

## Research Article

# Consensus Control of Second-Order Multiagent Systems with Particle Swarm Optimization Algorithm

Xiongfeng Deng <sup>1</sup>, Xiuxia Sun <sup>1</sup>, Ri Liu,<sup>2</sup> and Shuguang Liu<sup>1</sup>

<sup>1</sup>Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University, Xi'an 710051, China

<sup>2</sup>Theory Training Department, Air Force Harbin Flight Academy, Harbin 150001, China

Correspondence should be addressed to Xiuxia Sun; [gcxysxx@126.com](mailto:gcxysxx@126.com)

Received 21 March 2018; Revised 16 August 2018; Accepted 3 September 2018; Published 23 September 2018

Academic Editor: Carlos-Andrés García

Copyright © 2018 Xiongfeng Deng 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.

This paper considers the consensus problem of second-order multiagent systems. Firstly, an improved consensus control protocol is proposed. Then, the convergence of the proposed control protocol is analyzed by applying Lyapunov stability theory. In order to improve the control effect of a given system, the particle swarm optimization (PSO) algorithm is introduced and an improved PSO algorithm is proposed. Additionally, a mapping relationship with agents and the individuals of PSO algorithm is designed. Finally, two simulation examples are provided to illustrate the effectiveness of proposed control protocol and the control effect of PSO algorithm.

## 1. Introduction

The cooperative control problem of multiagent systems has been attracting outstanding attention in the past few years. The main reason is its widespread application in various fields, such as robot systems [1], sensor networks [2], and unmanned aerial vehicle (UAV) systems [3]. The basic problem of the cooperative control of multiagent systems is consensus, which is to design a suitable control law such that the output of all agents can achieve synchronization.

In existing literature, the consensus problem of multiagent systems has been widely researched. In [4], an observer-based distributed output feedback was considered to solve the consensus tracking problem of multiagent systems. In [5, 6], the iterative learning control was applied to deal with the consensus problem of multiagent systems, while the same problem was studied by using linear quadratic regulator (LQR) in [7, 8]. Moreover, the cooperative tracking problem of nonlinear multiagent systems was analyzed in [9, 10], and the consensus control problem of a class of third-order nonlinear multiagent was studied in [11]. Furthermore, distributed control method [12], impulsive control method [13], and adaptive fuzzy output feedback control approach [14] were also applied to solve the consensus problem of multiagent systems.

To improve the control effect, some optimization algorithms, such as genetic algorithm [15], differential evolution algorithm [16], and PSO algorithm [17], have been considered by researchers. As an optimization method, PSO algorithm has the advantages of good search performance, simple implementation, and few adjustment parameters [18]. Due to these advantages, PSO algorithm has gained increasing interest since it was first presented. Currently, there are many results on the applications of the PSO algorithm. For example, to analyze how different algorithmic parameters in a distributed implementation affect the total evaluation and resulting fitness, a distributed PSO algorithm for a class of multirobot systems with the obstacle surroundings was researched in [19]. The nondeterministic navigation problem of UAVs was solved by using the PSO algorithm in [20], where a neighborhood control scheme was designed to eliminate the inherent weakness of PSO algorithm. In [21], the PSO algorithm was applied to deal with the three-dimensional path planning problem of UAVs.

Different from the above research results, a clustering routing algorithm with nonlinear dynamic adaptive PSO algorithm for wireless sensor networks was proposed in [22], where the whole sensor field was positioned via the improved nonlinear dynamic adaptive inertia weight and

evaluation of fitness function. In [23], the PSO algorithm was applied to deal with the space trajectory optimization problem. Additionally, the result that the PSO algorithm was an efficient, reliable, and accurate method for determining optimal space trajectory was demonstrated. Furthermore, the relevant parameters of PSO algorithm are still prespecified with fixed value in some papers [19–21].

Motivated by these facts, we divert our attention to the consensus problem of a class of second-order multiagent systems in this work. The main contributions are outlined in two aspects. On the one hand, an improved control protocol for the given multiagent systems is proposed, and then the convergence is analyzed by the designed Lyapunov function. On the other hand, the PSO algorithm is introduced to improve control effect, where two adaptive laws are designed to adjust the updating laws of PSO algorithm. Also, a new mapping method is applied to solve the mapping problem of the agents' states and the individuals of PSO algorithm. Finally, some simulation examples are provided to illustrate the validity of our results.

The remainder of this paper is organized as follows. In Section 2, some preliminaries are briefly given. In Section 3, the consensus control problem of multiagent systems is analyzed. The fundamental theory of PSO algorithm and the mapping relationship with agents are introduced in Section 4, and simulation examples are provided in Section 5. Finally, some conclusions are drawn in Section 6.

## 2. Preliminaries

Let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$  denote an undirected graph which consists of  $n$  nodes, where  $\mathcal{V} = \{v_1, \dots, v_N\}$  is the set of vertices and  $\mathcal{E} = \{(v_i, v_j) : v_i, v_j \in \mathcal{V}\} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges. The weighted adjacency matrix is denoted by  $\mathcal{A} = [a_{ij}] \in R^{n \times n}$ , where  $a_{ij} > 0$  if and only if  $i \neq j$  and  $a_{ij} = 0$  otherwise. A path between  $v_i$  and  $v_j$  is a sequence of distinct vertices  $v_i, v_{i_1}, \dots, v_l, v_j$ , where  $(v_i, v_{i_1}) \in \mathcal{E}$ . An undirected graph is connected if there exists a path from any node  $i$  to any other node  $j$ . Multiagent systems consist of  $n$  agents. Each agent can be described as a node and the exchange information among agents can be expressed as an edge with definite weight in a generalized graph. Therefore, the problem of multiagent systems can be solved through the graph theory.

For the sake of discussion below, some definitions and lemmas are given as follows.

**Definition 1.** The consensus of a multiagent systems is said to be achieved if, for any initial condition, there exist  $\lim_{t \rightarrow \infty} x_i(t) = x_j(t)$  and  $\lim_{t \rightarrow \infty} v_i(t) = v_j(t)$  for  $i, j \in 1, \dots, n$  and  $i \neq j$ .

**Lemma 2** (see [24]). Let  $\sigma = [\sigma_1, \dots, \sigma_m]^T \in R^m$ ,  $\rho = [\rho_1, \dots, \rho_m]^T \in R^m$ , and  $C = [c_{ij}] \in R^{m \times m}$ ; if  $C$  is a symmetric matrix, that is,  $c_{ij} = c_{ji}$ , then we have

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m c_{ij} (\sigma_i - \sigma_j) \varphi(\rho_i - \rho_j) = \sum_{i=1}^m \sum_{j=1}^m c_{ij} \sigma_i \varphi(\rho_i - \rho_j) \quad (1)$$

where  $\varphi(\cdot)$  represents an odd function.

## 3. Consensus Analysis of Multiagent Systems

Considering a general second-order multiagent systems with undirected graph, the dynamics of the  $i$ th agent are described as

$$\begin{aligned} \dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= u_i(t) \end{aligned} \quad (2)$$

where  $x_i(t) \in R$ ,  $v_i(t) \in R$  and  $u_i(t) \in R$  are the position, velocity, and control input of agent  $i$ , respectively;  $i \in \{1, \dots, n\}$  and  $n$  is the number of agents.

In this paper, a suitable control protocol  $u_i(t)$  needs to be chosen for the multiagent systems (2) to achieve consensus. Inspired by the results of [25, 26], an improved control protocol is given as

$$\begin{aligned} u_i(t) &= -\alpha \sum_{j=1}^n a_{ij} (x_i(t) - x_j(t))^3 \\ &\quad - \beta \sum_{j=1}^n a_{ij} (v_i(t) - v_j(t))^3 \end{aligned} \quad (3)$$

and the general form of (3) is described as

$$\begin{aligned} u_i(t) &= -\sum_{j=1}^n a_{ij} [\alpha \varphi(x_i(t) - x_j(t)) + \beta \varphi(v_i(t) - v_j(t))] \end{aligned} \quad (4)$$

where  $\alpha$  and  $\beta$  are positive constants;  $a_{ij}$  is the  $(i, j)$  element of  $\mathcal{A}$  and  $\varphi(\cdot)$  is an odd function.

For the convenience of analysis, the time variable  $t$  will be ignored from here on.

According to (4), (2) is rewritten as

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= -\sum_{j=1}^n a_{ij} [\alpha \varphi(x_i - x_j) + \beta \varphi(v_i - v_j)] \end{aligned} \quad (5)$$

Hence, we have the following Theorem.

**Theorem 3.** Consider the multiagent systems (2) with the consensus protocol (4), and let the exchange information topology graph of agents be undirected and connected; then the consensus can be achieved, i.e.,  $x_i(t) = x_j(t)$  and  $v_i(t) = v_j(t)$  as  $t \rightarrow \infty$  for  $i, j \in 1, \dots, n$  and  $i \neq j$ .

*Proof.* Let  $x_{ij} = x_i - x_j$ , and then have

$$\begin{aligned} \dot{x}_{ij} &= \dot{x}_i - \dot{x}_j = v_i - v_j \\ \dot{v}_i &= -\sum_{j=1}^n a_{ij} [\alpha \varphi(x_{ij}) + \beta \varphi(v_i - v_j)] \end{aligned} \quad (6)$$

Design the Lyapunov function candidate

$$V = V_1 + V_2 \quad (7)$$

with

$$V_1 = \frac{1}{4} \alpha \sum_{i=1}^n \sum_{j=1}^n a_{ij} \varphi^2(x_{ij}) \quad (8)$$

$$V_2 = \frac{1}{2} \sum_{i=1}^n v_i^2 \quad (9)$$

From (8) and (9), it is obvious that  $V_1 \geq 0$  and  $V_2 \geq 0$ ; then  $V \geq 0$  can be obtained. Taking the derivative of function  $V_i$  ( $i = 1, 2$ ), we get

$$\dot{V}_1 = \frac{1}{2} \alpha \sum_{i=1}^n \sum_{j=1}^n a_{ij} (v_i - v_j) \varphi(x_{ij}) \quad (10)$$

$$\dot{V}_2 = -\alpha \sum_{i=1}^n \sum_{j=1}^n a_{ij} v_i \varphi(x_{ij}) - \beta \sum_{i=1}^n \sum_{j=1}^n a_{ij} v_i \varphi(v_i - v_j) \quad (11)$$

Considering Lemma 2, we have

$$\frac{1}{2} \alpha \sum_{i=1}^n \sum_{j=1}^n a_{ij} (v_i - v_j) \varphi(x_{ij}) = \alpha \sum_{i=1}^n \sum_{j=1}^n a_{ij} v_i \varphi(x_{ij}) \quad (12)$$

$$\begin{aligned} & \beta \sum_{i=1}^n \sum_{j=1}^n a_{ij} v_i \varphi(v_i - v_j) \\ &= \frac{1}{2} \beta \sum_{i=1}^n \sum_{j=1}^n a_{ij} (v_i - v_j) \varphi(v_i - v_j) \end{aligned} \quad (13)$$

From (12) and (13), one gets

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \beta (v_i - v_j) \varphi(v_i - v_j) \quad (14)$$

Due to the assumption that  $\mathcal{G}$  is undirected and connected and  $\varphi(\cdot)$  is an odd function, it is obtained that  $x\varphi(x) > 0$ . Hence, the following result is held:

$$\dot{V} = -\frac{1}{2} \beta \sum_{i=1}^n \sum_{j=1}^n a_{ij} [(v_i - v_j)^2 \varphi^2(x_{ij})] \quad (15)$$

$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \beta (v_i - v_j) \varphi(v_i - v_j) \leq 0$$

Let  $\Gamma = \{(x_{ij}, v_i) \mid \dot{V} = 0\}$ . From (14),  $\dot{V} = 0$  hints that  $v_i = v_j$  ( $i \neq j$ ). Furthermore,  $\dot{v}_i = \dot{v}_j$  ( $i \neq j$ ) can be deduced. Therefore, there exists  $\xi \in R$  such that  $v \in \text{span}(\mathbf{1} \times \xi)$ , where  $v = [v_1, \dots, v_n]^T$ ,  $\mathbf{1} = [1, \dots, 1]^T$ , and  $\text{span}(\cdot)$  represents the vector space of  $v$ .

Combining  $v_i = v_j$  ( $i \neq j$ ) and (7), then we have

$$\dot{v}_i = -\sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) \quad (16)$$

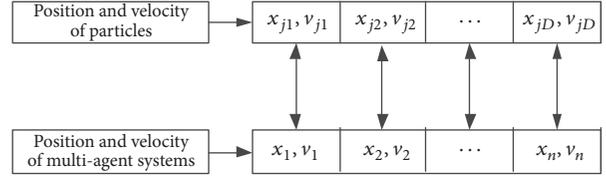


FIGURE 1: Mapping scheme.

In addition,

$$\begin{aligned} (\mathbf{1} \times \xi)^T \dot{v} &= (\mathbf{1} \times \xi)^T \left[ -\sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) \right] \\ &= -\xi \sum_{i=1}^n \sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) \end{aligned} \quad (17)$$

In view of the exchange information topology graph  $\mathcal{G}$  is undirected, and there exists  $\varphi(x_{ij}) = -\varphi(x_{ji})$ ; we have

$$\begin{aligned} & \xi \sum_{i=1}^n \sum_{j=1}^n a_{ij} \alpha \varphi(x_{ji}) + \xi \sum_{i=1}^n \sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) \\ &= 2\xi \sum_{i=1}^n \sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) = 0 \end{aligned} \quad (18)$$

From (16) and (18),  $(\mathbf{1} \times \xi)^T \dot{v} = 0$  can be obtained, indicating that  $\dot{v}$  is orthogonal to  $\mathbf{1} \times \xi$ . Hence,  $\dot{v} = 0$  is deduced, and then we have

$$-\sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) = 0 \quad (19)$$

and

$$-\sum_{i=1}^n x_i \sum_{j=1}^n a_{ij} \alpha \varphi(x_{ij}) = 0 \quad (20)$$

which hints that

$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \alpha (x_i - x_j) \varphi(x_{ij}) = 0 \quad (21)$$

From (21), we have  $a_{ij} \alpha (x_i - x_j) \varphi(x_{ij}) = 0$ ,  $\forall i \neq j$ ; then  $x_i = x_j$ ,  $\forall i \neq j$  is obtained. Thereby, the results that  $x_i = x_j$  and  $v_i = v_j$  for  $i, j \in 1, \dots, n$  and  $i \neq j$  are obtained. The proof is completed.  $\square$

In order to improve the control effect, the PSO algorithm will be introduced in the following discussion.

## 4. PSO Algorithm and the Mapping Relationship with Agents

**4.1. PSO Algorithm.** As an optimization method, PSO algorithm seeks the optimum solution of an optimization problem via selecting some particles. The characteristics of each

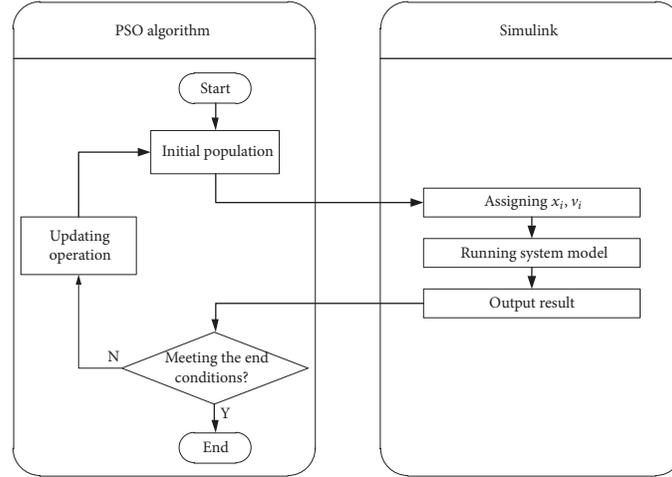


FIGURE 2: The flowchart of optimization.

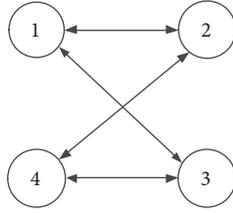


FIGURE 3: Exchange information topology graph.

particle include position, velocity, and fitness value. The fitness value is calculated by the fitness function, where the merit of the particle is described by the fitness value. In the solution space, each particle's position and velocity are updated by the individual extremum  $P_{best}$  and the population extremum  $G_{best}$ , where  $P_{best}$  and  $G_{best}$  are updated by comparing the fitness value of the new particle.

Considering a  $D$  dimension search space, let  $X = (X_1, \dots, X_m)$  be a population with  $m$  particles. The position and velocity of the  $j$ th particle are defined as  $X_j = (x_{j1}, \dots, x_{jD})^T$  and  $V_j = (v_{j1}, \dots, v_{jD})^T$ , respectively. The extremum of individual and population are defined as  $P_{best,j} = (P_{best,j1}, \dots, P_{best,jD})^T$  and  $G_{best} = (G_{best,1}, \dots, G_{best,D})^T$ , respectively. The velocity and position updating laws of the  $j$ th particle at the  $(k+1)$ th iteration are given as

$$V_{jd}^{k+1} = \omega V_{jd}^k + \kappa_1 \gamma_1' (P_{best,jd}^k - X_{jd}^k) + \kappa_2 \gamma_2' (G_{best,d}^k - X_{jd}^k) \quad (22)$$

$$X_{jd}^{k+1} = X_{jd}^k + V_{jd}^{k+1} \quad (23)$$

where  $X_{jd}^k$  and  $V_{jd}^k$  are the  $d$ th position and velocity component of the  $j$ th individual, respectively;  $\omega \in (0, 1]$  is inertia weight;  $\kappa_i$  ( $i = 1, 2$ )  $\geq 0$  are acceleration factors;  $\gamma_i$  ( $i = 1, 2$ )  $\in [0, 1]$  are random numbers between 0 and 1;  $P_{best,jd}^k$  is the  $d$ th best position component of the  $j$ th individual and  $G_{best,d}^k$  is the  $d$ th best position of the entire population.

In this paper, the updating laws (22) and (23) are improved and the results are given as

$$V_{jd}^{k+1} = \omega V_{jd}^k + \kappa_1 \gamma_1' (P_{best,jd}^k - X_{jd}^k) + \kappa_2 \gamma_2' (G_{best,d}^k - X_{jd}^k) \quad (24)$$

$$X_{jd}^{k+1} = X_{jd}^k - \Delta V_{jd}^{k+1} \operatorname{sgn}(\Delta V_{jd}^{k+1}) \quad (25)$$

where

$$\gamma_1' = \frac{\|P_{best,j}^k\|}{\sqrt{\|P_{best,j}^k\|^2 + \|G_{best}^k\|^2}} \quad (26)$$

$$\gamma_2' = \frac{\|G_{best}^k\|}{\sqrt{\|P_{best,j}^k\|^2 + \|G_{best}^k\|^2}} \quad (27)$$

$$\Delta V_{jd}^{k+1} = (V_{jd}^k - V_{jd}^{k+1}) \quad (28)$$

*Remark 4.* From (26) and (27), note that the attributes of  $\gamma_1'$  and  $\gamma_2'$  are the same as those of  $\gamma_1$  and  $\gamma_2$ . However, compared with  $\gamma_1$  and  $\gamma_2$ ,  $\gamma_1'$  and  $\gamma_2'$  not only overcome the uncertainty of random variation but also have the capacity of adaptive adjustment. In addition, due to  $\Delta V$  being introduced in the updating law (25), the smoothness of the convergence result can be guaranteed in the end of the algorithm.

As an important part of PSO algorithm, the fitness function in this paper is designed as

$$f_j^k = \sum_{d=1}^D (\|x_{jd}^k - x_{jd}^{k-1}\| + \|v_{jd}^k - v_{jd}^{k-1}\|) \quad (29)$$

where  $D = n$  is the dimension of each particle;  $f_j^k$  is the fitness value of the  $j$ th particle in the  $k$ th iteration;  $x_{jd}^0$  and  $v_{jd}^0$  are the initial states of agents and  $k \in (1, \dots, K)$  represents the  $k$ th iteration.

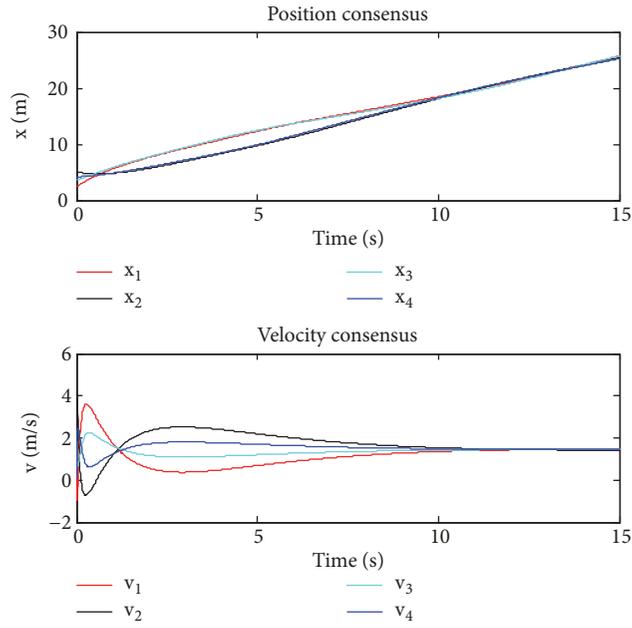


FIGURE 4: States consensus with control protocol (3).

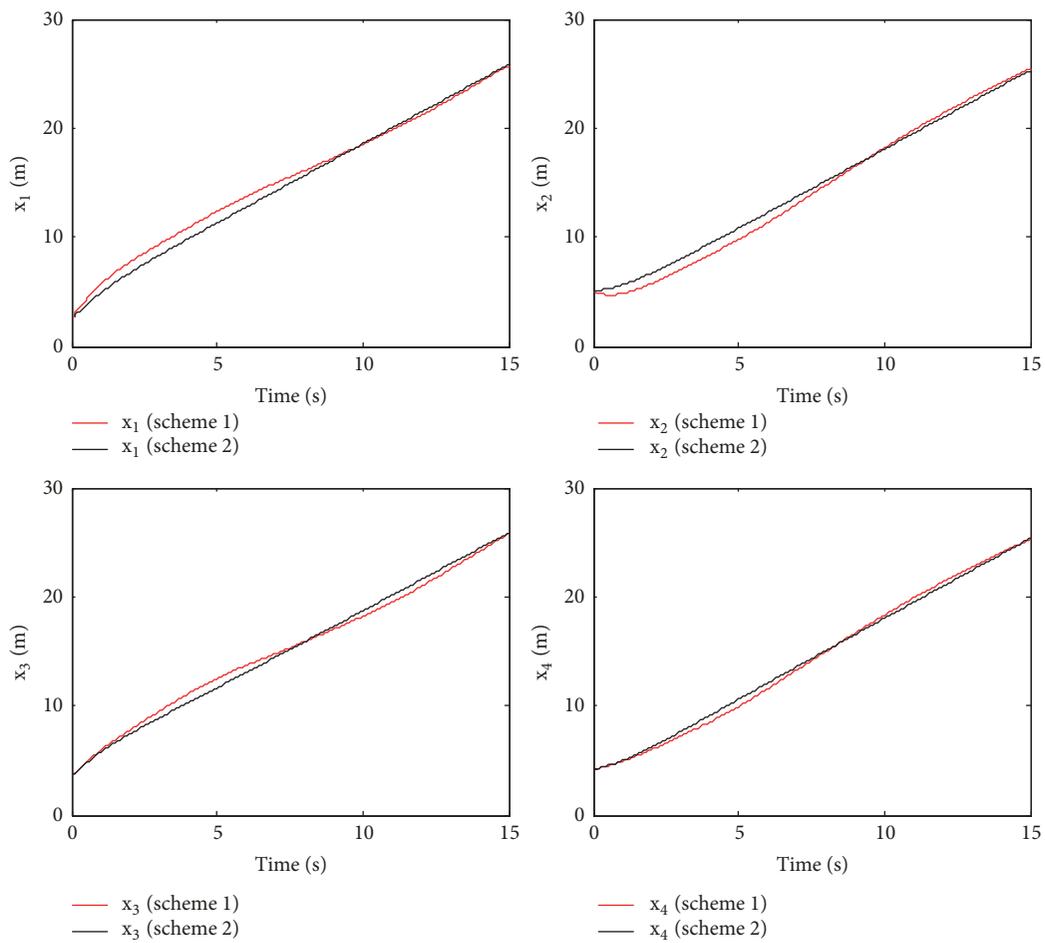


FIGURE 5: Position comparison results with standard PSO algorithm and control protocol (3).

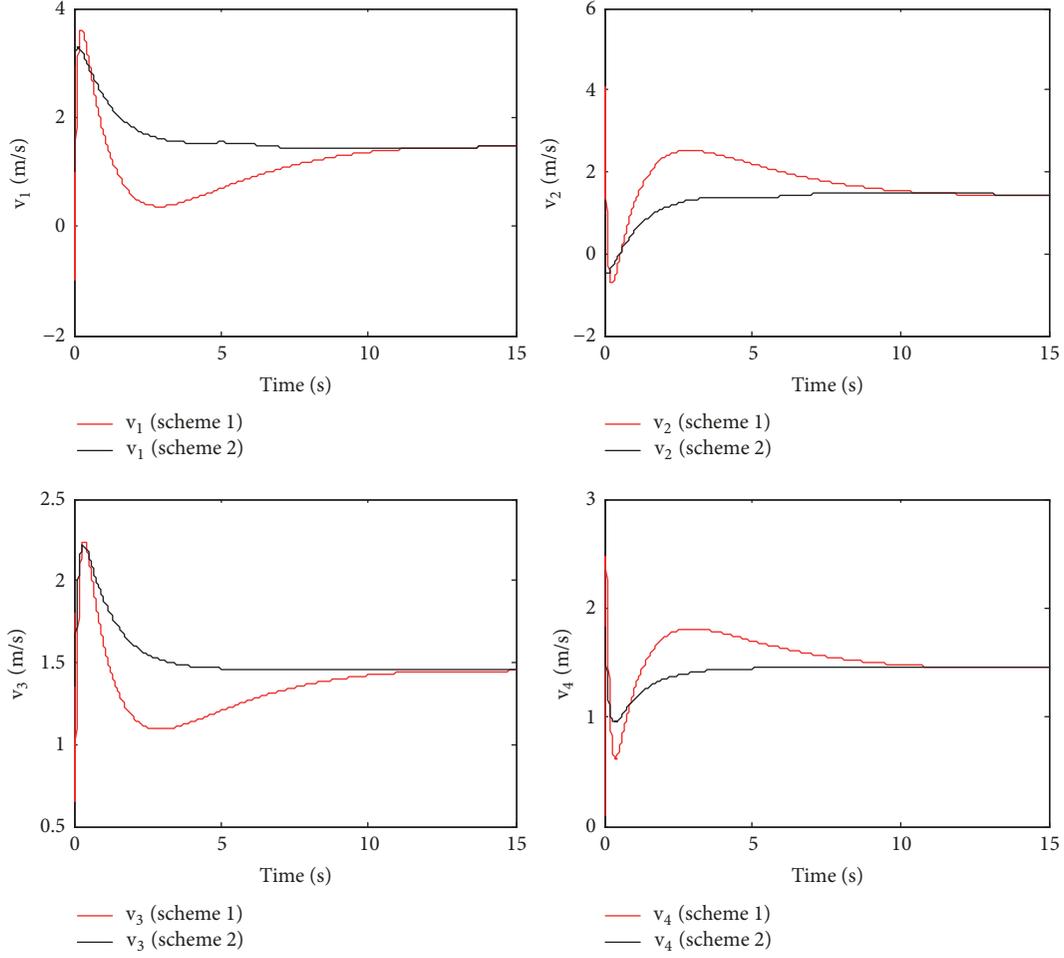


FIGURE 6: Velocity comparison results with PSO algorithm and control protocol (3).

Let  $t_{\max}$  and  $\Delta t$  be the simulation time and simulation step, respectively. Hence, the maximum number of iterations  $K$  is defined as

$$K = \frac{t_{\max}}{\Delta t} \quad (30)$$

**4.2. Mapping Relationship Design.** For a multiagent systems with  $n$  agents, each agent is considered as a particle. Let  $x_i$  and  $v_i$  be the position and velocity of the  $i$ th agent, respectively. Then the sets of position and velocity are expressed as  $X_{MAS} = (x_1, \dots, x_n)$  and  $V_{MAS} = (v_1, \dots, v_n)$ , respectively. In the PSO algorithm, let  $X_{PSO,j} = (x_{j1}, \dots, x_{jD})$  and  $V_{PSO,j} = (v_{j1}, \dots, v_{jD})$  be the position and velocity of the  $j$ th particle, respectively. Consequently, the mapping relationship between particles and agents is shown in Figure 1.

The operation process of PSO algorithm is given as follows:

**Step 1.** Initialize the population and the position and velocity of all agents.

**Step 2.** Assign the position and velocity  $x_i$  and  $v_i$  to each particle.

**Step 3.** Run system model and calculate the fitness function (29).

**Step 4.** Judge the termination conditions and update  $X_{jd}^k$  and  $V_{jd}^k$  according to (24) and (25).

**Step 5.** Repeat Steps 2 and 4.

The flowchart is shown in Figure 2.

## 5. Simulation Analysis

Considering the second-order multiagent systems (2) with four agents, the exchange information topology graph  $\mathcal{G}$  is shown in Figure 3.

The weighted adjacency matrix  $\mathcal{A}$  is

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (31)$$

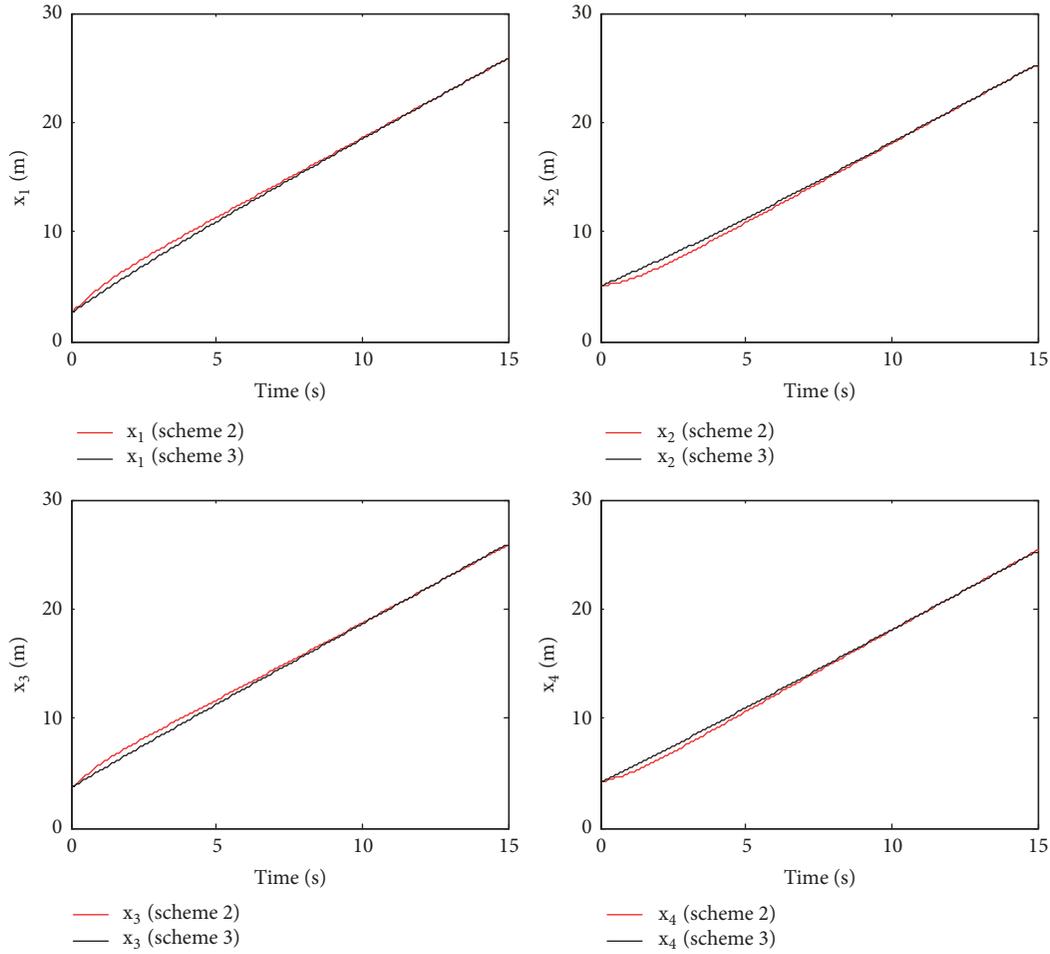


FIGURE 7: Position comparison results with standard PSO algorithm and improved PSO algorithm.

TABLE I: List of parameters.

multi-agent systems	PSO algorithm
$n = 4$	$m = 100$
$\alpha = 2.5, \beta = 2.0$	$\omega = 0.2$
$t_{\max} = 15s, \Delta t = 0.01s$	$\kappa_1 = 0.7, \kappa_2 = 1.5$
$x^0 = [2.5, 5, 3.6, 4]^T (m)$	$\lambda_1 = 0.4, \lambda_2 = 0.5$
$v^0 = [3.0, 3.7, 1.8, 1.4]^T (m/s)$	$K = 1500$

The parameter settings are shown in Table 1.

In addition, the updating laws (26), (27), and (28) are used for (24) and (25), respectively. According to the number of agents, it can be deduced that the dimension  $D = 8$ . To illustrate the validity of proposed control protocol (3) and the control effect of PSO algorithm, the following simulation examples are provided.

*Example 5* (comparison the control protocol (3) and the standard PSO algorithm). In this example, we consider the consensus of multiagent systems (2) with control protocol (3)

(scheme 1) and the standard PSO algorithm (22) and (23) (scheme 2); the simulation results are shown in Figures 4–6.

The consensus of position and velocity with the control protocol (3) are shown in Figure 4. It can be found that the control effects are not good from Figure 4. Applying the standard PSO algorithm, the comparison results are drawn in Figures 5 and 6. Compared with control protocol (3), the consensus achieving time and the overshoot have been greatly improved by using the PSO algorithm.

*Example 6* (comparison the standard PSO algorithm and the improved PSO algorithm). In this example, the consensus of the multiagent systems (2) with the standard PSO algorithm (22) and (23) (scheme 2) and the improved PSO algorithm (24) and (25) (scheme 3) is considered; the simulation results are shown in Figures 7 and 8.

From Figures 7 and 8, it can be seen that the output of all agents have been further improved. Compared with the standard PSO algorithm, the improved PSO algorithm not only reduces the consensus achieving time and overshoot but also makes the curves smoother. In general, in the ability to improve the control effect of the multiagent systems, the

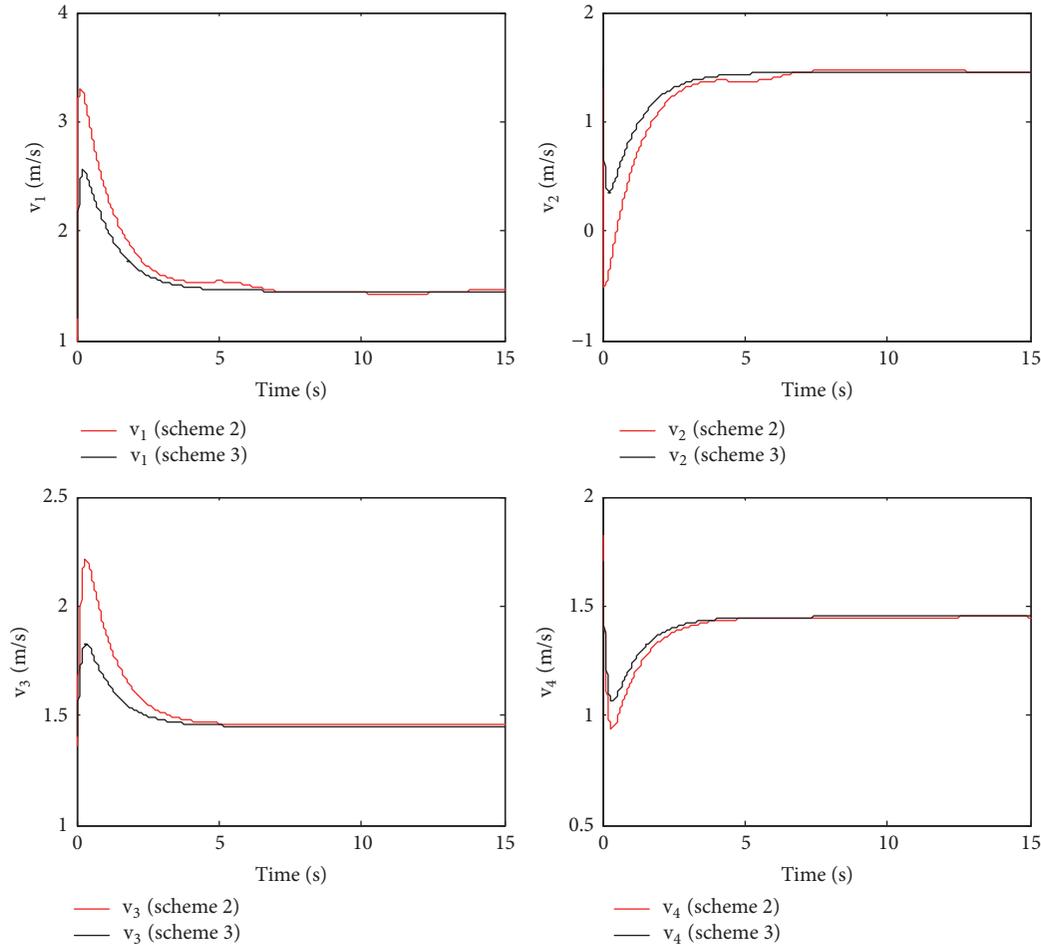


FIGURE 8: Velocity comparison results with standard PSO algorithm and improved PSO algorithm.

improved PSO algorithm proposed in this paper is superior to the other two proposed approaches.

## 6. Conclusion

In this paper, we discussed the consensus problem of a class of second-order multiagent systems with the PSO algorithm. An improved consensus protocol was designed, and then the convergence was analyzed by the Lyapunov stability theory. The PSO algorithm was introduced to improve the control effect of the multiagent systems. And then an improved PSO algorithm was proposed. In addition, we designed a mapping relationship with the agents and the individuals of PSO algorithm. Finally, the simulation results showed that the consensus with the proposed control protocol can be achieved. Also, the control effect can be improved by using PSO algorithm; and the better control effect can be obtained by utilizing the improved PSO algorithm.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the Aeronautical Science Foundation of China (Grant no. 20155896025).

## References

- [1] H. Chen, "Robust stabilization for a class of dynamic feedback uncertain nonholonomic mobile robots with input saturation," *International Journal of Control, Automation, and Systems*, vol. 12, no. 6, pp. 1216–1224, 2014.
- [2] W. W. Yu, G. R. Chen, Z. D. Wang, and W. Yang, "Distributed consensus filtering in sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 6, pp. 1568–1577, 2009.
- [3] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

- [4] Y. L. Jiang, J. C. Liu, and S. Q. Wang, "Consensus tracking algorithm via observer-based distributed output feedback for multi-agent systems under switching topology," *Circuits, Systems and Signal Processing*, vol. 33, no. 10, pp. 3037–3052, 2014.
- [5] X. F. Deng, X. X. Sun, and R. Liu, "Quantized consensus control for second-order nonlinear multi-agent systems with sliding mode iterative learning approach," *International Journal of Aeronautical and Space Sciences*, vol. 19, no. 2, pp. 518–533, 2018.
- [6] X. F. Deng, X. X. Sun, R. Liu, and S. G. Liu, "Consensus control of time-varying delayed multiagent systems with high-order iterative learning control," *International Journal of Aerospace Engineering*, vol. 2018, pp. 1–12, 2018.
- [7] D. H. Nguyen, "A sub-optimal consensus design for multi-agent systems based on hierarchical LQR," *Automatica*, vol. 55, pp. 88–94, 2015.
- [8] J. Y. Ma, Y. S. Zheng, and L. Wang, "LQR-based optimal topology of leader-following consensus," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 17, pp. 3404–3421, 2015.
- [9] J. Q. Hu, J. D. Cao, K. Yuan, and T. Hayat, "Cooperative tracking for nonlinear multi-agent systems with hybrid time-delayed protocol," *Neurocomputing*, vol. 171, pp. 171–178, 2016.
- [10] R. Abdulghafor, S. Turaev, A. Zeki, and I. Al-Shaikhli, "Reach a nonlinear consensus for MAS via doubly stochastic quadratic operators," *International Journal of Control*, vol. 91, no. 6, pp. 1431–1459, 2018.
- [11] Y. M. Xin, Y. X. Li, X. Huang, and Z. S. Cheng, "Consensus of third-order nonlinear multi-agent systems," *Neurocomputing*, vol. 159, no. 1, pp. 84–89, 2015.
- [12] G. Q. Hu, "Robust consensus tracking of a class of second-order multi-agent dynamic systems," *Systems Control Letters*, vol. 61, no. 1, pp. 134–142, 2012.
- [13] D. D. Li, J. Ma, H. M. Zhu, and M. Sun, "The consensus of multi-agent systems with uncertainties and randomly occurring nonlinearities via impulsive control," *International Journal of Control, Automation, and Systems*, vol. 14, no. 4, pp. 1005–1011, 2016.
- [14] W. Wang, D. Wang, Z. H. Peng, and H. Wang, "Cooperative adaptive fuzzy output feedback control for synchronization of nonlinear multi-agent systems in the presence of input saturation," *Asian Journal of Control*, vol. 18, no. 2, pp. 619–630, 2016.
- [15] X. F. Zhang, Y. S. Ding, and Z. X. Liang, "Adaptive co-evolution model for multi-agent system," *Journal of Donghua University*, vol. 27, no. 2, pp. 285–288, 2010.
- [16] D. K. He, G. Y. Gao, F. L. Wang, and Y. Liu, "Multi-agent differential evolution algorithm," *Control and Decision*, vol. 26, no. 7, pp. 961–966, 2011.
- [17] D. S. Shu, Z. X. Huang, J. Y. Li, and X. Y. Zuo, "Application of multi-agent particle swarm algorithm in distribution network reconfiguration," *Journal of Electronics*, vol. 25, no. 6, pp. 1179–1185, 2016.
- [18] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, Nagoya, Japan, October 1995.
- [19] E. D. Mario and A. Martinoli, "Distributed particle swarm optimization for limited-time adaptation with real robots," *Robotica*, vol. 32, no. 2, pp. 193–208, 2014.
- [20] A. Banks, J. Vincent, and K. Phalp, "Particle swarm guidance system for autonomous unmanned aerial vehicles in an air defence role," *Journal of Navigation*, vol. 61, no. 1, pp. 9–29, 2008.
- [21] J. L. Foo, J. S. Knutzon, V. Kalivarapu, J. H. Oliver, and E. Wimer, "Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 4, pp. 271–290, 2009.
- [22] D. A. Li, H. L. Hao, G. L. Ji, and J. M. Zhao, "An adaptive clustering algorithm based on improved particle swarm optimisation in wireless sensor networks," *International Journal of High Performance Computing and Networking*, vol. 8, no. 4, pp. 370–380, 2015.
- [23] M. Pontani and B. A. Conway, "Particle swarm optimization applied to space trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1429–1441, 2010.
- [24] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1503–1509, 2008.
- [25] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10–11, pp. 1002–1033, 2007.
- [26] W. W. Yu, G. R. Chen, and M. Cao, "Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems," *Automatica*, vol. 46, no. 6, pp. 1089–1095, 2010.

