

Research Article

An Improved Fast Flocking Algorithm with Obstacle Avoidance for Multiagent Dynamic Systems

Jialiang Wang,¹ Hai Zhao,¹ Yuanguo Bi,¹ Shiliang Shao,² Qian Liu,¹ Xingchi Chen,¹
Ruofan Zeng,¹ Yu Wang,¹ and Le Ha¹

¹ College of Information Science & Engineering, Northeastern University, Shenyang 110819, China

² Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

Correspondence should be addressed to Yuanguo Bi; biyuanguo@ise.neu.edu.cn

Received 8 October 2013; Revised 12 May 2014; Accepted 24 May 2014; Published 24 June 2014

Academic Editor: Jin L. Kuang

Copyright © 2014 Jialiang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Flocking behavior is a common phenomenon in nature, such as flocks of birds and groups of fish. In order to make the agents effectively avoid obstacles and fast form flocking towards the direction of destination point, this paper proposes a fast multiagent obstacle avoidance (FMOA) algorithm. FMOA is illustrated based on the status of whether the flocking has formed. If flocking has not formed, agents should avoid the obstacles toward the direction of target. If otherwise, these agents have reached the state of lattice and then these agents only need to avoid the obstacles and ignore the direction of target. The experimental results show that the proposed FMOA algorithm has better performance in terms of flocking path length. Furthermore, the proposed FMOA algorithm is applied to the formation flying of quad-rotor helicopters. Compared with other technologies to perform the localization of quad-rotor helicopter, this paper innovatively constructs a smart environment by deploying some wireless sensor network (WSN) nodes using the proposed localization algorithm. Finally, the proposed FMOA algorithm is used to conduct the formation flying of these quad-rotor helicopters in the smart environment.

1. Introduction

Flocking is the behavior exhibited when a group of birds, called a flock, are foraging or in flight. There are similar behaviors such as the shoaling behavior of fish, the swarming behavior of insects, and the herd behavior of land animals [1].

Reynolds [2] firstly introduced three rules including flocking centering, obstacle avoidance, and velocity matching to imitate flocking by computer simulations. In [3, 4], Reynolds further explains the three rules. Toner and Tu [5] and Shimoyama et al. [6] conduct researches about multiagent flocking. Toner adopts a continuous model to analyze multiagent flocking. Shimoyama et al. propose to describe behavior of multiagents based on particle system and use function rules of overall perception to consider

the flocking of multiagent. Mogilner and Edelstein-Keshet [7, 8] use a continuous model to research the flocking of multiagent, and they consider the practical sensing problems.

Jadbabaie et al. [9] conduct researches on the flocking behavior of multiagent based on the rule of the nearest agent and verify that the agents tend to move with the same direction under the assumption that all agents can detect each other.

Quadrotor helicopters have been widely used nowadays due to their unique advantages. Flocking has also been considered as an approach to control the behavior of unmanned air vehicles (UAVs). Flocking is a common technology in screensavers and has been successfully used in animations. Flocking has been widely used in many films [10] to generate crowding. Tim Burton's *Batman Returns* (1992) features flocking bats, and Disney's *The Lion King* (1994)

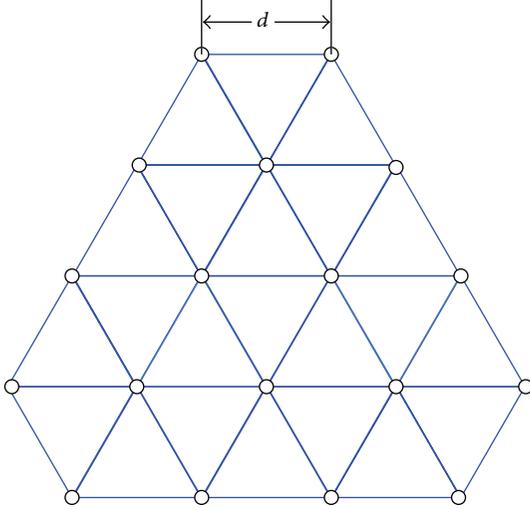


FIGURE 1: Architecture of lattice.

includes a wildebeest stampede. Flocking behavior has been used for other interesting applications. For example, it has been applied to Internet multichannel radio stations [11], information visualization [12], and optimization tasks [13].

Lattice graph is used in order to describe the relationship between different agents. For multiagent dynamical systems, if the configurations of agents satisfy the constraint that $\|q_j - q_i\| = d, \forall i, j \in N_f$, where q_j and q_i represent the positions of agents j and i , respectively, and N_f denotes the set of all of the agents, these agents form a lattice as shown in Figure 1(a); then, d is the goal distance (the length of side for the lattice). However, if the configurations of agents satisfy the constraint that $-\delta \leq \|q_j - q_i\| - d \leq \delta, \forall i, j \in N_f$, then these agents form a quasilattice due to the movements of the agents. To form a lattice is the ideal status for the multiagent dynamical systems [2].

2. Olfati-Saber Algorithm of Obstacle Avoidance for Multiagent Dynamical System

Olfati-Saber proposed an algorithm of obstacle avoidance [14]. In this algorithm, obstacle is regarded as a moving agent, and system merges this velocity and the velocity of that agent entering the region of obstacle and then regards the merged velocity as next-time velocity for the agent. The magnitude of velocity, but not its direction, is considered while merging velocity. Besides, obstacle is still regarded as one agent while it is moving away from the obstacle, so this will disturb the behavior of the agent and even the process of obstacle avoidance for these agents.

In the algorithm, the system input is given by

$$u_i = u_i^\alpha + u_i^\beta + u_i^\gamma, \quad (1)$$

where α, β , and γ represent three different agents and u_i^α, u_i^β , and u_i^γ are the interaction mechanisms for the agent set

named $(\alpha, \alpha), (\alpha, \beta)$, and (α, γ) , respectively, and u_i^γ is also the item of feedback control.

The calculation of the three input parameters is given by

$$\begin{aligned} u_i^\alpha &= c_1^\alpha \sum_{j \in N_i^\alpha} \varphi_\alpha (\|q_j - q_i\|_\sigma) n_{i,j} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij} (q) (p_j - p_i), \\ u_i^\beta &= c_1^\beta \sum_{j \in N_i^\beta} \varphi_\beta (\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} \\ &\quad + c_2^\beta \sum_{j \in N_i^\beta} b_{i,k} (q) (\hat{p}_{i,k} - p_i), \\ u_i^\gamma &= -c_1^\gamma \sigma_1 (q_i - q_\gamma) - c_2^\gamma (p_i - p_r), \end{aligned} \quad (2)$$

where $\sigma_1(z) = z/\sqrt{1 + \|z\|^2}$, c_η^γ is a positive constant, and $\eta = 1, 2, \nu = \alpha, \beta, \gamma$. (q_r, p_r) means the position and velocity of γ agent, respectively. Vectors $n_{i,j}$ and $\hat{n}_{i,k}$ are the direction vectors used to describe the relation of interaction for these agents, and they are given by

$$n_{ij} = \frac{q_j - q_i}{\sqrt{1 + \varepsilon \|q_j - q_i\|^2}}, \quad \hat{n}_{i,k} = \frac{\hat{q}_{i,k} - q_i}{\sqrt{1 + \varepsilon \|\hat{q}_{i,k} - q_i\|^2}}. \quad (3)$$

Supposing an obstacle O_k whose radius and center are R_k and y_k , respectively, and that there is an α agent whose status is (q_i, p_i) , then the position $\hat{q}_{i,k}$ and the velocity $\hat{p}_{i,k}$ of β spherical agent $((i, k) \in \nu_\alpha \times \nu_\beta)$ are, respectively, defined by

$$\begin{aligned} \hat{q}_{i,k} &= \mu q_i + (1 - \mu) y_k, \\ \hat{p}_{i,k} &= \mu P p_i, \end{aligned} \quad (4)$$

where $\mu = R_k/\|q_i - y_k\|$, $a_k = (q_i - y_k)/\|q_i - y_k\|$, and $P = I - a_k a_k^T$.

If the interaction mechanism between the two agent sets named (α, α) and (α, γ) is fixed. Analyzing the interaction mechanism of (α, β) agent, there are two parts for the mutual effect; the first one is repulsive interaction between β agent and α agent, and the second part is the velocity matching between them. The two parts exist all the time under the condition of β agent entering the perception region of α agent, which means that the neighbor condition of α agent obstacle that $N_i^\beta = \{j \in \nu_\beta : \|\hat{q}_{i,k} - q_i\| < r'\}$ is met. Besides, the value of c_1^β and c_2^β does not change all the time.

After analyzing the existing algorithm, we know that the algorithm has the following flaws.

- (a) It only judges the distance of α agent and obstacle. While the distance is smaller than the distance perception, then the system will execute the process of obstacle avoidance, but without considering the velocity direction of α agent, which means that the velocity direction of α agent does not move toward the obstacle; under this case, if α agent can detect the obstacle, the factor of $c_2^\beta \sum_{j \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i)$ while calculating the u_i^β will block the move path while α agent is moving away from obstacle.
- (b) It only obtains the velocity matching between α agent and β agent, which means that the system aims to change the velocity direction of α agent, till it tends to the same with β agent; thus, the final velocity direction of α agent is uncertain. So this is not beneficial for the form of flocking and access to target point.

The major contribution of this paper is to propose an improved algorithm for obstacle avoidance, which aims to make the agents fast form flocking and approach the destination point; at the meantime, all agents can effectively avoid the obstacles. In addition, this paper innovatively constructs a smart environment to implement the localization of quadrotor helicopters, and the proposed FMOA algorithm is used to conduct the formation flying of quadrotor helicopters in the smart environment.

3. Fast Multiagent Obstacle Avoidance (FMOA) Algorithm

The proposed algorithm is considered under the status of whether the flocking has formed. If there exists at least one agent having the same velocity with α agent and the distance between them reaches the given value d , then it means that the flocking has formed.

If the action range of α agents is r , then the dangerous radius of an obstacle is $d_{\text{dan}} = R_k + r_{\text{safe}}$, where R_k denotes the radius of obstacle and r_{safe} represents the predefined safe region of obstacle. Besides, using d_q denotes the vector between α agent and the center of obstacle as Figure 2 shows, and the tangent vector between α agent and dangerous region of obstacle is defined as d_{t1} and d_{t2} , respectively.

After agents detect the dangerous region of obstacle (while $d' \leq r$), the system judges the direction of its velocity; if it moves within the region of obstacle (while $\det[d_{t1}, p_i] < 0 \wedge \det[d_{t2}, p_i] > 0$), the system will execute the process of obstacle avoidance. Otherwise, no obstacle exists.

While α agent enters the dangerous region of obstacle, the system regards the obstacle as β agent; the magnitude and direction of its velocity for α agent are discussed by the status of whether flocking has formed.

System can judge whether α agent enters the dangerous region of β agent by calculating the value of $\det[d_{t1}, p_i]$, $\det[d_{t2}, p_i]$, and $\det[d_q, p]$, which are defined by

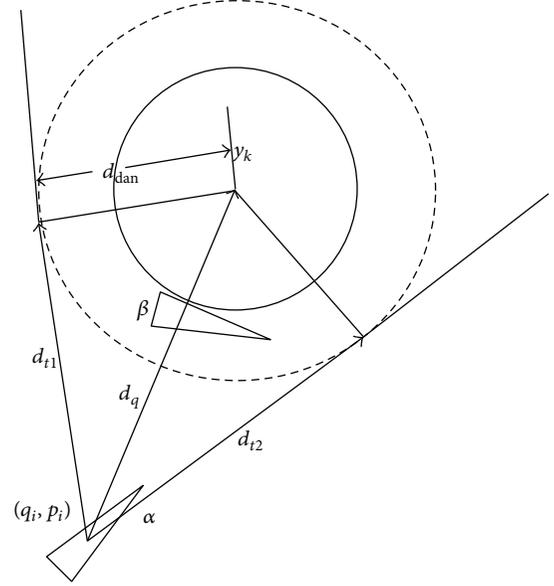


FIGURE 2: The description of agent and dangerous area.

$$\begin{aligned} \det [d_q, p] &= \det \begin{bmatrix} d_x & p_x \\ d_y & p_y \end{bmatrix} = d_x p_y - d_y p_x \\ &= [d_x, d_y] \begin{bmatrix} p_y \\ -p_x \end{bmatrix}, \end{aligned} \quad (5)$$

where $[d_x, d_y] \begin{bmatrix} p_y \\ -p_x \end{bmatrix} = d_q T$; thus, $T = \begin{bmatrix} p_y \\ -p_x \end{bmatrix}$.

So the above judgment can be simplified to judge its value. If vector $d_q T \geq 0$, it means that the velocity vector is above the median vector as Figure 3(a) shows; oppositely, if vector $d_q T < 0$, it represents that the velocity vector is under the median vector as Figure 3(b) shows.

3.1. Flocking Has Not Formed. The target point γ is the final goal for α agent. So, while agents avoid obstacles, the target point γ is the guided direction for these agents.

(a) While γ agent is located in the region as Figure 4(a) shows, then the velocity direction of α agent should be altered to the tangential direction of obstacle as $p_{\beta 1}$, which makes it avoid obstacle with the least deviation:

$$p_{\alpha 1} = \begin{cases} \frac{\|p_i\|_\sigma \cdot (q_i - q_\gamma)}{\|(q_i - q_\gamma)\|_\sigma}, & \{d' \leq r\} \cap \{\det [d_{t1}, p_i] < 0 \wedge \det [d_q, p_i] > 0 \wedge \det [d_{t1}, d_\gamma] \geq 0\} \\ \frac{\|p_i\|_\sigma \cdot (q_i - q_\gamma)}{\|(q_i - q_\gamma)\|_\sigma}, & \{d' \leq r\} \cap \{\det [d_q, p_i] < 0 \wedge \det [d_{t2}, p_i] > 0 \wedge \det [d_{t2}, d_\gamma] \leq 0\}. \end{cases} \quad (6)$$

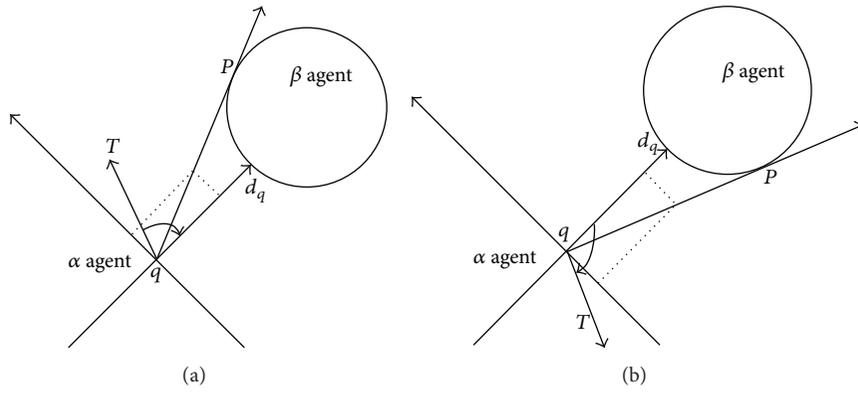


FIGURE 3: The relative position relationship between agent and obstacle.

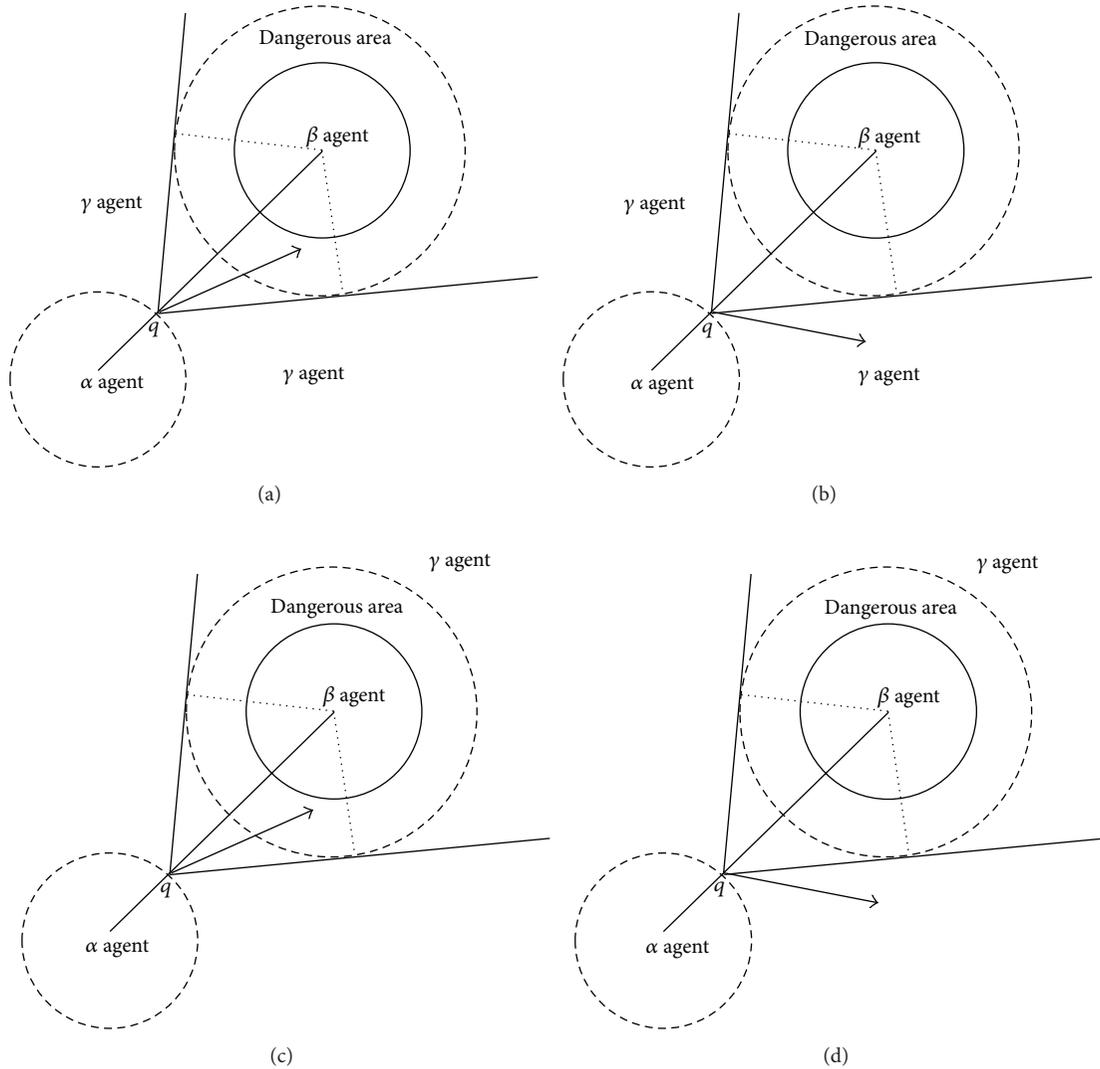


FIGURE 4: Different conditions between agents while flocking has not formed.

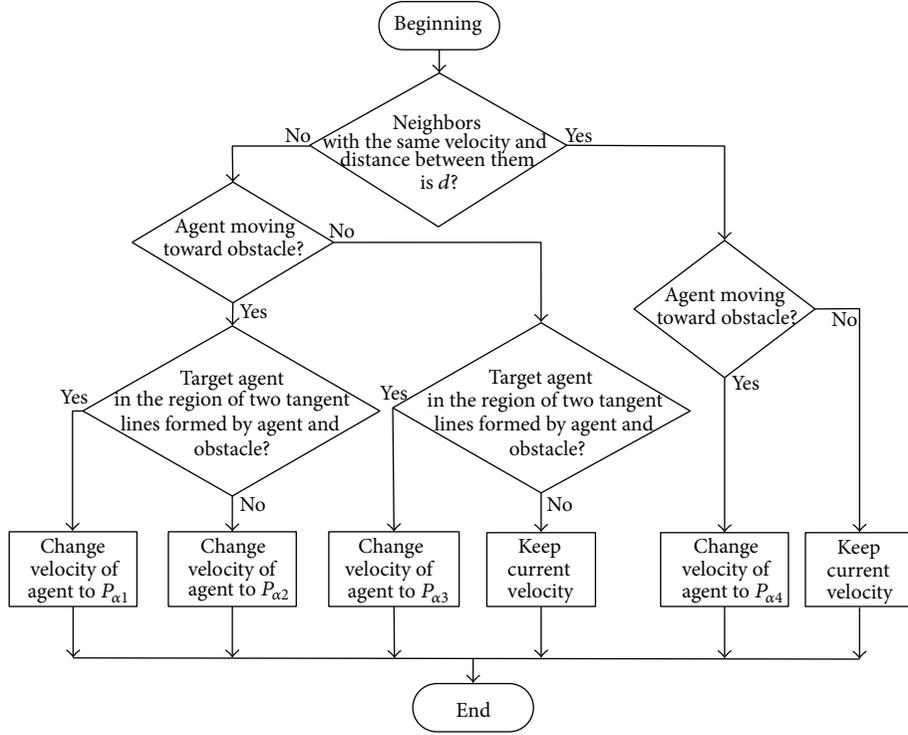


FIGURE 5: The workflow of FMOA algorithm.

(b) While γ agent is located in the region as Figure 4(b) shows, then the velocity direction of α agent should be altered to the target point. In this case, system does not change the velocity direction of α agent, because the obstacle will not affect the movement of α agent.

(c) While γ agent is located in the region as Figure 4(c) shows, then the velocity direction of α agent should be altered to the tangential direction. The velocity of α agent is given by

$$P_{\alpha 2} = \begin{cases} \frac{\|P_i\|_{\sigma} \cdot d_{t1}}{\|d_{t1}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_{t1}, P_i] < 0 \wedge \det [d_q, P_i] > 0 \wedge \det [d_{t1}, d_{\gamma}] < 0 \wedge \det [d_{t2}, d_{\gamma}] > 0 \} \\ \frac{\|P_i\|_{\sigma} \cdot d_{t2}}{\|d_{t2}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_q, P_i] < 0 \wedge \det [d_{t2}, P_i] > 0 \wedge \det [d_{t1}, d_{\gamma}] < 0 \wedge \det [d_{t2}, d_{\gamma}] > 0 \}. \end{cases} \quad (7)$$

(d) While γ agent is located in the region as Figure 4(d) shows, then the velocity direction of α agent should be altered to the tangential direction. The velocity of α agent is given by

$$P_{\alpha 3} = \begin{cases} \frac{\|P_i\|_{\sigma} \cdot d_{t1}}{\|d_{t1}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_{t1}, P_i] > 0 \wedge \det [d_{t1}, d_{\gamma}] < 0 \wedge \det [d_{t2}, d_{\gamma}] > 0 \} \\ \frac{\|P_i\|_{\sigma} \cdot d_{t2}}{\|d_{t2}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_{t2}, P_i] < 0 \wedge \det [d_{t1}, d_{\gamma}] < 0 \wedge \det [d_{t2}, d_{\gamma}] > 0 \}. \end{cases} \quad (8)$$

3.2. *Flocking Has Formed.* After flocking is formed, α agent has the same velocity with its neighbors, so if it moves toward the direction of obstacle, its next-time velocity should be changed to the tangential direction of the obstacle. The velocity of α agent is given by

$$P_{\alpha 4} = \begin{cases} \frac{\|P_i\|_{\sigma} \cdot d_{t1}}{\|d_{t1}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_{t1}, P_i] < 0 \wedge \det [d_q, P_i] > 0 \} \\ \frac{\|P_i\|_{\sigma} \cdot d_{t2}}{\|d_{t2}\|_{\sigma}}, \\ \{d' \leq r\} \cap \{ \det [d_q, P_i] < 0 \wedge \det [d_{t2}, P_i] > 0 \}. \end{cases} \quad (9)$$

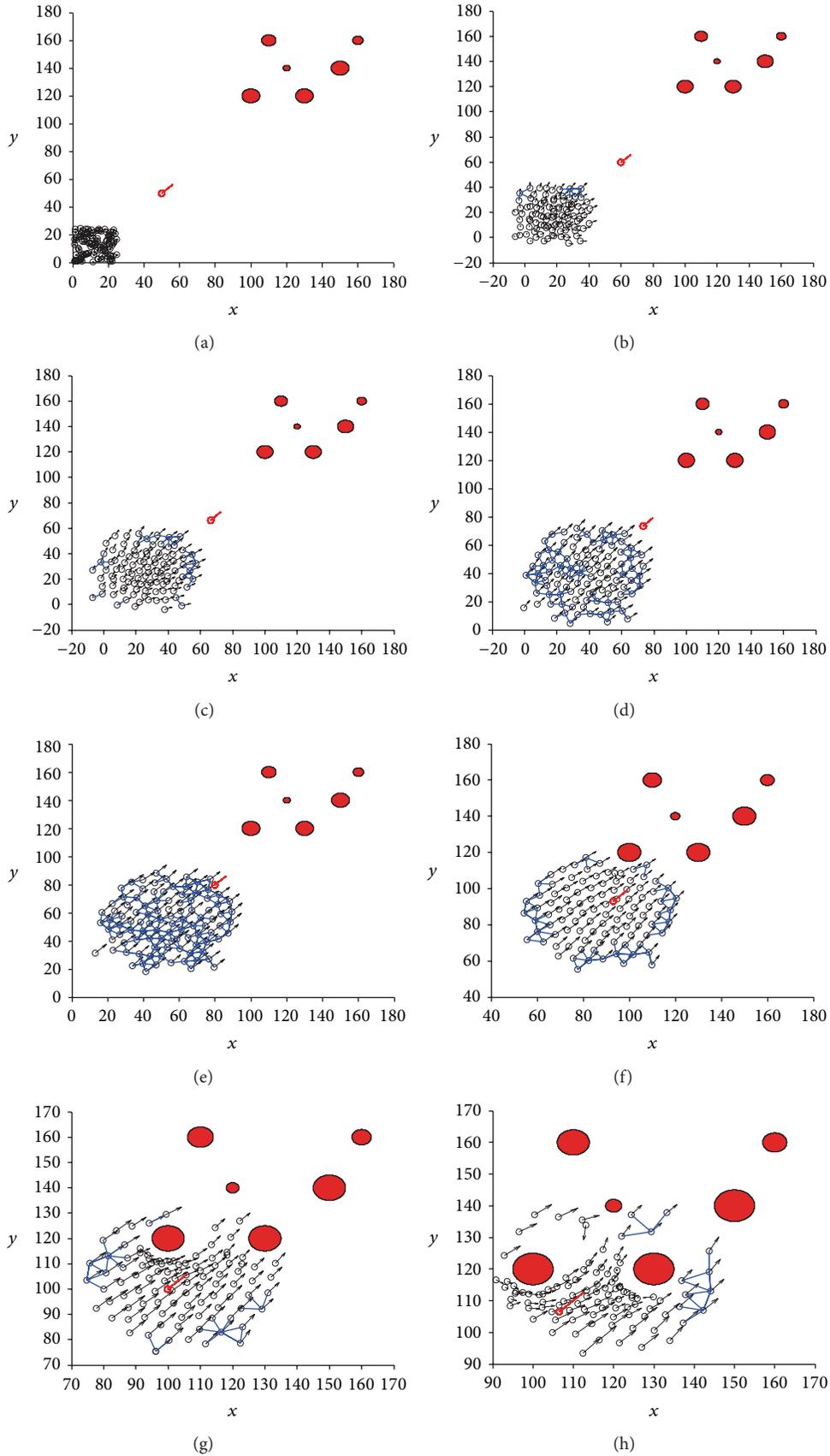


FIGURE 6: Continued.

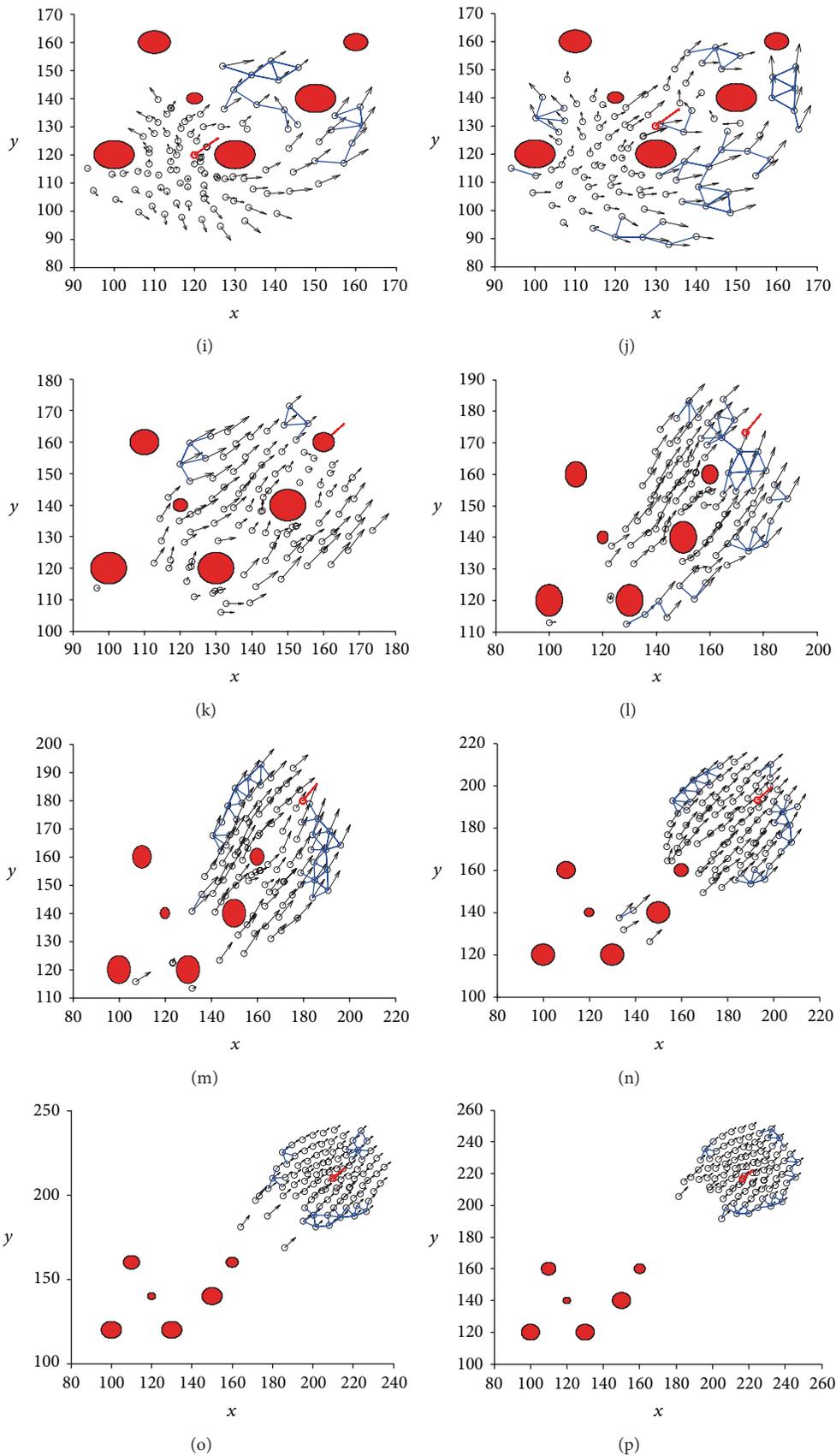


FIGURE 6: Continued.

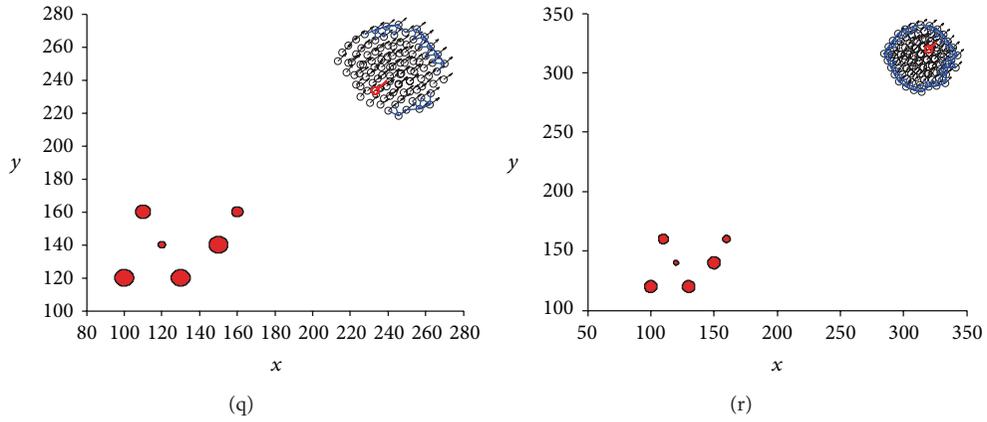


FIGURE 6: Simulation process of the proposed FMOA algorithm with avoiding obstacles.

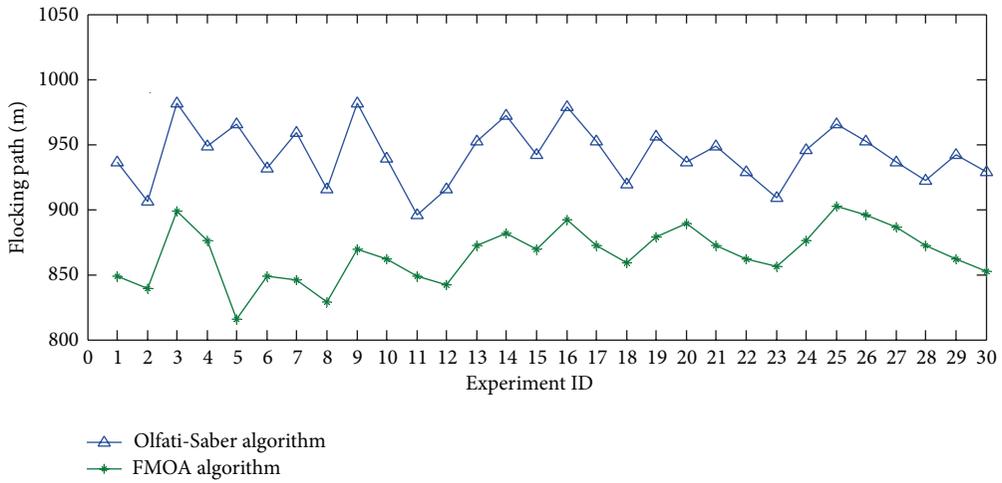


FIGURE 7: Experimental results about flocking path.

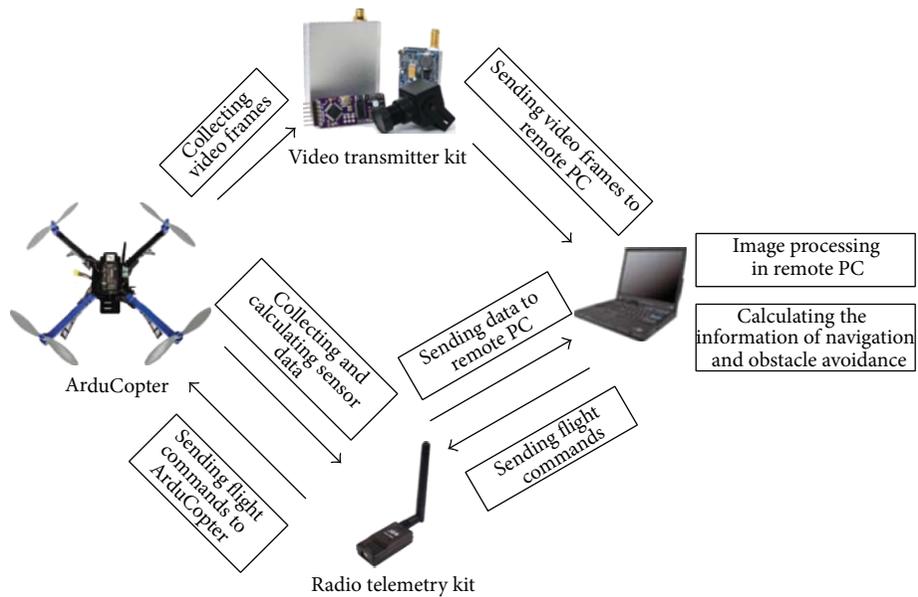


FIGURE 8: Workflow of the whole quadrotor helicopter system.

If it does not move toward the direction of obstacle, then it should just follow its neighbors, because the obstacle will not affect the movement of the α agent at all.

System continuously adjusts the movement of agents according to the following conditions: whether there exist the neighbors with the same velocity and the distance between the agent and its neighbor is d , whether the direction of moving agent is toward the obstacle, and whether target agent is located in the region of two tangent lines formed by agent and obstacle. Then, system will adjust the velocity of agents according to the above different cases, and the whole algorithm can be described as Figure 5 shows.

4. Simulation Experiment Test

In the experiments, Matlab 7.0 is used as the simulation software. System randomly generates 100 agents in the given region as Figure 6(a) shows. The red circle represents the obstacles, and the smaller black circles represent the agents. Besides, the execution parameters of communication distance, lattice length, and safe distance of obstacle are set as 8.8 m, 7 m, and 2 m separately.

For each agent, its arrow describes the direction of the velocity, and the length of the arrow means the magnitude of the velocity. At any certain time, if agents form lattice, then blue lines are added between these agents. These following pictures show the simulation process of FMOA flocking with avoiding obstacles for these agents.

As Figure 6 shows, we can know that agents can effectively avoid obstacles and form flocking toward the direction of destination point. Besides, in order to test the performance of the proposed FMOA algorithm, we tested its flocking path and compared it with Olfati-Saber algorithm. Flocking path length means the total length of paths that all of the agents move from the initial status to the final flocking status after avoiding all obstacles. For the flocking path, the less its value is, the less time the agents spend to reach the destination point with the proposed FMOA algorithm, since these agents have the same magnitude of velocity. In addition, the time complexity of the proposed FMOA algorithm is still $O(n^2)$, which is the same as that of Olfati-Saber algorithm. These experiments are executed with different obstacles and parameter settings. As the experiment results of Figure 7 show, the proposed FMOA algorithm has better efficiency than Olfati-Saber algorithm.

5. The Application of the Proposed FMOA Algorithm

Nowadays, there are many researchers focusing on the research about localization of quadrotor helicopters. Compared with other solutions used to locate the positions of quadrotor helicopters by GPS, laser finder range, and ultra-wideband, this paper innovatively constructed smart indoor environment by deploying WSN nodes.

5.1. Current Techniques about Localization of Quadrotor Helicopters

- (a) *Laser Range Finder*. Abraham Bachrach and others in MIT focus on the research on embedding laser range finder to the hardware platform of quadrotor helicopter, so as to conduct localization of the quadrotor helicopter and explore unknown area. For example, one of their publications [15] focused on the implementation of exploring unknown environment by embedding laser range finder to the quadrotor helicopter.
- (b) *Ultra-Wideband (UWB) Signal*. In order to solve the localization of quadrotor helicopter in indoor environment, Damien B and others in MIT propose to deploy beacons in the test environment. One of their papers [16] targets the algorithm of using some known beacons (at least three beacons are needed) to accurately locate the positions of unknown agents in the indoor environment.
- (c) *Global Position System (GPS)*. Ruijie He and others in MIT have conducted research of embedding GPS devices into the hardware platform of quadrotor helicopter, so as to adopt the quadrotor helicopter to implement object tracking and path planning, and so forth. One of their publications [17] focuses on the design of light-weight quadrotor helicopter and implementation of tracking objects. In fact, GPS can work well outdoors, but in the indoor environment, it usually cannot be used to implement the localization.

5.2. The Construction of Smart Environment. In order to locate the position of the quadrotor helicopter in indoor environment, wireless sensor network (WSN) nodes are deployed in the smart environment. Accordingly, the wireless communication is enabled for each quadrotor helicopter so as to form a WSN node, so quadrotor helicopters can communicate with the WSN nodes deployed in the smart environment.

Based on the analysis of localization error [18], this paper proposes a fast localization algorithm as follows.

Step 1. Calculate the distance between two nodes (including all quadrotor helicopters).

Step 2. Unknown nodes send broadcast packet periodically, and each packet includes the information format $\{ID, T_s, (a, b, c)\}$, where ID is the identification of the unknown node, T_s is the time when the reference node sends the packet, and (a, b, c) is the coordinate value of an unknown node.

Step 3. Reference node i receives the localization packet and then calculates the distance L_i between certain reference node i and the unknown node.

Step 4. For these N packets, according to the C_N^4 set, the system performs the following judgments repeatedly.

(a) Use $\cos a_{i,j} = (L_i^2 + L_j^2 - L_{ij}^2)/2L_iL_j$ to calculate the included angle formed by the unknown node and any two reference nodes (for the set composed of the four ones), where L_i and L_j represent the distances between the unknown node and the two reference nodes, respectively, and L_{ij} is the distance between the two reference nodes. Choose the reference nodes whose six included angles are all within the range $[70.5^\circ - \varphi, 70.5^\circ + \varphi]$ and store them in the set H ; then, $K = K + 1$ (the initial value of K is 0), and return to Step 4; otherwise, go to the next step.

(b) While both of the following conditions are satisfied, then $K = K + 1$. (1) The included angle formed by any two reference nodes (for the set composed of the three reference nodes) and the unknown node (vertex of the included angle) is within the range $[\pi/3 - \theta, \pi/3 + \theta]$, where θ is the angle threshold. (2) The line connected by the fourth reference node and the unknown node and the plane formed by the other three reference nodes will form an included angle. When the included angle is within the range $[\pi/2 - \omega, \pi/2 + \omega]$, where ω is the angle threshold, then return to Step 4; otherwise, go to the next step.

Step 5. Finally, calculate all of the elements in the set H , and obtain their average value H_{avg} . Then, H_{avg} is the estimated position of the unknown node.

5.3. The Proposed Algorithm Used in Formation Flying of Quadrotor Helicopters. In order to apply the proposed algorithm (FMOA), ArduCopters are used as the quadrotor helicopter platform in the constructed smart indoor environment. Real-time data and images are transmitted between ArduCopters and remote PC. Video frames are sent to remote PC by Video Transmitter Kit; accordingly, the process (such as flight attitude, navigation, and obstacles avoidance) is handled in remote PC [19–26], and then flight commands are sent to ArduCopters through Radio Telemetry Kit [27]. The workflow of the whole quadrotor helicopter system is shown as in Figure 8.

(a) *Quadrotor Helicopters in the Same Height.* From the analysis of Figure 9(a), the values of destination points (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) , and (x_6, y_6) are $(0, L)$, $(-L \cdot \tan \beta, 0)$, $(L \cdot \tan \beta, 0)$, $(-2L \cdot \tan \beta, 0)$, $(0, -L)$, and $(2L \cdot \tan \beta, 0)$, respectively. These values are very important for each helicopter flying from the current position to these destination positions, respectively. There are totally 23 WSN nodes placed in the smart environment as shown in Figure 9(b). After obtaining the position set of all of the quadrotor helicopters, their formation flying is controlled by the proposed FMOA algorithm.

The algorithm of quadrotor helicopters in the same height is presented as follows.

Step 1. Quadrotor helicopters all hover in the smart room.

Step 2. Input variables of L and c are sent by remote PC.

Step 3. Choose a front quadrotor helicopter, so as to make sure that all other quadrotor helicopters reach the anticipated formation under the shortest path.

Step 4. Choose other two quadrotor helicopters near the front one as the second layer ones.

Step 5. The other three quadrotor helicopters are processed as the third layer ones.

Step 6. Quad-rotor helicopters obtain the coordinates of their current position by the WSN nodes, and send these coordinates to remote PC.

Step 7. Remote PC sends calculated destination point to each quadrotor helicopter by WSN nodes.

Step 8. Each quadrotor helicopter flies to the final destination position by the proposed FMOA algorithm.

(b) *Quadrotor Helicopters in the Different Height.* From the analysis of Figure 9(a), while the height is H for the three layers of these quadrotor helicopters and the height of the highest quadrotor helicopter is H_0 , then the destination points for (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , (x_4, y_4, z_4) , (x_5, y_5, z_5) , and (x_6, y_6, z_6) are calculated by the input variables H_0 , H , L_1 , and L_2 . This destination information is very important for each helicopter flying from current position to these positions, respectively. After obtaining all of the positions of quadrotor helicopter, their formation control is conducted by the proposed FMOA algorithm (see Figure 10).

The algorithm of quadrotor helicopters in the same height is presented as follows.

Step 1. Quadrotor helicopters all hover in the smart room.

Step 2. Input variables of H_0 , H , L_1 , and L_2 are sent by remote PC.

Step 3. Choose a front quadrotor helicopter with the height H_0 , so as to make sure that all other quadrotor helicopters reach the anticipated formation under the shortest path.

Step 4. Choose other two quadrotor helicopters near the front one as the second layer ones with the height $H_0 - H$.

Step 5. The other three quadrotor helicopters are processed as the third layer ones with the height $H_0 - 2H$.

Step 6. Quad-rotor helicopters obtain the coordinates of their current position by the WSN nodes, and send these coordinates to remote PC.

Step 7. Remote PC sends calculated destination point to each quadrotor helicopter by WSN nodes.

Step 8. Each quadrotor helicopter flies to the final destination position by the proposed FMOA algorithm.

The proposed FMOA algorithm can successfully conduct the formation flying of these quadrotor helicopters. As to the

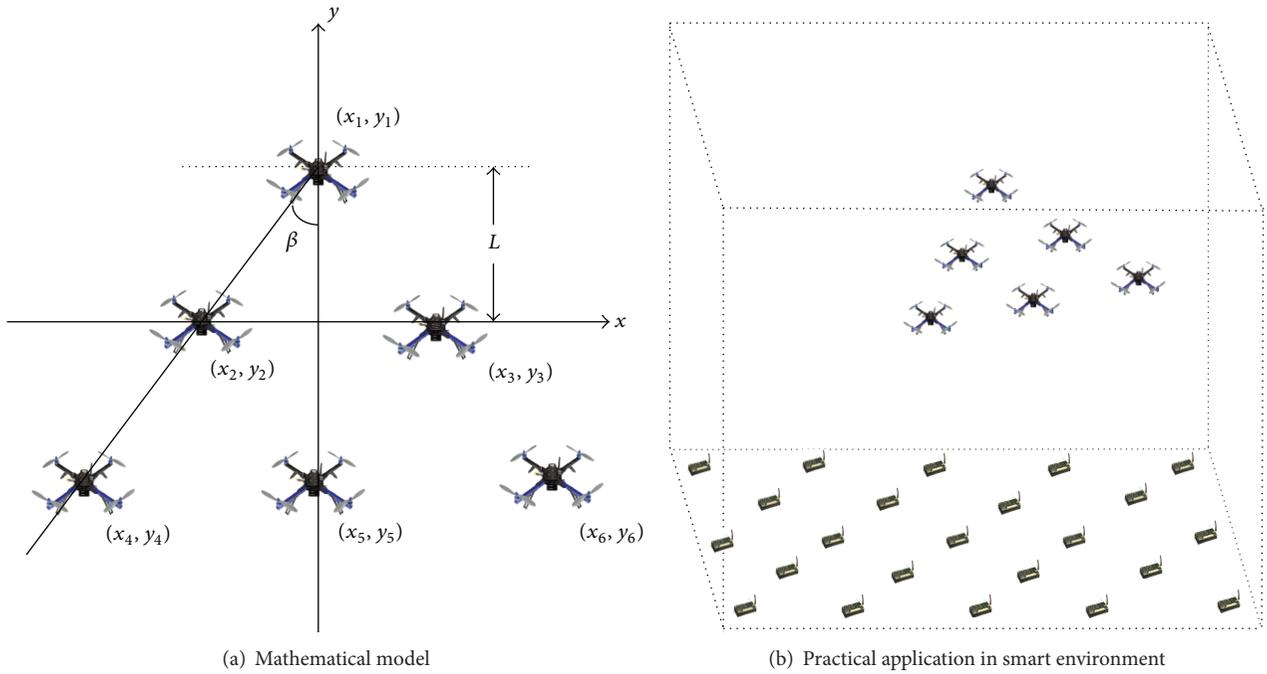


FIGURE 9: The model of formation control for quadrotor helicopters under the condition of different height.

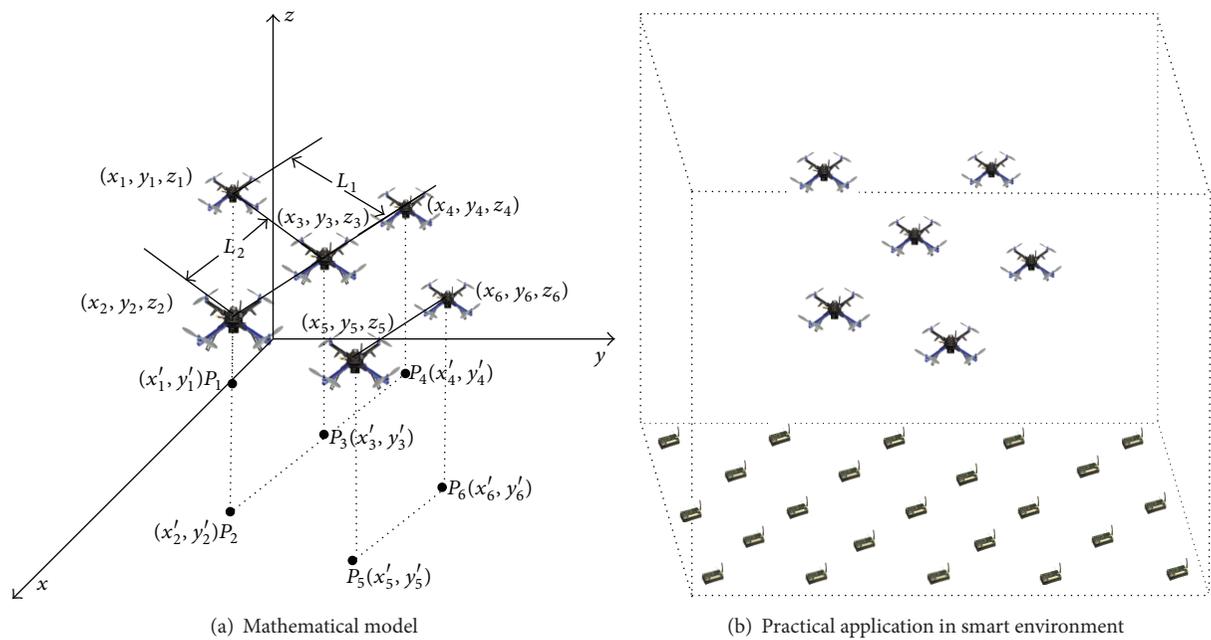


FIGURE 10: The model of formation control for quadrotor helicopters under the condition of different height.

process of obstacle avoidance, the methods of optical flow and image entropy are used for these quadrotor helicopters with single camera. For the future work, we will focus on the research about the believability of the emergent behaviour [28].

6. Conclusion

On the base of the existing Olfati-Saber algorithm, this paper proposed a fast multiagent obstacle avoidance algorithm to make the multiagent system effectively avoid obstacles and

fast form flocking towards the direction of goal target. One of the contributions of this paper is that the direction of agent velocity is comprehensively considered while avoiding obstacles. This paper analyzed the proposed algorithm theoretically under the conditions of whether the flocking is formed, and the subsequent experimental results show that the performance of FMOA is better than Olfati-Saber algorithm in terms of flocking path. In addition, this paper constructed a smart environment by innovatively deploying WSN nodes in indoor environment to locate the positions of all of the quadrotor helicopters; the proposed FMOA algorithm can successfully conduct the formation flying of quadrotor helicopters.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China (no. 61101121), the National High Technology Research and Development Program (863 Program) (no. 2013AA102505), the Key Laboratory Project Funds of Shenyang Ligong University under Grant no. 4771004kfs03, and the Educational Committee of Liaoning Province Science and Technology Research Projects under Grant no. L2013096. The authors thank Jan Vitek, Ales Plsek, and Lei Zhao for their help while the first author studied at Purdue University as a Visiting Scholar from September 2010 to September 2012. The authors also thank the anonymous reviewers for their valuable comments.

References

- [1] [http://en.wikipedia.org/wiki/Flocking_\(behavior\)](http://en.wikipedia.org/wiki/Flocking_(behavior)).
- [2] C. W. Reynolds, "Flocks, herds, and schools: a distributed behavioral model," *ACM SIGGRAPH Computer*, vol. 21, no. 4, pp. 25–34, 1987.
- [3] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Proceedings of the of Game Developers Conference*, pp. 763–782, San Francisco, Calif, USA, 1999.
- [4] C. W. Reynolds, "Interaction with a group of autonomous characters," in *Proceedings of the of Game Developers Conference*, pp. 449–460, San Francisco, Calif, USA, 2000.
- [5] J. Toner and Y. Tu, "Flocks, herds, and schools: a quantitative theory of flocking," *Physical Review E*, vol. 58, no. 4, pp. 4828–4858, 1998.
- [6] N. Shimoyama, K. Sugawara, T. Mizuguchi, Y. Hayakawa, and M. Sano, "Collective motion in a system of motile elements," *Physical Review Letters*, vol. 76, no. 20, pp. 3870–3873, 1996.
- [7] A. Mogilner and L. Edelstein-Keshet, "Spatio-angular order in populations of self-aligning objects: formation of oriented patches," *Physica D: Nonlinear Phenomena*, vol. 89, no. 3–4, pp. 346–367, 1996.
- [8] A. Mogilner and L. Edelstein-Keshet, "A non-local model for a swarm," *Journal of Mathematical Biology*, vol. 38, no. 6, pp. 534–570, 1999.
- [9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [10] J. M. E. Gabbai, *Complexity and the aerospace industry: understanding emergence by relating structure to performance using multi-agent systems [Ph.D. thesis]*, University of Manchester, Manchester, UK, 2005.
- [11] J. Ibáñez, A. F. Gómez-Skarmeta, and J. Blat, "DJ-Boids: emergent collective behavior as multichannel radio station programming," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pp. 248–250, January 2003.
- [12] A. V. Moere, "Time-varying data visualization using information flocking boids," in *Proceedings of the IEEE Symposium on Information Visualization (INFO VIS '04)*, pp. 97–104, October 2004.
- [13] Z. Cui and Z. Shi, "Boid particle swarm optimisation," *International Journal of Innovative Computing and Applications*, vol. 2, no. 2, pp. 78–85, 2009.
- [14] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [15] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in GPS-denied environments using onboard sensing," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 1–8, St Paul, Minn, USA, 2012.
- [16] D. B. Jourdan, J. J. Deyst Jr., M. Z. Win, and N. Roy, "Monte Carlo localization in dense multipath environments using UWB ranging," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICU '05)*, pp. 314–319, September 2005.
- [17] R. He, A. Bachrach, M. Achtelik et al., "On the design and use of a micro air vehicle to track and avoid adversaries," *International Journal of Robotics Research*, vol. 29, no. 5, pp. 529–546, 2010.
- [18] J. Wang, H. Zhao, J. Xu, and Y. Bi, "Webit&NEU: an embedded device for the Internet of things," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 839540, 10 pages, 2014.
- [19] W. Sun, Z. Zhao, and H. Gao, "Saturated adaptive robust control for active suspension systems," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3889–3896, 2013.
- [20] W. Sun, H. Gao, and B. Yao, "Adaptive robust vibration control of full-car active suspensions with electrohydraulic actuators," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2417–2422, 2013.
- [21] W. Sun, H. Gao Sr., and O. Kaynak, "Finite frequency H_∞ control for vehicle active suspension systems," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 416–422, 2011.
- [22] W. Sun, Y. Zhao, J. Li, L. Zhang, and H. Gao, "Active suspension control with frequency band constraints and actuator input delay," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 530–537, 2012.
- [23] W. Sun, H. Gao, and O. Kaynak, "Adaptive backstepping control for active suspension systems with hard constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1072–1079, 2013.
- [24] W. Sun, J. Li, Y. Zhao, and H. Gao, "Vibration control for active seat suspension systems via dynamic output feedback with limited frequency characteristic," *Mechatronics*, vol. 21, no. 1, pp. 250–260, 2011.
- [25] W. Sun, H. Gao, and O. Kaynak, "Vibration isolation for active suspensions with performance constraints and actuator saturation," *IEEE/ASME Transactions on Mechatronics*, 2014.

- [26] W. Sun, H. Pan, Y. Zhang, and H. Gao, "Multi-objective control for uncertain nonlinear active suspension systems," *Mechatronics*, vol. 24, no. 4, pp. 318–327, 2014.
- [27] J. Wang, H. Zhao, Y. Bi, X. Chen, R. Zeng, and Y. Wang, "An improved task scheduling algorithm for intelligent control in tiny mechanical system," *Mathematical Problems in Engineering*, vol. 2014, Article ID 307869, 8 pages, 2014.
- [28] C. Delgado-Mata, J. Ibanez Martinez, S. Bee, R. Ruiz-Rodarte, and R. Aylett, "On the use of virtual animals with artificial fear in virtual environments," *New Generation Computing*, vol. 25, no. 2, pp. 145–169, 2007.