity λ_2 of the swarm in 40 time steps.

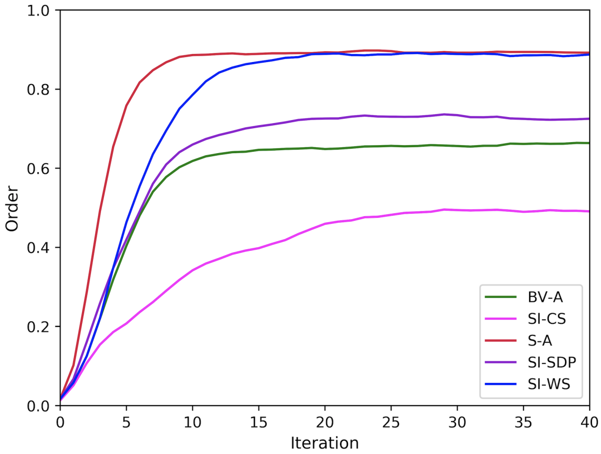


Figure 8. The order of the random distribution of initial positions.

Is the communication cost of SI-WS low enough? In Figure 9, the communication cost of SI-WS is very low compared with the other algorithms, and is about one-third less than that of BV-A. The cumulative costs of SI-CS and SI-SDP are exactly the same, because they only communicate with six of their neighbors at each time step. BV-A and S-A both have a large communication costs, and the cost of S-A is almost twice that of BV-A.

Among the three algorithms with the best results, BV-A and S-A have the largest communication costs. SI-WS has a relatively small communication cost but also achieved good results. This should be attributed to the reconnection technology for second-order neighbors. The reconnection operation introduces properties similar to the characteristics of small world networks to the swarm topology. Due to the huge number of neighbors referred to in S-A, the communication cost is double that of SI-WS, which can be seen in Figure 9. Hence, S-A is not practical in applications. All these results suggest that SI-WS has better convergence consistency and faster convergence speed and a relatively small communication cost.

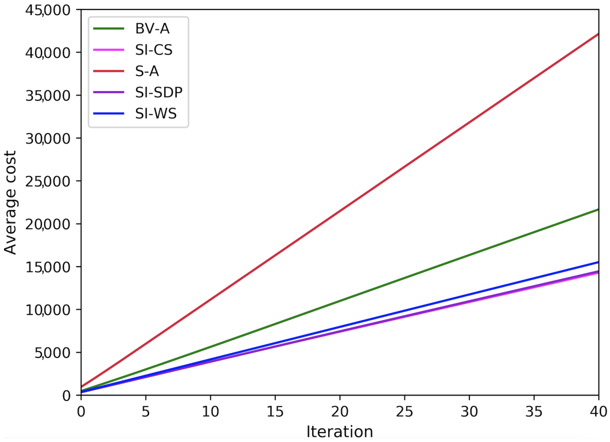


Figure 9. The cumulative communication cost of the random distribution of initial positions.

4.3. Performance with Regularly Distributed Initial Positions and Random Directions

At this point, we set the regularly distributed initial positions, and the directions were still random. The initial positions are shown in Figure 10a and the topology of the swarm was regular, as shown in Figure 10b. One-hundred sets of experiments for five different algorithms were conducted.

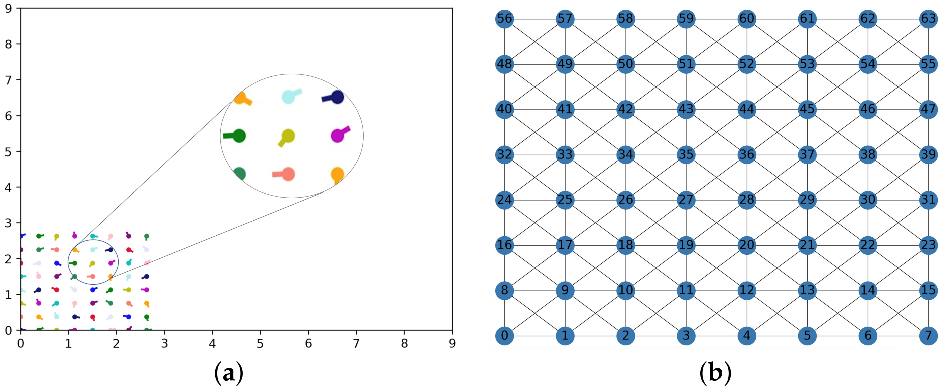


Figure 10. Diagram for the regular initial distribution. (**a**) Flight space and initial positions, (**b**) Initial communication topology.

As in the previous subsection, SI-WS still gets greater algebraic connectivity. The dynamical topology algebraic connectivity λ_2 is shown in Figure 11.

SI-WS still gets a larger speed of consensus. Both S-A and SI-WS for swarm consensus got rapid speeds and converged highly, as shown in Figure 12. The convergence speed of SI-WS was at least doubled compared to BV-A.

The performance of SI-SDP is better than that of SI-CS because it refers to the six neighbors with the largest velocity differences. This further proves that in swarm velocity consensus, UAVs with large differences have a greater effect than others on neighbors.

The communication cost of SI-WS is still very low. In Figure 13, the communication cost of SI-WS is about one-fifth less than that of BV-A. S-A has the largest communication cost, and the cost is more than twice the cost of the other algorithms.

In the situation where the initial positions were uniformly distributed, the experimental results turned out almost the same as those of the randomly distributed initial positions.

Overall, the convergence speed of our method SI-WS doubled that of the ordinary BV-A, and the communication cost was reduced by at least one-fifth. The above experimental results sufficiently verify the effectiveness of the method inspired by the swarm intelligence and small world network characteristics for rapid velocity consensus in UAV swarms.

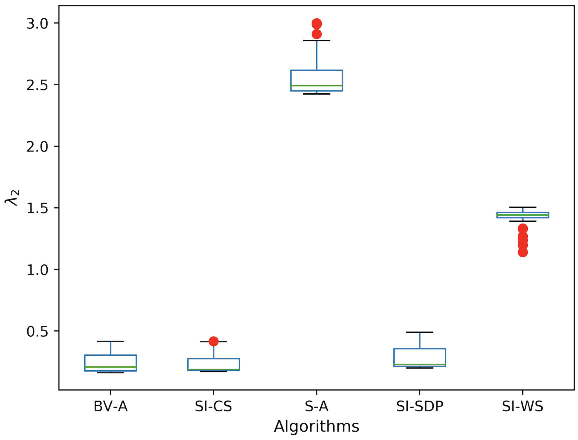


Figure 11. The topology algebraic connectivity λ_2 in 40 time steps.

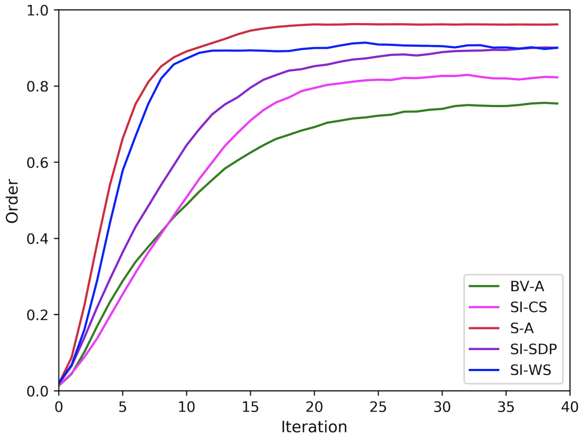


Figure 12. The order of the regular distribution of initial positions.

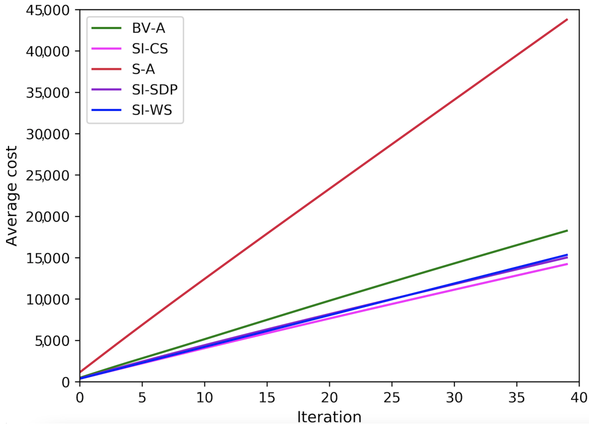


Figure 13. Cumulative communication costs with regular distributions for the initial positions.

5. Discussion

In this paper, we combined swarm intelligence with the characteristics of the small world network and proposed a method for swarm consensus convergence. Communicating with only some neighbors can reduce the burden of communication in the swarm and allow the swarm to adapt to the real environment with limited communication. When selecting UAVs with specific attributes for interaction, results show that UAVs with large differences can further improve the speed of convergence. This may be because more differentiated individuals have a better influence on neighbors. Based on the method of screening specific neighbors for communication, the communication cost is reduced. Inspired by the small world network characteristics, the method of adding second-order candidate neighbors was adopted to achieve faster convergence speed.

The relationship between convergence speed and the communication topology was revealed. It is the small world characteristic that improves the speed of swarm consensus convergence. The communication cost is further reduced when only communicating with only some neighbors. The decentralized method we proposed solves the problem of constructing an approximate small world network in a distributed situation. Although the method is effective, the construction method is relatively simple and blind, which could be analyzed in depth in future research. In addition, although our methods have achieved good results on the 2D plane, we still need to do a lot of work to transfer the method to 3D space reasonably because there is information about height in 3D space which is not involved in 2D plane. There are more constraints that need to be considered when the UAV flies in 3D space, and many situations that will not appear in the 2D plane, so our method may not be applicable. More in-depth research is needed.

6. Conclusions

In this paper, we combined small world network characteristics and the consensus convergence theory. We have verified the feasibility of applying the small world network characteristics and swarm intelligence method to the study of swarm consensus convergence. The proposed methods can be used to solve the problem of distributed swarm control when the communication is limited. Compared with the ordinary BV-A algorithm, the convergence speed of our method is greater by at least 100%, and the communication cost is reduced by at least one-fifth. The method proposed in this paper has two advantages. On the one hand, the method can reduce the communication cost of the classic Vicsek model. On the other hand, it also improves the convergence speed of swarm decision making. The new distributed algorithm for constructing an approximative small world network for a swarm network was verified to be of great effectiveness.

Author Contributions: X.J. and W.Z. proposed the method; S.C., J.C. and Z.H. designed and performed the experiments; X.J., J.L., L.L. and W.Y. analyzed the experimental data and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grants No.61806212.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare that there are no conflict of interest regarding the publication of this paper.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	unmanned aerial vehicle
UAVs	unmanned aerial vehicles
WS	small world network

PSO	Particle Swarm Optimization
BV-A	basic swarm algorithm for consensus convergence
SI-CS	swarm intelligence inspired algorithm communicating
S-A	algorithm communicating with all second-order neighbors
SI-SDP	swarm intelligence inspired algorithm with speed difference preference with six closest neighbors
SI-WS	swarm intelligence inspired algorithm with small world characteristics

References

1. Zhang, J.; Yan, J.; Zhang, P. Multi-UAV formation control based on a novel back-stepping approach. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2437–2448. [CrossRef]
2. Chen, H.; Zelazo, D.; Wang, X.; Shen, L. Convergence Analysis of Signed Nonlinear Networks. *IEEE Trans. Control Netw. Syst.* **2020**, *7*, 189–200. [CrossRef]
3. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [CrossRef]
4. Chung, T.H.; Clement, M.R.; Day, M.A.; Jones, K.D.; Davis, D.; Jones, M. Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1255–1262.
5. Olfati-Saber, R. Ultrafast consensus in small-world networks. In Proceedings of the 2005, American Control Conference 2005, Portland, OR, USA, 8–10 June 2005; pp. 2371–2378.
6. Antonelli, G.; Arrichiello, F.; Caccavale, F.; Marino, A. Decentralized time-varying formation control for multi-robot systems. *Int. J. Robot. Res.* **2014**, *33*, 1029–1043. [CrossRef]
7. Yang, A.; Naeem, W.; Irwin, G.W.; Li, K. Stability analysis and implementation of a decentralized formation control strategy for unmanned vehicles. *IEEE Trans. Control Syst. Technol.* **2013**, *22*, 706–720. [CrossRef]
8. Li, S.; He, J.; Li, Y.; Rafique, M.U. Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 415–426. [CrossRef] [PubMed]
9. Jin, L.; Li, S.; Luo, X.; Li, Y.; Qin, B. Neural dynamics for cooperative control of redundant robot manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3812–3821. [CrossRef]
10. Sauter, J.; Matthews, R.; Robinson, J.; Moody, J.; Riddle, S. Swarming unmanned air and ground systems for surveillance and base protection. In Proceedings of the AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference, Seattle, WA, USA, 6–9 April 2009; p. 1850.
11. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 27–31 July 1987; pp. 25–34.
12. Pearce, D.J.; Giomi, L. Linear response to leadership, effective temperature, and decision making in flocks. *Phys. Rev. E* **2016**, *94*, 022612. [CrossRef] [PubMed]
13. Olfati-Saber, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [CrossRef]
14. Rubenstein, M.; Cornejo, A.; Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **2014**, *345*, 795–799. [CrossRef] [PubMed]
15. Xiao, L.; Boyd, S. Fast linear iterations for distributed averaging. *Syst. Control Lett.* **2004**, *53*, 65–78. [CrossRef]
16. Watts, D.J.; Strogatz, S.H. Collective dynamics of ’small-world’ networks. *Nature* **1998**, *393*, 440–442. [CrossRef] [PubMed]
17. Klavins, E. Communication complexity of multi-robot systems. In *Algorithmic Foundations of Robotics V*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 275–291.
18. Fiedler, M. Algebraic connectivity of graphs. *Czechoslov. Math. J.* **1973**, *23*, 298–305. [CrossRef]
19. Merris, R. Laplacian matrices of graphs: A survey. *Linear Algebra Appl.* **1994**, *197*, 143–176. [CrossRef]
20. Genter, K.L. Fly with Me: Algorithms and Methods for Influencing a Flock. Ph.D. Thesis, The University of Texas at Austin, Austin, TX, USA, 2017.