Maintaining Complex Formations and Avoiding Obstacles for Multi-Agents

Yali Wang¹, Youqian Feng¹, Zhonghai Yin¹ and Cheng Zhou^{1,*}

Abstract: This paper researches some problems in complex formation for multi-agents, in which two matrices are proposed to record the formation. The pattern matrix is used to describe the pattern of the formation; meanwhile, the location matrix is used to record the location of each agent. Thus, all desired positions of each agent will be obtained by geometrical relationship on the basis of two matrices above. In addition a self-adaptation flocking algorithm is proposed to control all agents to form a desired formation through the method of formation control when far away from obstacles; otherwise, agents will freely fly to pass through the area of obstacles. In the simulation, three scenarios are designed to verify the effectiveness of our method. The results show that our method also can be applied in three dimensions. All agents will form a stable formation and keep the same velocity at last.

Keywords: Multi-agents, formation control, self-adaptation, distributed, velocities consensus.

1 Introduction

Flocking is a form of collective behavior of a large number of interacting agents with a common group objective to complete certain tasks [Tanner, Jadbabaie and Pappas (2007); Meng, Jia, Du et al. (2014)]. Formation control is one of the most actively studied topics within the realm of multi-agent systems and their coordination control has attracted large amounts of research interest. Based on different sensing capability and the topology of multi-agents [Oh, Park and Ahn (2015)], formation control problems can be divided into three parts: position-based control, displacement-based control, and distance-based control. According to the fundamental ideas in control methods, formation control can be divided into four parts: leader-following, behavior-based, virtual structure, and consensus [Benzerrouk, Adouane and Martinet (2014); Lü, Chen and Chen (2016); Ge, Han, Ding et al. (2018); Essghaier, Beji, El Kamel et al. (2011); Xie and Wang (2007)].

In 2006, Olfati-Saber [Olfati-Saber (2006)] proposed a theoretical framework for design and analysis of distributed flocking algorithms. It primarily discussed potentials to control agents for free flocking and obstacle avoidance. However, this method cannot control agents to form a desired formation. Su et al. [Su, Wang and Lin (2009)] proposed

¹ Department of Basic Courses, Air Force Engineering University, Xi'an, 710051, China.

^{*} Corresponding Author: Cheng Zhou. Email: zhou417461659@163.com.

a method to track a virtual leader based on the artificial potential field method. By the influence of the potential field between neighboring robots, the organization of the swarm is stable and robust. Obstacle avoidance is driven by a repulsive potential field. The method of potential functions with leader-follower has mainly been researched by many scholars [Vilca, Adouane and Mezouar (2016); Yan, Guan, Luo et al. (2017); Shi, Zhao and Zhang (2019); Park and Yoo (2015); Mas and Kitts (2010); Brandao, Barbosa, Mendoza et al. (2014)] for formation control and obstacles avoidance. For example, an application of the multi-layer control scheme to guide a formation of three unmanned aerial vehicles (UAV) in trajectory tracking missions was presented [Mas and Kitts (2010)]. The centralized and distributed architecture of the lead-follower method was presented under the cluster perspective [Brandao, Barbosa, Mendoza et al. (2014)]. The key idea of this combination method is to use leader-follower to maintain a desired formation, and to use potential function to avoid obstacles. The differences between those papers are the various application backgrounds or special potential functions for obstacles avoidance. However, the velocity of leader will change dramatically when near by obstacles, which has a severe impact on formation control. Further, agents are probably trapped into a region with a large number of obstacles for an inappropriate method of obstacle avoidance.

This paper aims to address the following issues. Firstly, the graph of formation is simple in most papers. A new model of formation needs to be constructed when the number of agents increases or decreases. This paper aims to provide a unitized method to record the formation graph by using the method of pattern matrix. Secondly, many papers often omit the obstacle environment or pass around obstacles with a fixed formation. However, the jitter of the leader will impose a significant effect on formation control when agents are close to obstacles. More seriously, agents will be trapped in the region with many obstacles. Hence, a self-adaptation distributed flocking algorithm is introduced to provide a strategy for multi-agents formation control and obstacle avoidance.

The paper is structured as follows. Section 2 introduces some preliminaries and problem description. Section 3 discusses the models of matrices to describe a complex formation. Furthermore, the control protocol of our method is proposed in Section 4. A simulation of multi-agent system is presented in Section 5, and the simulation result is analyzed. The last section gives brief conclusions and discusses future works.

2 Preliminaries

2.1 Graph theory

A graph G is a pair (v,ε) that consists of a set of vertices $v = \{1,2,\dots,n\}$ and edges $\varepsilon \subseteq \{(i,j): i, j \in v, j \neq i\}$. The graph G can be undirected or directed. Thus, it is undirected if there exists $(i,j) \subseteq \varepsilon \Leftrightarrow (j,i) \subseteq \varepsilon$; otherwise, it is directed. Consider a group of n agents in a graph G, and the adjacency matrix is defined to describe the position relation for agent i and its neighbors, specified as

$$a_{ij} = \begin{cases} 1 \Leftrightarrow (i,j) \subseteq \varepsilon \\ 0 \Leftrightarrow (i,j) \not\subset \varepsilon \end{cases}$$
(1)

The elements in the adjacent matrix are other than just 0 or 1. Furthermore, the set of neighbors of agent i is defined as

$$N_i = \{j \in v : a_{ij} = 1\} = \{j \in v : (i, j) \in \mathcal{E}\}.$$
(2)

Meanwhile, all agents have the ability to sense their neighbors. Let r be the communication range, and r' be the recognition range for obstacles. Let p_i and p_j be the positions of agents i and j, respectively. The adjacency matrix and neighbors can be rewritten as

$$N_i^{\alpha} = \{ j \in v : a_{ij} = 1 \} = \{ j \in v : \| \boldsymbol{p}_j - \boldsymbol{p}_i \| \le r \},$$
(3)

$$a_{ij} = \begin{cases} 1, & \left\| \boldsymbol{p}_j - \boldsymbol{p}_i \right\| \le r \\ 0, & \left\| \boldsymbol{p}_j - \boldsymbol{p}_i \right\| > r \end{cases}$$
(4)

where $\|\cdot\|$ is the Euclidean norm. It means that adjacency matrix $a_{ij}=0$ if the distance between agent *i* and *j* is outside the scope of communication range *r*; otherwise $a_{ij}=1$. Meanwhile, the neighbors of obstacles of agent *i* can be described as

$$N_i^{\beta} = \{k : || \, \hat{q}_{ik} - q_i \, || < r' \, , \, k \in O_k\} \,, \tag{5}$$

where \hat{q}_{ik} is the location of each obstacle, and O_k is the set of obstacles. It means that agent *i* is the neighbor of obstacle if their distance is within the obstacle recognition range r'. It is important that agent would not need to know all the information from other agents or environment, i.e., agents just need to communicate with their neighbors, regardless of other agents outside the scope of communication range r. If all agents have the same interaction range, the proximity net *G* becomes an undirected graph. This paper contains some assumptions as follows.

- (1) All agents have the same spherical sensing range.
- (2) The delay of communication is omitted.

2.2 Problem description

n agents will form a desired formation, in which agent 1 is called "leader", and others numbered by $2,3,\dots,n$ are "followers". The leader has an independent motion; while followers will be influenced by the leader and other followers [Olfati-Saber (2006)]. The kinematic model of agents is defined as

$$\begin{cases} \dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i \end{cases}$$
(6)

where $p_i = (x_i, y_i, z_i)^T$ and $v_i = (\dot{x}_i, \dot{y}_i, \dot{z}_i)^T$ are the vectors of position and velocity of agent *i*, and $u_i = (u_x, u_y, u_z)^T$ is the control protocol of agent *i*. For the leader, its dynamics can be described as

$$\begin{cases} \dot{p}_{1} = v_{1}(t) \\ \dot{v}_{1} = f(v_{1}(t), t) \end{cases}$$
(7)

where $p_i = (x_1, y_1, z_1)^T$ is the position of agent *i*, $v_1 = (\dot{x}_1, \dot{y}_1, \dot{z}_1)^T$ is the velocity, and $f(v_1(t), t)$ is the input vector of dynamics of leader. A potential function which is influenced by the neighbors of all agents, the control protocol u_i can be specified as

$$\boldsymbol{u}_{i} = -\boldsymbol{k}_{p} \sum_{i}^{n} \nabla_{p_{i}} \boldsymbol{V}_{i} - \boldsymbol{k}_{v} \sum_{j \in N_{i}} (\boldsymbol{v}_{i} - \boldsymbol{v}_{j}) .$$

$$\tag{8}$$

where $k_p > 0, k_v > 0$ are the parameters of position and velocity. V_i is the total potential of agent *i*, which is denoted by

$$V_i = \sum_{j \in N_i} V_{ij} \left(\left\| \boldsymbol{p}_{ij} \right\| \right), \tag{9}$$

 V_{ij} is a differentiable, nonnegative, function of the distance $\|\mathbf{p}_{ij}\|$ between agents *i* and *j*. Thus, the problem of formation can be described as

$$\lim_{t \to \infty} (\boldsymbol{p}_i - \boldsymbol{p}_i^d) = 0$$

$$\lim_{t \to \infty} (\boldsymbol{v}_i - \boldsymbol{v}_1) = 0, i \in n$$
(10)

where p_i^d is the desired position of agent *i* which is relative to the leader. v_1 is the velocity of the leader. The whole formation control problem can be regarded that all agents will reach to their desired positions and have the same velocity with leader agent over time.

2.3 Control scheme

In Fig. 1(a), the formation control of multi-agent system consists of a guidance law as an outer loop and a flight control as an inner loop, i.e., outer loop makes decisions and inner loop executes actuators. Hence, a closed control loop will be constructed by using the strategy of outer and inner loop for sampling the state with time T. It should be known that this paper mainly discusses outer loop for making decisions of multi-agents. Here, as shown in Fig. 1(b), the control processes of multi-agent system are detailed as follows. Firstly, the initial parameters of each agent (position, velocity, acceleration and so on), global object point, and obstacles will be given to construct the Environment as an initial state of multi-agents. For outer loop, there are two parts of Preprocessing and Control protocol, specified as follows. (1) The preprocessing of formation control will be executed. Thus, the desired positions of each agent will be obtained based on the pattern matrix F and conversion function T(N) in location matrix. (2) Three modules of Interaction, Navigation, and Obstacle Avoiding are used to accomplish control protocol, in which two strategies of free-flying and self-adaptation flocking algorithm are proposed to obtain control signal to make agents form a desired formation from pattern matrix F. Further, in the inner loop, new states of multi-agents will be obtained in actuators among individual agents for sampling the state $(\mathbf{p}, \mathbf{v}, \mathbf{u})$ with T.

880



Figure 1: The control scheme of multi-agent system

It is important to know that this control protocol is a distributed strategy. Each agent will execute their individual control signal based on the interactions with their surrounding agents, obstacles, and desired positions from different modules. The control protocol will continue to control agents to move until agents reach to target states (p_i^d, v_1) of each agent. Further, as new mission arrives, multi-agents will move again through control scheme above.

3 The models of matrices

3.1 Pattern matrix

Pattern matrix is the fundamental matrix to record the graphical information for formation. Furthermore, the pattern matrix can be formally represented by

	f_{11}	f_{12}	•••	f_{1n}		
F _	f_{21}	f_{22}	•••	f_{2n}		
1' -	:	÷	·	÷	,	'
	f_{n1}	f_{n2}		f_{nn}		

 $f_{ij} = 1$ is used to present the presence of agent; otherwise $f_{ij} = 0$. Any complex formation can be presented by an $n \times n$ pattern matrices. Furthermore, consider that a pattern matrix contains n agents, in which all agents are numbered from left to right and sequentially incremented for every row.

	0	0	0	0	0	0	0	0		0	0	0	\bigcirc	\mathcal{O}_{2}	0	0	0
Г	0	0	0	1	1	0	0	0		0	0	$O_{\mathbf{r}}$	• -	ź	0	0	0
	0	0	1	1	1	0	0	0		0	\mathcal{O}_{6}	-	÷	Ň	0	0	0
	0	1	1	1	1	0	0	0		0	ě-	*	ب ھُ-	×	0	0	0
$F_{1} =$	0	0	0	1	1	0	0	0		0	0	0	•	*	0	0	0
	0	0	0	1	1	0	0	0		0	0	0	•)	0	0	0
	0	0	0	1	1	0	0	0		0	0	0	•		0	0	0
	0	0	0	1	1	0	0	0		0	0	0	16 ●	$\overset{1}{>}$	0	0	0
(a) pattern matrices of ("1") (b) Numbering												g of	F_1				

Figure 2: An example of pattern matrix

In an 8×8 pattern matrices, a number '1' formation contains 17 agents. The pattern matrices of F_1 ("1") is denoted by Fig. 2(a). Further, the number of F_1 is shown in Fig. 2(b). Meanwhile, an appropriate dimension of pattern matrix should be considered to avoid the explosion for the complexity of the matrix. However, this problem is not the key issue for this paper, and it will not be further studied here.

3.2 Location matrix

Location matrix is introduced to record the location information in a formation. Firstly, a conversion function T(N) is defined to convert the number of agents in pattern matrix to the information in location matrix, specified as,

$$T(N) = U(r,c), \tag{12}$$

where *N* is the number of each agent in pattern matrices. *r* and *c* are the row and column in location matrix, respectively. Meanwhile, a notation $[r,c]_N$ is utilized to simplify the conversion function T(N), i.e., the agent of number *N* is located at rows *r* and columns *c*. For example, in Fig. 2(b), the agent of number 5 can be denoted by $[2,3]_s$. It means that agent 5 is located at row 2 and column 3 in location matrix.

3.3 Calculation of desired position

It is clear that the desired position $p_j^d = (x_j^d, y_j^d, z_j^d)$ of agent *j* can be computed by the actual position of leader $p_1 = (x_1, y_1, z)$ through geometrical relations based on pattern matrix *F* and conversion function T(N). Additionally, a relative coordinate system x'oy' (the altitude of *z* is constant) is shown in Fig. 3.



Figure 3: Calculation of desired position in relative and global coordinate system

It is constructed by pattern matrices where y' axis has the same direction with the velocity of leader and x' is the vertical direction of y'. Let the distances of rows and columns of agents be $d_{x'}$ and $d_{y'}$ in pattern matrices. In x'oy', the relative desired position $p'_j = (x'_j, y'_j, z)$ of agent j can be calculated by $p_1 = (x_1, y_1, z)$, specified as,

$$\begin{bmatrix} x'_{j} \\ y'_{j} \\ z'_{j} \end{bmatrix} = \begin{bmatrix} x_{1} \\ y_{1} \\ z_{1} \end{bmatrix} + \begin{bmatrix} c_{j} - c_{1} \\ r_{1} - r_{j} \\ 0 \end{bmatrix} \begin{bmatrix} d_{x'} \\ d_{x'} \\ 0 \end{bmatrix}^{T}.$$
(13)

where c_j and r_j are the rows and columns in location matrices based on conversion function T(N). Furthermore, the desired position $p_j^d = (x_j^d, y_j^d, z_j^d)$ of agent *j* can be calculated by

$$\begin{cases} x_{j}^{d} = x_{1} + (y_{j}' - y_{1})\cos\theta + (x_{j}' - x_{1})\sin\theta \\ y_{j}^{d} = y_{1} + (y_{j}' - y_{1})\sin\theta - (x_{j}' - x_{1})\cos\theta \\ z_{j}^{d} = z_{1} \end{cases}$$
(14)

where θ is the angle of velocity v_1 with x axis in global coordinate system xoy. x'_j and y'_j are the relative desired position.

4 Control protocol

Olfati-Saber [Olfati-Saber (2006)] proposed a distributed flocking algorithm, where a systematic method is provided for the construction of collective potentials for flocking. Meanwhile, according to the different constructions in collective potentials, those agents in flocking can be divided into three kinds. Firstly, some agents are composed of flocking members, called α -agent. Secondly, some are all kinds of obstacles, called β -agent. Thirdly, the others are virtual agents associated with respective object point, called γ -agent. A comparison of the distributed flocking algorithm and our method is shown in Fig. 4.



Figure 4: The control protocol of collective potentials

In Fig. 4(a), all α -agents will arrive at the equilibrium point in the effect of potentials. β -agent will generate repulsion to make all agents get away from obstacles. Meanwhile, the γ -agent will generate navigation information to guide all agents to the object point. However, the distributed flocking algorithm is not capable of completing a complicated task for forming a specific formation from scatter to order.

In Fig. 4(b), the method of this paper is specified as follows. Firstly, α -agent has the same function with Fig. 4(a). It can be used to remain a certain distance and avoid collision with other agents. Secondly, β -agent has the same function with Fig. 4(a), in which the repulsion will make all agents away from obstacles. However, the velocity direction of the leader will jitter continuously when near to obstacles. It is clear that the jitter will impose a significant effect on formation control. Consider that agents will prefer to avoid obstacles when reaching to the region of obstacles. Thus, a free-flying strategy is used to control all agents to pass through the region of obstacles, while, a formation control strategy is used to control all agents to form a desired formation when all agents are far away from obstacles. Thirdly, γ -agent is different from Fig. 4(a). It contains two functions: leader and followers. The γ -agents to move to the objective collectively. The rest γ -agents of followers are located at their desired location. Thus, all agents will construct an intended formation under the whole virtual agents.

4.1 Formation control for the absence of obstacles

The control protocol u_i of *i* th agent can be specified as,

$$\boldsymbol{u}_i = \boldsymbol{u}_i^{\alpha} + \boldsymbol{u}_i^{\gamma}, \qquad (15)$$

where u_i^{α} is the control feedback of *i* th agent from surrounding α -agents. And u_i^{γ} is the navigation information of *i* th agent from its corresponding γ -agent. As the Euclidean norm ||z|| is not differentiable at z = 0. Thus, a new σ norm is defined to ensure the differentiability of potential at any point. $||z||_{\sigma}$ specifies as

$$||z||_{\sigma} = \frac{1}{\varepsilon} [\sqrt{1 + \varepsilon ||z||^2} - 1], \qquad (16)$$

where ε is a positive parameter. Thus, for any point, the gradients $\nabla || z ||_{\sigma}$ is defined as

Maintaining Complex Formations and Avoiding Obstacles

$$\nabla \parallel z \parallel_{\sigma} = \frac{z}{\sqrt{1+\varepsilon} \parallel z \parallel^2} = \frac{z}{1+\varepsilon} \parallel z \parallel_{\sigma}, \tag{17}$$

Hence, it is clear that $||z||_{\sigma}$ is differentiable at z = 0. Meanwhile, a term of action functions $\phi_{\alpha}(\cdot)$ is designed to ensure the smoothness about the potential of attraction and repulsion. The forces will disappear for outside the range of communication r and the forces are minimum for a desired distance $d \cdot \phi_{\alpha}(\cdot)$ is specified as,

$$\phi_{\alpha}(z) = \rho_{h}(z / r_{\alpha} \cdot \phi(z - d_{\alpha})), \ \phi(z) = \frac{1}{2} [(a_{0} + b_{0}) \cdot \sigma_{1}(z + c_{0}) + (a_{0} - b_{0})].$$
(18)

where $d_{\alpha} = ||d||_{\sigma}$, and $\sigma_1(z) = z/\sqrt{1+||z||^2}$. Thus, $\phi(z)$ is an anomalous sigmoidal function. The parameters a_0, b_0, c_0 satisfy $0 < a_0 \le b_0$. Meanwhile, let $c_0 = |a_0 - b_0|/\sqrt{4a_0b_0}$ ensure $\phi(0) = 0$. $\rho_h(z)$ is a piecewise function, specified as

$$\rho_{h}(z) = \begin{cases} 1 & z \in [0,h) \\ \frac{1}{2} [1 + \cos(\pi \cdot \frac{z-h}{1-h})] & z \in [h,1] \\ 0 & others \end{cases}$$
(19)

where $h \in (0,1)$. It is clear that $\rho_h(z)$ is a smoothness function, and there exists $\rho'_h(z) = 0$ for any $z \in [1,\infty)$. Further, a smooth potential function V(p) is defined as

$$V(p) = \frac{1}{2} \sum_{i} \sum_{j \neq i} \psi_{\alpha} \left(\left\| p_{j} - p_{i} \right\|_{\sigma} \right),$$

$$(20)$$

where $\psi_{\alpha}(z)$ is a smooth attractive/repulsive potential, specified as

$$\psi_{\alpha}(z) = \int_{d_{\alpha}}^{z} \phi_{\alpha}(s) ds .$$
(21)

According to the formulas above, u_i^{α} is computed as

$$\boldsymbol{u}_{i}^{\alpha} = c^{\alpha} \sum_{j \in N_{i}^{\alpha}} \phi_{\alpha}(||\boldsymbol{p}_{j} - \boldsymbol{p}_{i}||_{\sigma}) \cdot \boldsymbol{n}_{ij} \cdot \boldsymbol{a}_{ij}(\boldsymbol{p})$$

$$(22)$$

 c^{α} is a positive parameter. \mathbf{n}_{ij} is the gradients function about relative distance between α -agents, specified as $\mathbf{n}_{ij} = \nabla || \mathbf{p}_j - \mathbf{p}_i ||_{\sigma}$. \mathbf{p}_i and \mathbf{p}_j are the location vector. $a_{ij}(\mathbf{p})$ is the adjacent matrix from Eq. (5), i.e., in the sensing range, $a_{ij}(\mathbf{p}) = 1$; others, $a_{ij}(\mathbf{p}) = 0$. As agent 1 is the leader, the other agents 2,..., N are followers. The γ -agent of leader is located at global object point. The leader will guide all followers to move to the desired destination. Similarly, the γ -agent of followers is located at their desired location, which will control all followers to form a desired formation. The navigation output u_i^{γ} can be described as

$$\boldsymbol{u}_i^{\boldsymbol{\gamma}} = \boldsymbol{c}_1^{\boldsymbol{\gamma}} \cdot \boldsymbol{p}_i^{\boldsymbol{\gamma}} + \boldsymbol{c}_2^{\boldsymbol{\gamma}} \cdot \boldsymbol{v}_i^{\boldsymbol{\gamma}} \,. \tag{23}$$

where the former can be considered as an attractive force from agent *i*. And the latter is the damping force from velocities matching. c_1^{γ} and c_2^{γ} are positive parameters. As the

885

navigation outputs are different from leader and followers, u_i^{γ} can be divided into two kinds: leader and followers. Leader will receive the global object point and velocity. p_1^{γ} and v_1^{γ} are obtained by

$$\begin{cases} p_{1}^{\gamma} = p_{\gamma} - p_{1} \\ v_{1}^{\gamma} = v_{\gamma} - v_{1} \end{cases} \quad i = 1,$$
(24)

where p_{γ} and v_{γ} are global object point and velocity vectors, i.e., agent will reach to p_{γ} with velocity v_{γ} . The object points of follower are their respective desired locations, and desired velocities of followers are the velocity of leader. The desired locations and velocities of followers will be changed for the motion of leader. p_i^{γ} and v_i^{γ} are obtained by

$$\begin{cases} \boldsymbol{p}_{i}^{\gamma} = \boldsymbol{p}_{i}^{d} - \boldsymbol{p}_{i} \\ \boldsymbol{v}_{i}^{\gamma} = \boldsymbol{v}_{1} - \boldsymbol{v}_{i} \end{cases} \quad i \neq 1.$$

$$(25)$$

4.2 Free-flying strategy for the presence of obstacles

The obstacles avoidance should take precedence over formation control for the presence of obstacles. The free-flying strategy will be used to control all agents to pass through the region of obstacles. The control output u_i^o of obstacles can be described as

$$\boldsymbol{u}_{i}^{o} = \boldsymbol{u}_{i}^{\alpha} + \boldsymbol{u}_{i}^{\beta} + \boldsymbol{u}_{i}^{\gamma_{o}}.$$
(26)

where $\boldsymbol{u}_i^{\alpha}$ is the same with formula (22). It is used to maintain a certain distance between every agent. \boldsymbol{u}_i^{β} is the control output which is used to maintain a safe distance d'between agents and obstacles. The control output of neighbors of obstacles for agent *i* can be computed as

$$\boldsymbol{u}_{i}^{\beta} = c_{1}^{\beta} \cdot \sum_{k \in N_{i}^{\alpha}} \varphi_{\beta}(\| \hat{\boldsymbol{p}}_{ik} - \boldsymbol{p}_{i} \|_{\sigma}) \cdot \hat{\boldsymbol{n}}_{ik} + c_{2}^{\beta} \cdot \sum_{j \in N_{i}^{\alpha}} b_{ik}(\boldsymbol{p}) \cdot (\hat{\boldsymbol{v}}_{ik} - \hat{\boldsymbol{v}}_{i})$$
(27)

Similarly, a term of action functions of $\varphi_{\beta}(z) = \rho_{h}(z/d_{\beta}) \cdot (\sigma_{1}(z-d_{\beta})-1)$ and $d_{\beta} = ||d'||_{\sigma}$ is used to keep the forces disappeared outside the range of obstacles sensing and the forces are minimum for a safe distance $d' \cdot \hat{n}_{ik}$ is the gradients function about relative distance between agents and obstacles, specified as $\hat{n}_{ik} = \nabla || \hat{p}_{ik} - p_{i} |||_{\sigma} \cdot c_{1}^{\beta}$ and c_{2}^{β} are positive parameters. $b_{ik}(p)$ is the adjacent matrix between agents and obstacles, i.e., in the range of obstacles sensing, $b_{ik}(p) = 1$; others, $b_{ik}(p) = 0$. Agents will receive the information of global object point and velocity to pass through the region of obstacles preferentially. The navigation feedback $u_{i}^{\gamma_{o}}$ of each agent generated by global γ -agents can be computed as

$$\boldsymbol{u}_{i}^{\gamma_{o}} = \boldsymbol{c}_{1}^{\gamma} \cdot (\boldsymbol{p}_{\gamma} - \boldsymbol{p}_{i}) + \boldsymbol{c}_{2}^{\gamma} \cdot (\boldsymbol{v}_{\gamma} - \boldsymbol{v}_{i}) .$$

$$(28)$$

where p_i and v_i are the actual location and velocity vector. p_{γ} and v_{γ} are location and velocity vector of global object point.

5 Stability analysis

In order to analyze the stability of multi-agents, a coordinate system is established from the origin of leader. The position vector of agent *i* can be described by \mathbf{r}_i , and its velocity is $\dot{\mathbf{r}}_i = \mathbf{v}_i - \mathbf{v}_1$. A moving frame can be obtained on the basis of Eq. (3) in coordinate system above, denoted by,

$$\frac{d}{dt} \begin{bmatrix} \mathbf{r}_i \\ \dot{\mathbf{r}}_i \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{r}}_i \\ \mathbf{u}_i - \dot{\mathbf{v}}_1 \end{bmatrix}.$$
(29)

For the relative position vector $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, the control output \mathbf{u}_i of agent *i* can be described as

$$\boldsymbol{u}_{i} = -\boldsymbol{c}_{1}^{\alpha} \cdot \sum_{j \in N_{i}^{\alpha}} \nabla \boldsymbol{V} - \boldsymbol{c}_{1}^{\gamma} \cdot \boldsymbol{r}_{i} - \boldsymbol{c}_{2}^{\gamma} \cdot \boldsymbol{\dot{r}}_{i}$$
(30)

Generally, the Lyapunov function is selected as

$$\Phi = \frac{1}{2} \sum_{j \in N_i^{\alpha}} \left(\dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_i + 2c_1^{\alpha} \cdot \sum_{j \in N_i^{\alpha}} \mathbf{V} + c_2^{\gamma} \cdot \mathbf{r}_i^2 \right).$$
(31)

Take the time derivative to Eq. (31), we have,

$$\dot{\Phi}(t) = \sum_{j \in N_i^{\alpha}} \left(\dot{\boldsymbol{r}}_i \cdot (\boldsymbol{u}_i + \boldsymbol{c}_1^{\alpha} \cdot \sum_{j \in N_i^{\alpha}} \nabla \boldsymbol{V} + \boldsymbol{c}_1^{\gamma} \cdot \boldsymbol{r}_i \right) = \sum_{j \in N_i^{\alpha}} \left(\dot{\boldsymbol{r}}_i \cdot (-\boldsymbol{c}_2^{\gamma} \cdot \dot{\boldsymbol{r}}_i) = -\boldsymbol{c}_2^{\gamma} \cdot \sum_{j \in N_i^{\alpha}} \left(\dot{\boldsymbol{r}}_i \cdot \dot{\boldsymbol{r}}_i \right) \le 0$$
(32)

 Φ is strictly positive for every non-zero $\mathbf{r}_i, V, \dot{\mathbf{r}}_i$ in Eq. (31), and $\dot{\Phi} = 0$ if and only if $\dot{\mathbf{r}}_i = 0, i = 1, 2, \dots, n$. So $\dot{\Phi} = 0$ is the singleton $\{(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)^T = (0, 0, \dots, 0)^T\}$. The maximum invariant set of system on the set $\{(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)^T | \mathbf{r}_i \ge 0, i = 1, 2, \dots, n\}$ is the singleton $\{(0, 0, \dots, 0)^T\}$. It follows from LaSalle Invariance Principle that $\{(0, 0, \dots, 0)^T\}$ is globally asymptotically stable for the system in Eq. (29). As three polynomials in Eq. 31 are all nonnegative, we have $\dot{\mathbf{r}}_i = 0$ for $\Phi(t) = 0$, i.e., $\lim_{t \to \infty} \dot{\mathbf{r}}_i = 0$.

6 Simulation and result analysis

This section designs two scenarios to verify the efficiency of our method. Firstly, a delta formation of 6 agents is designed in three dimensions for the absence of obstacles. Secondly, a delta formation of 28 agents is designed in two-dimensional plane for the presence of 7 obstacles.

- (1) A delta formation in three dimensions
- (2) Fig. 5 represents the performance of delta formation for four time periods during simulation (0 s, 42 s, 84 s, and 105 s). Agents will form a delta formation on the basis of the strategy of formation control. Furthermore, the positions and velocities of agents are shown in Fig. 6(a) and Fig. 6(b) in each iteration. It is clear that six agents will keep a stabilized distance; while followers will remain the same altitudes and velocity with leader rapidly.



Figure 5: A delta formation in three dimensions



Figure 6: The positions and velocities of each agent for delta formation in each axis

The velocity oscillations are shown at the beginning of simulation in Fig. 6(b). The reason is represented as follows. Considering the control scheme of multi-agent system, all agents are constantly adjusting their motions to complete formation control based on relatively simple interactions among individuals or environments until the system reaches a steady state. The control scheme with interactions is low cost and simple to realize, and possesses some practical value in engineering. Meanwhile, the velocities become smooth and steady as time passes, which also verify the stability of this control scheme.

(3) A delta formation and obstacles avoidance

This scenario contains 28 agents and 7 obstacles. All agents will form a delta formation and avoid obstacles through the strategy of self-adaptation flocking algorithm. The positions and radius of all obstacles are specified as

	95	80	80	80	60	109	60	
$P_{obs} =$	57	120	82	40	100	30	80	
	7	5	4	8	5	6	7	



Figure 8: The positions and velocities of each agent for delta formation

Fig. 7 shows the performance of 28 agents to form a delta formation in two-dimensions and avoid obstacles through self-adaptation flocking algorithm for four time periods during simulation (0 s, 60 s, 90 s, and 180 s). Furthermore, the positions and velocities of all agents in each iteration are shown in Fig. 8(a) and Fig. 8(b). In Fig. 7, agents will form a desired formation from the initial position during iteration 0 to 280; while agents will avoid obstacles based on self-adaptation flocking algorithm after iteration 280 to 480; finally, agents will form the desired formation again after passing through obstacles. Further, all agents will form a desired formation on the basis of the strategy of formation control quickly with a stabilized distance and velocity when all agents are far away from obstacles; otherwise, agents will freely fly to pass through the area of obstacles without any collisions.

Furthermore, the traditional potential method is compared with self-adaptation flocking algorithm under the same conditions. One agent is trapped into the region of obstacles through traditional potential method, shown in Fig. 9. The main reason is probably that the repulsion force of β -agent takes the attractive force of γ -agent out. Additionally, the velocity error ($\sum_{i\in n} (v_i - v_{\gamma})$) in each iteration of our method and traditional potential method is depicted in Fig. 10. It is clear that all agents will complete velocities matching through self-adaptation flocking algorithm quickly when agents are far away from obstacles; while potential method will be significantly affected by obstacles, in which all agents cannot complete velocities matching.



Figure 9: An agent is trapped into the region of obstacles



Figure 10: The comparison between our method and traditional potential method

7 Conclusion

The methods of matrices are proposed to describe complex formation for multi-agents. Firstly, pattern matrices are used to record the graphical information for a desired formation. Secondly, location matrix is used to record the desired location for each agent. Furthermore, the desired positions of whole agents can be obtained through those two matrices in a global coordinate system. In order to form a desired formation and avoid obstacles, a self-adaptation flocking algorithm is proposed for multi-agents. Agents will form a desired formation through the strategy of formation control when far away from obstacles; and agents will freely fly when near by the area of obstacles.

References

Brandao, A. S.; Barbosa, J. P.; Mendoza, V.; Sarcinelli-Filho, M.; Carelli, R. (2014): A multi-layer control scheme for a centralized UAV formation. *International Conference on Unmanned Aircraft Systems*, pp. 1181-1187.

Benzerrouk, A.; Adouane, L.; Martinet, P. (2014): Stable navigation in formation for a multi-robot system based on a constrained virtual structure. *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1806-1815.

Essghaier, A.; Beji, L.; El Kamel, M. A.; Abichou, A.; Lerbet, J. (2011): Co-leaders and a flexible virtual structure based formation motion control. *International Journal of Vehicle Autonomous Systems*, vol. 9, no. 1/2, pp. 108-125.

Ge, X.; Han, Q. L.; Ding, D.; Zhang, X. M.; Ning, B. (2018): A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems. *Neurocomputing*, vol. 275, pp. 1684-1701.

Lü, J.; Chen, F.; Chen, G. (2016): Nonsmooth leader-following formation control of nonidentical multi-agent systems with directed communication topologies. *Automatica*, vol. 64, pp. 112-120.

Meng, D.; Jia, Y.; Du, J.; Zhang, J. (2014): On iterative learning algorithms for the formation control of nonlinear multi-agent systems. *Automatica*, vol. 50, no. 1, pp. 291-295.

Mas, I.; Kitts, C. (2010): Centralized and decentralized multi-robot control methods using the cluster space control framework. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 115-122.

Oh, K. K.; Park, M. C.; Ahn, H. S. (2015): A survey of multi-agent formation control. *Automatica*, vol. 53, pp. 424-440.

Olfati-Saber, R. (2006): Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401-420.

Park, B. S.; Yoo, S. J. (2015): Adaptive leader-follower formation control of mobile robots with unknown skidding and slipping effects. *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 587-594.

Su, H.; Wang, X.; Lin, Z. (2009): Flocking of multi-agents with a virtual leader. *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293-307.

Shi, L.; Zhao, H.; Zhang, X. (2019): Research on obstacle avoidance control strategy of networked systems based on leader-follower formation tracking. *Proceedings of 2018 Chinese Intelligent Systems Conference, Lecture Notes in Electrical Engineering*, vol. 529, pp. 105-115.

Tanner, H. G.; Jadbabaie, A.; Pappas, G. J. (2007): Flocking in fixed and switching networks. *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863-868.

Vilca, J.; Adouane, L.; Mezouar, Y. (2016): Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration. *Distributed Autonomous Robotic Systems*, pp. 237-254.

Xie, G.; Wang, L. (2007): Consensus control for a class of networks of dynamic agents. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 17, no. 10-11, pp. 941-959.

Yan, J.; Guan, X.; Luo, X.; Chen, C. (2017): Formation control and obstacle avoidance for multi-agent systems based on virtual leader-follower strategy. *International Journal of Information Technology & Decision Making*, vol. 16, no. 3, pp. 865-880.