



Two-Stage Flocking Formation Control Based on Constrained Boundary

Xiaodong Zheng^(✉), Tao Wang, Shuangsi Xue, Jinyu Liu, Hui Cao,
and Feihu Hu

School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China
zxd_xjtu@stu.xjtu.edu.cn

Abstract. Flocking is a special behavior that exists in the natural world. In this paper, a two-stage flocking formation algorithm is presented. The algorithm utilizes a varied range boundary to limit the agents in the group. As time increases, the agents tend to form an α -lattice structure formation. The algorithm improved the converging speed of the flock formation. Simulations and analysis are provided to demonstrate the algorithm for 2D cases.

Keywords: Constrained flocking · Graph theory · Complex network

1 Introduction

In animal herds, a unique large group behavior existed is called “Flocking” [9], which refers to a large number of individuals interacting with each other. This behavior is prevalent to see in the natural world, such as the migration of wildebeest in East Africa, a school of fish salmon avoiding the predation of sea lions, and the doves hovering above a square freely but organized without a leader.

The flocking phenomenon is very enlightening for engineering practice [6, 11]. In modern society, the behavior of large groups of automated vehicles becomes more and more essential and gradually comes to the front. By 2030, more than 20 million Tesla pure electrical cars will be sold annually, which means there will be one in ten cars that are self-driving on the roads. Although each automated car may be smart enough to avoid collision and find the optimal route to the destination, from a higher-level perspective, the large group of smart cars are exactly a flock, and the dynamics of it possess the properties of flocking.

On the other hand, flocking behavior is still a very challenging topic for researchers [2, 4, 8, 14]. The first thing for studying the flocking behavior is to understand how the flock is formed. Reynolds [12] initialized three necessary rules for forming a flock: First, the individual must not collide with other agents in the group. Second, each individual has to match the velocity with others while moving along with the group. Third, every individual in the group should stay close to its neighbors. These three basic rules formulate the basics for flocking.

The first rule regulates the minimum distance between individuals in the flock so that they will not crash while moving; the second rule guarantees the group of agents can move in a particular flock formation. Furthermore, the last rule ensures the flock will not be torn apart by the accumulation effects of the former two rules as time passes by.

The main contents of this paper are divided into four parts. The first section states the preliminaries, including basic concepts of graph theory, the dynamic agents, and the *alpha-lattice* structure; the second section discusses the problem of flocking and presents an algorithm to solve the problem. Experiments show the effects of the algorithm in the third section, and then the paper is concluded in the last part.

2 Preliminaries

In this section, the application of graph theory in flock forming and moving control are discussed. The basic concepts of graph theory and how they are applied to flocking are introduced. Also, the α -lattices structure is presented as it has been utilized to form the basic bricks of the flock.

2.1 Graph Topology of Flocks

A graph G consists of vertices set $\mathcal{V} = 1, 2, \dots, n$ and edges set $\mathcal{E} \in \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ [3, 5]. Each agent in the flock is represented as a node, $i \in \mathcal{V}$, and the relationship between two communicating nodes (i, j) is represented as an edge $e(i, j)$. If the $e(i, j) \in \mathcal{E} \iff e(j, i) \in \mathcal{E}$ for all edges, then the graph is said to be undirected. The interconnection relationship among nodes in the graph is represented by an adjacency matrix $\mathbf{A} = a[i, j] : i \in \mathcal{V}, j \in \mathcal{V}$. For $a[i, j] \neq 0$, then $e(i, j) \in \mathcal{E}$. If the graph is undirected and unweighted, the adjacency matrix is symmetric ($\mathbf{A} = \mathbf{A}^T$) and the elements in \mathbf{A} satisfy:

$$a_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The neighbors of a node i in an undirected graph are defined as $N_i = \{j \in \mathcal{V} : a_{ij} = 1, i \neq j, (i, j) \in \mathcal{V}\}$.

2.2 Dynamics of Agents

Consider a group of dynamic agents. The individual agent can communicate with others, and they can exchange the position and velocity information of themselves. Let the communication range for agents is $R \in \mathbb{R}$, and the safe distance between agents is $D \in \mathbb{R}$. The motion of agents [13] is described as:

$$\begin{cases} \dot{q}_i = p_i, \\ \dot{p}_i = u_i \end{cases} \quad (2)$$

where $q_i \in \mathbb{R}^n (n = 2, 3)$ represents position information of agent i , and $p_i \in \mathbb{R}^n (n = 2, 3)$ represents the velocity of agent i . $u_i \in \mathbb{R}^n (n = 2, 3)$ is the control input, which represents the acceleration for agent i in its moving direction.

For an agent i in the flock, its neighbors can be defined as:

$$N_i = \{j \in \mathcal{V}, j \neq i : \|q_j - q_i\| < R\} \tag{3}$$

where $\|*\|$ is the Euclidean norm in cartesian coordinate system, and R is the maximum communication range between agent i and agent j . From the Eq. 3, one can know for an agent i , the communication scope is a circle with radius R . Let d_{ij} expresses the distance between two agents, then:

$$d_{ij} = \|q_j - q_i\|, \quad \forall j \in N_i \tag{4}$$

To fulfill the first rule of Reynolds, i.e. every agent should not collide with each other, then, $D \leq d_{ij} \leq R$. To further regulates the safe distance and the communication range, we have:

$$R = \kappa D, \quad \kappa \geq 1 \tag{5}$$

For a graph G , the proximity net $G(q) = (\mathcal{V}, \mathcal{E}(q))$ [1] satisfies:

$$\mathcal{E}(q) = \{e(i, j) : (i, j) \in \mathcal{V} \times \mathcal{V}, d_{ij} < R, i \neq j\} \tag{6}$$

2.3 Basic Brick: α -lattices

There are many types of flock structure exits, and the *absolute - α - lattice* structure is one of the most basic structures. To form a *α - lattice* structure flock, the distance between every pair of nodes must be equal to a fixed number d :

$$d = d_{ij} = \|q_j - q_i\| \quad \forall j \in N_i \tag{7}$$

In general, the *α - lattice* structure is common to see in the natural world, but the *absolute α - lattice* is hard to form. Therefore, the constraint for forming flock by using 8 could be relaxed to :

$$d_{quasi} = d_{ij} = \|q_j - q_i\| \in [d - \delta, d + \delta] \quad \forall j \in N_i \tag{8}$$

where δ is a small distance for forming the relaxed *quasi - α - lattice*(or *quasi-crystals*) [13]. Also we could define the deviation function to measure how much the *quasi - α - lattice* structure is different from the *absolute - α - lattice* structure [10]:

$$f = \frac{1}{\|\mathcal{E}(q) + 1\|} \sum_{i=1}^n \sum_{j \in N_i} \Psi(\|\delta_{ij}\|) \tag{9}$$

and the *pairwise potential* function:

$$\Psi(x) = \frac{1}{k} \left(\frac{1}{\frac{1}{z} + 1} e^{(\frac{1}{z}x+1)} - e \right) \tag{10}$$

where $\delta_{ij} = d_{quasi} - d$, and $z > 0$.

The *pairwise energy* function has purposes for measuring the deviation energy of the formation in which it differs from α -lattice. When z is small, it is similar to the quadratic equation on the x coordinate. The deviation energy will increase quickly as the δ value increases. However, as z increases, the *pairwise potential* function in the former stage ($x > 0$) increases slower, which means that if the δ value is within a certain range, the deviation could be tolerated or alleviated, but when the δ value is getting bigger, then the punishment increases faster. A non-negative map called σ norm [10] is defined so that the distance could be differentiable at zero:

$$\|x\|_\sigma = \sqrt{1 + \frac{1}{a} \|x\|^2} - 1 \quad a \in [0, \infty] \tag{11}$$

Also a *bump function* 12 is utilized to construct a smooth potential function [10]:

$$A_h(z) = \begin{cases} 1, & z \in [0, h] \\ \frac{1}{2} [1 + \sin(\pi \frac{(z-h)}{(1-h)} + \frac{\pi}{2})], & z \in [h, 1] \\ 0, & otherwise \end{cases} \tag{12}$$

where $h \in (0, 1)$. Then, an adjacent matrix on the proximity function $G(q)$ could be:

$$\mathbf{A}(\mathbf{q}) = a_{ij}(q) = \begin{cases} A_h(\|d_{ij}\|_\sigma / \|r\|_\sigma) \in [0, 1], & j \neq i \\ 0, & i = j \end{cases} \tag{13}$$

The adjacency matrix is an indicator function that $a_{ij} = 0$ when $h = 1$. This can be interpreted as the parameter h is controlling the *adjacent range* by the ratio of actual distance between two agents and the communication range in σ domain.

2.4 Control Forces

Flocking of agents depends on three forces: the force to avoid neighboring agents, the force to make individual agents reach a consensus state, and the force to track objectives. Therefore, three control input is generated to form and moving the flocks. Olfati-Saber [13] proposed the control protocols with obstacle avoidance based on three types of agents: α agent, β agent, and γ agent. The control protocols in Olfati-Saber’s third algorithm for individual agent i is:

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma \tag{14}$$

where each of the three controls are:

$$u_i^\alpha = \zeta_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|d_{ij}\|_\sigma) \mathbf{W}_{ij} + \zeta_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \tag{15}$$

$$\phi_\alpha(z) = \Lambda_h(z/r_\alpha) \left(z/\sqrt{1+(z-\|d\|_\sigma)^2} \right) \tag{16}$$

where the $W_{i,j}$ is the gradient of the σ map function:

$$W_{ij} = \nabla\sigma(q) = \frac{q_j - q_i}{a\sqrt{1 + \frac{1}{a}\|d_{ij}\|^2}} = \frac{q_j - q_i}{1 + a\|d_{ij}\|_\sigma} \tag{17}$$

Similarly, the u_i^β is:

$$u_i^\beta = \zeta_1^\beta \sum_{j \in N_i^\beta} \phi_\beta(\|\widetilde{d}_{ij}\|_\sigma) \widehat{W}_{ij} + \zeta_2^\beta \sum_{m \in N_i^\beta} b_{im}(q)(p_m - p_i) \tag{18}$$

$$\phi_\beta(z) = \Lambda_h(z/d_\beta) (\sigma_1(z - d_\beta) - 1) \tag{19}$$

$$b_{im}(q) = \Lambda_h(\|\hat{q}_{im} - q_i\|_\sigma/d_\beta) \tag{20}$$

where the $\|\widetilde{d}_{ij}\|_\sigma$ represents sigma distance between the β agent j and agent i , and the \widehat{W}_{im} is:

$$\widehat{W}_{im} = \nabla\sigma(q) = \frac{q_m - q_i}{a\sqrt{1 + \frac{1}{a}\|d_{im}\|^2}} = \frac{q_m - q_i}{1 + a\|d_{im}\|_\sigma} \tag{21}$$

where the letter m represents a virtual β agent sticking around the surface of the obstacles, and for the γ agent:

$$u_i^\gamma = -\zeta_1^\gamma \sigma_1(q_i - q_m) - \zeta_2^\gamma (p_i - p_m) \tag{22}$$

The β agents produce external repulsive force to prevent the flock from crashing into the obstacles. The α agents and the γ agents form the α - lattice structured flock and trace a dynamic or static target. Without the α agents, the flock will eventually become fragmented because of the repulsive force among nodes. The repulsive force can change agents' initial speed and speed direction, and the agents will go fragmented when nothing restricts them.

Finally the velocity and position update protocol are:

$$V_{new}^i = V^i + u_i \Delta t, \quad P_{new}^i = P^i + V^i \Delta t + u_i \frac{(\Delta t)^2}{2} \tag{23}$$

where $V^i \in \mathbb{R}^n$, ($n = 2, 3$) is the velocity of agent, and $P^i \in \mathbb{R}^n$, ($n = 2, 3$) is the position of agent i .

3 Constrained Flocking Formation

Reza Olfati-Saber [13] proposed several algorithms for forming α lattice flock utilizing a multi-species framework. Th flock formed eventually becomes a lattice-like shape and can chase static or moving objects. However, the algorithms proposed by Reza [13] cannot realize formation control and strongly depend on the

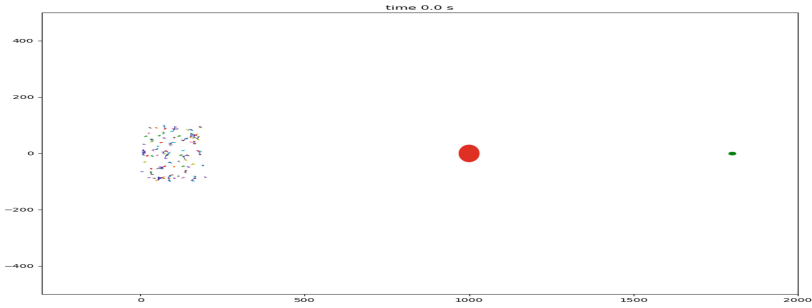


Fig. 1. Initial distribution: red disk is the obstacles, green point is the target

parameter settings. Inappropriate parameter settings will bring fragmentation to the flock. Also, if the initial distributions of individuals are too separated and the communication range for agents is limited, the flock will not be formed. Even if the majority of the agents are formed a flock, it is still possible to lose some *outliers* who have no neighbors at all in the beginning for an extended period. In this section, a constrained flocking formation is proposed to solve this problem by using a constrained boundary at the beginning of the formation.

3.1 Long Time Convergence

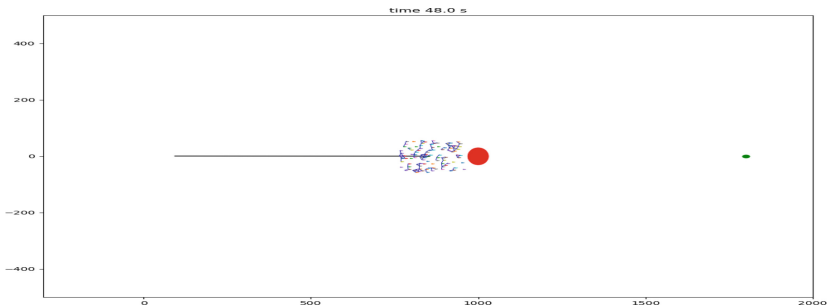


Fig. 2. The group of agents hasn't form complete flock until they meet the obstacles

Using Olfati's algorithm with the initial position of agents set as $\mathcal{X}, \mathcal{Y} \sim U(0, 200)$ (Fig. 1) and sensing distance set as 10, the result shows that the group of agents will spend a long time converging to a flock if the target is set far from the group's initial position (Fig. 2).

3.2 Constrained Flocking Shape Formation

In this section, a boundary constrained algorithm is presented for controlling the initial flocking formation shape. Using a virtual constraint boundary, the flock

is formed much quicker than merely using a single target point. The flocking process is divided into two stages. In the first stage, the agents are constrained to form an α – lattice structured shape that can support them to fly together later. After the formation is complete, the constrained boundary is eliminated, and the flock starts chasing their predetermined target.

3.3 Algorithms

To achieve the first stage of convergence, the group of agents has to find the central position of the flock as the initial target. Whenever the distance between agents is close enough, which means agents can sense others, they will establish connections. Also, a *time-varied-ranged* constraints boundary is built to prevent agents from flying out of scope and give an initial push to the agents initialed on the boundary. To achieve a line boundary, the following lemma is used:

Lemma 1. *For a hyperplane boundary with unit normal \mathcal{O}_m and passes through the point m , the position and velocity of the β agent are determined by:*

$$\hat{q}_{i,m} = Kq_i + (I - K)m, \quad \hat{p}_{i,m} = Kp_i \tag{24}$$

where $\hat{q}_{i,m}$ is the point that the boundary passes through, and $K = I - \mathcal{O}_m \mathcal{O}_m^T$, and the safe range between obstacles and agents could be changed by time $D(t)_m \propto \epsilon t, \epsilon \in \mathbb{R}$.

Proof. See [10] Lemma 3

With Lemma 1, the next thing is to determine how many point we need so that they can form a line. Consider a line with width D' , assume the initial desired distance from the wall is $D_m(0)$, the number of points needed n is easy to calculate, and the control input is also modified: $n = \text{roundoff}(\frac{D'}{D(0)_m}) + 1$ and the γ control input is changed to $u_i = u_i^\alpha + u_i^\beta + u_i^\gamma$, where u_i^γ is modified to:

$$u_i^{\gamma'} = -\zeta_1^{\gamma'} \sigma_1 (q_i - q_m) + \text{Constant} \tag{25}$$

We can use the *relative connectivity* [7, 13] to judge when to stop the first stage and go to the second stage:

$$C = (1/n - 1) \text{rank}(\mathbf{A}'(G(q))) \in [0, 1] \tag{26}$$

where \mathbf{A}' is the [0–1] indicator adjacent matrix (not the spatial adjacent matrix defined by 13).

Algorithm 1. Two Stage flocking

```

1: First Stage:
2: Initialize a population of agents with uniform random values positions
3: Select the geometric center as the first stage target
4: while Relative Connectivity condition not reached do
5:   for Each agent  $i$  do
6:     Update velocity and position of the agent using 23 with control input 25
7:     Evaluate the Relative Connectivity  $C$ 
8:     if  $C \geq 1 - \delta$ , ( $0 \leq \delta \ll 1$ ) then
9:       Go to Second Stage
10:    end if
11:  end for
12: end while
13: Second Stage:
14: Remove the virtual constrained boundary and reset the objective to the original
    tracing target
15: while Objective is not reached do
16:   for Each agent  $i$  do
17:     Update velocity and position of the agent using 23 with control input 14
18:     Evaluate the center of mass  $P_c$ 
19:     if the center of mass reaches target then
20:       break while
21:     end if
22:   end for
23: end while

```

4 Experiment

Due to the limited length of this article, the experiment only shows Stage 1 as the algorithm shows. The experiment selects 100 nodes with a size of 10 randomly distributed in a square region centered at 0, and the distribution of X,Y coordinates complies $\mathcal{X}, \mathcal{Y} \sim U(0, 300)$ in a 2D plane. The boundary set is bigger than the distributed region of agents. The parameter for the control is set as $c_1^\alpha = 40$, $c_1^\beta = 1500$, $c_1^\gamma = 60$.

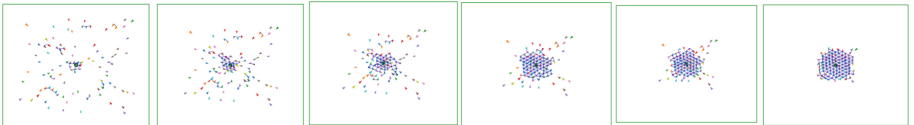


Fig. 3. Constrained flock formation

Figure 3 shows the evolution of the formation of the flock. The agents are assigned a random direction and velocity in the beginning. The green boundary limits the agents so that they will not fly out by random movement. The green point is the stage 1 target, which produces a centripetal force for all agents. Also, every agent will have the pairwise resisting force to avoid the collision, and the boundary also produces resisting force to alter the velocity and direction of the agents who are close to it. Figure 4 shows the shape and statistical network properties of the flock at $t = 0$ s and $t = 12$ s (0.1 s is the step size). In the beginning, the degree, clustering coefficient, and the number of triangles that pass through each agent are small or zero, but after the α -lattice structure is formed, the values are changed and tend to be stable. The trajectories plot Fig. 5 shows that all agents are going towards center of the flock. However, once the agents have reached the center, they begin changing their directions locally, just like *marching on the spot*.

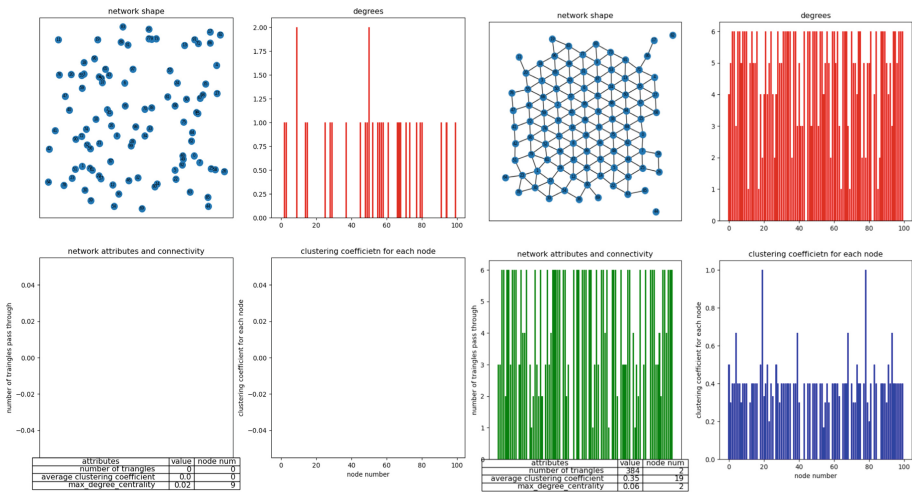


Fig. 4. Flocking formation shape and network attributes

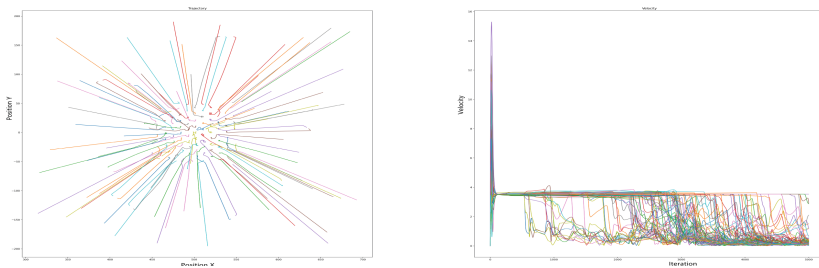


Fig. 5. Trajectories of agents and the velocities

5 Conclusion

In this paper, the flocking behavior is stated. The problem in the early stage of flock formation is solved by utilizing a constrained boundary algorithm. The experiment results show the converging speed for the flocking formation is improved, and the α – lattice structure is presented in the formation. However, there are still some future works to do to better control the formation of more complex structures, and the algorithm’s complexity is about to be improved further.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Commun. Mag. (COMMAG)* **40**(8), 102–114 (2002). <https://doi.org/10.1109/MCOM.2002.1024422>
2. Bhowmick, C., Behera, L., Shukla, A., Karki, H.: Flocking control of multi-agent system with leader-follower architecture using consensus based estimated flocking center, pp. 166–171 (2016). <https://doi.org/10.1109/IECON.2016.7793149>
3. Bollobás, B.: *Modern Graph Theory*. GTM, vol. 184. Springer, New York (1998). <https://doi.org/10.1007/978-1-4612-0619-4>
4. Deghat, M., Anderson, B.D.O., Lin, Z.: Combined flocking and distance-based shape control of multi-agent formations. *IEEE Trans. Autom. Control* **61**(7), 1824–1837 (2016). <https://doi.org/10.1109/TAC.2015.2480217>
5. Diestel, R.: *Graph Theory*. Graduate Texts in Mathematics. Springer, Heidelberg (2005). <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/3540261826>
6. Garcia, E., Cao, Y., Casbeer, D.W.: Periodic event-triggered synchronization of linear multi-agent systems with communication delays. *IEEE Trans. Autom. Control* **62**(1), 366–371 (2017). <https://doi.org/10.1109/TAC.2016.2555484>
7. Jian, L., Hu, J., Shi, K.: Consensus of linear multi-agent systems by distributed event-triggered control with functional observers. In: *Chinese Control Conference, CCC 2019*, 1 July, pp. 5770–5775 (2019). <https://doi.org/10.23919/ChiCC.2019.8865173>
8. Khan, S., Momen, S., Mohammed, N., Mansoor, N.: Patterns of flocking in autonomous agents, pp. 92–96 (2018). <https://doi.org/10.1109/ICoIAS.2018.8494115>
9. Okubo, A.: Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. *Advances Biophys.* **22**, 1–94 (1986)
10. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Autom. Control* **51**(3), 401–420 (2006). <https://doi.org/10.1109/TAC.2005.864190>
11. Ren, W., Beard, R.W.: *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, London (2008). <https://doi.org/10.1007/978-1-84800-015-5>
12. Reynolds, C.W.: *Flocks, herds and schools: a distributed behavioral model* (1987). <https://doi.org/10.1145/37401.37406>

13. Saber, R., Murray, R.: Flocking with obstacle avoidance. In: Proceedings of the 42nd IEEE Conference on Decision and Control, vol. 2, pp. 2022–2028 (2003). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.7181>
14. Zheng, X., Xue, S., Cao, H., Wang, F., Zhang, M.: A cost-efficient smart IoT device controlling system based on Bluetooth mesh and cloud computing, pp. 3374–3379 (2020). <https://doi.org/10.1109/CAC51589.2020.9326634>