

# 一种多机器人分布式编队策略与实现

户晓玲,王健安

(太原科技大学 华科学院,山西 太原 030024)

**摘要:**多机器人编队问题已经成为当前研究的热点,在执行一些如地震救援、军事搜索等复杂任务时,多个机器人通过保持一定的队形可以提高复杂任务的执行效率。为了让机器人依据要完成的目标快速形成目标队形,选择合适的编队控制策略十分重要。对此,提出了一种新的基于微粒群模型的多机器人编队算法。该算法利用分布式计算的方式控制多机器人编队,通过构造包含所有机器人位置信息的队形适应值函数,将函数取最优解时的变量作为目标队形的位置,采用微粒群算法优化适应值函数,将优化过程中的最优解作为机器人的最优位置,使机器人朝着最优解运动,成功实现了四种目标队形(线形、三角形、圆形和六边形)的编队控制和队形变换,仿真结果证明了该算法的有效性。

**关键词:**微粒群算法;分布式计算;机器人编队;队形变换

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2019)01-0021-05

**doi:**10.3969/j.issn.1673-629X.2019.01.005

## A Multi-robot Distributed Formation Strategy and Implementation

HU Xiao-ling, WANG Jian-an

(Huake College of Taiyuan University of Science and Technology, Taiyuan 030024, China)

**Abstract:** The issue of multi-robot formation has become a hot spot in current research. When performing complex tasks such as earthquake rescue and military search, multiple robots can improve the execution efficiency of complex tasks by maintaining a certain formation. In order to quickly form a target formation based on the objectives to be achieved for robots, it is important to select the appropriate formation control strategy. For this, we propose a new multi-robot formation algorithm based on particle swarm optimization. This algorithm uses distributed computing to control the formation of multiple robots. By constructing a shape-adaptive value function that contains all robot position information, the function is optimally solved. The time variable is used as the position of the target formation. The particle swarm optimization algorithm is used to optimize the fitness function. The optimal solution in the optimization process is used as the optimal position of the robot. The robot moves toward the optimal solution and successfully achieves four target formations (linear, triangular, circular and hexagonal) formation control and formation transformation. The effectiveness of the algorithm is proved by simulation results.

**Key words:** particle swarm optimization; distributed computation; robots formation; formation change

## 0 引言

自 20 世纪 90 年代开始,多机器人编队问题一直被众多机器人爱好者广泛研究。E. Bahceci 等<sup>[1]</sup>在 2003 年将多机器人编队问题定义为:一群机器人通过相互之间的协调,形成并保持一种特定几何形状的队形。多机器人编队控制<sup>[2-5]</sup>是指多个机器人组成的群体在适应环境约束的条件下保持特定的队形搜索到指定目标的一门控制技术。多机器人编队在执行如目标搜索、地图探测以及地震搜救等复杂任务中,编成特定队形展开目标搜索十分重要。因此,如何让机器人根

据指定的搜索目标控制编队迅速形成指定队形具有重要的研究意义。

对多机器人编队控制的研究最早有 Reynolds 提出的基于行为的编队控制模型<sup>[6]</sup>、J. P. Desai 等提出的领航跟随法和图论法<sup>[7]</sup>、Khatbi 等提出的人工势场法<sup>[8]</sup>,以及 Ren 等提出的虚拟结构法<sup>[9]</sup>。典型的成果有英国的 JAYDE 和美国的 VIJAY 用控制矩阵和状态转换图来实现编队的变化<sup>[10-11]</sup>,国内有沈阳自动化设计的 MARCAS 系统和陈卫东等提出的行为融合加权法<sup>[12-13]</sup>。随着任务难度越来越大,基本的编队算法已

收稿日期:2018-01-19

修回日期:2018-05-24

网络出版时间:2018-09-21

基金项目:山西省青年科技研究基金(201701D221107)

作者简介:户晓玲(1981-),女,硕士,研究方向为智能最优化方法。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20180920.1536.024.html>

经无法满足目前的需求,一些新的编队控制算法应运而生。

文中提出一种新的基于微粒群模型的机器人编队控制算法,该算法通过构造适应值函数,并采用微粒群算法获取适应值函数的最优解,将优化过程中的最优解作为机器人新的运动方向,并利用 MATLAB 软件进行仿真。

## 1 机器人编队优化模型

### 1.1 问题的描述和定义

定义一个二维连续空间表示物理环境:  $T^2 = \{(x, y), 0 \leq x \leq x_{\max}, 0 \leq y \leq y_{\max}\}$ 。

$\Omega$  为包含所有机器人的群体集合,  $\Omega = \{S_1, S_2, \dots, S_n\}$ ;  $S_i$  表示简单智能的机器人,坐标为  $P_{Si} = (x_i, y_i)$ 。

(1) 线形编队表示机器人与  $X$  轴横坐标的夹角为  $\theta$  ( $\theta \in (0 \sim 360^\circ)$ ), 两相邻机器人之间的距离为常数  $d$  的队形。

(2) 圆形编队表示机器人均匀排列成以极点为圆心, 半径为  $r$  的圆, 其中  $\theta_i$  为第  $i$  个机器人与  $X$  轴正半轴的夹角。

(3) 三角形编队表示以  $S_1$ 、 $S_2$  和  $S_3$  为三个顶点机器人排成的一个正三角形。假设机器人数量为  $n$ , 定义函数  $\text{round}(X)$  表示将  $X$  四舍五入取整。

第一条边上的机器人为:  $A_1 \sim A_{\text{round}((n+3)/3)}$ ;

第二条边上的机器人为:  $A_{\text{round}((n+3)/3)} \sim A_{\text{round}((n+\text{round}((n+3)/3))/2)}$ ;

第三条边上的机器人为:  $A_{\text{round}((n+\text{round}((n+3)/3))/2)} \sim A_n, A_1$ 。

(4) 六边形编队表示六个机器人排成正六边形。设机器人数量为  $n$ , 每条边上从起始机器人到终止机器人的确定方法为: 用  $S_1$  表示第一条边的终点机器人编号,  $S_i$  表示第  $i$  条边的终点机器人编号, 定义符号  $\lfloor \cdot \rfloor$  表示向下取整。则:

$$S_i = S_{i-1} + \lfloor (n - S_{i-1}) + (7 - i) / (7 - i) \rfloor, \\ i = 1, 2, \dots, 5, S_0 = 0$$

采用分布式控制方式, 将每个机器人初始化为一个微粒, 每个机器人根据微粒群算法 (PSO)<sup>[14]</sup> 优化目标队形 (线形、三角形、圆形、六边形) 的位置, 通过更新  $P_{\text{best}}$  来获得  $g_{\text{best}}$ , 机器人向  $g_{\text{best}}$  的方向运动, 实现多机器人编队和队形变换。

### 1.2 编队优化算法

编队优化算法的基本思路是构造一个包含  $\Omega$  中单个机器人位置信息的适应值函数  $f$ , 将  $f$  取极小值时的变量作为目标队形的最终位置。用 PSO 寻找  $f$  极值中的  $p_g$  (微粒迭代过程中的最好位置) 作为机器人  $\Omega$  的运动方向。机器人的位置和速度迭代公式如下:

$$X_i^{t+1} = X_i^t + \Delta t V_i^{t+1} \quad (1)$$

$$V_i^{t+1} = g(p_{gi} - p_{Ai}) \quad (2)$$

式 1 采用闭环控制, 机器人的期望位置  $p_{gi}$  和机器人所感知到的实际位置  $p_{Ai}$  之差为误差信号  $\partial$ , 即  $\partial = p_{gi} - p_{Ai}$ 。通过将机器人的运行速度  $V$  设定为  $\partial$  的函数来驱动机器人移动到正确的位置。现设  $V = \omega \partial$ ,  $\omega$  为权值, 有:

$$V_i^{t+1} = \begin{cases} \omega(p_{gi} - p_{Ai}), & V_i^{t+1} \leq v_{\max} \\ v_{\max}, & V_i^{t+1} > v_{\max} \end{cases} \quad (3)$$

算法的总体流程如图 1 所示。

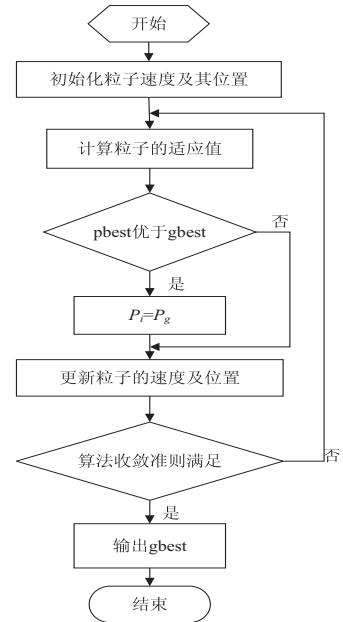


图 1 算法总体流程

### 1.3 不同队形适应值函数的构造

(1) 线形队形  $f$  如下:

$$f(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = \sum_{i=1}^{n-1} \{ [d \cos \theta - (x_i - x_{i+1})]^2 + [d \sin \theta - (y_i - y_{i+1})]^2 \} \quad (4)$$

其中,  $\theta$  指直线与  $X$  轴正方向夹角;  $d$  指两相邻机器人之间的直线距离。

(2) 三角形队形  $f$  见式 5, 设  $\text{round}((n+3)/3) = p$ ,  $\text{round}((n + \text{round}((n+3)/3))/2) = q$ 。

$$f(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = \sum_{i=1}^p \{ [d_1 \cos \theta_1 - (x_i - x_{i+1})]^2 + [d_1 \sin \theta_1 - (y_i - y_{i+1})]^2 \} + \sum_{j=p}^q \{ [d_2 \cos \theta_2 - (x_j - x_{j+1})]^2 + [d_2 \sin \theta_2 - (y_j - y_{j+1})]^2 \} + \sum_{k=q}^n \{ [d_3 \cos \theta_3 - (x_k - x_{k+1})]^2 + [d_3 \sin \theta_3 - (y_k - y_{k+1})]^2 \} +$$

$$\begin{aligned} & [(d_3 \cos \theta_3 - (x_n - x_1))^2 + \\ & [d_3 \sin \theta_3 - (y_n - y_1)]^2] \end{aligned} \quad (5)$$

其中,  $d_i$  为第  $i$  条边上相邻机器人之间的距离;  $\theta_i$  为第  $i$  条边与  $X$  轴正方向的夹角。通过调整  $f$  中的  $d_i$ ,  $\theta_i$  即可得到任意形状的三角形。若为正三角形, 则  $\theta_i = \theta_{i-1} + 2\pi/3, i = 2, 3$ 。

(3) 圆形队形是以极点为圆心,  $r$  为半径的圆, 适应值函数为:

$$\begin{aligned} f(r_1, \theta_1, r_2, \theta_2, \dots, r_n, \theta_n) = & \sum_{i=1}^n (r_i - r)^2 + \\ & \sum_{j=2}^n ((\theta_j - \theta_1) - (j-1) \times \frac{2\pi}{n})^2 \end{aligned} \quad (6)$$

(4) 六边形适应值函数如式 7 所示:

$$\begin{aligned} f(x_1, y_1, x_2, y_2, \dots, x_n, y_n) = & \sum_{i=1}^{s_1} \{ [d_1 \cos \theta_1 - (x_i - x_{i+1})]^2 + \\ & [d_1 \sin \theta_1 - (y_i - y_{i+1})]^2 \} + \\ & \sum_{j=S_1}^{s_2} \{ [d_2 \cos \theta_2 - (x_j - x_{j+1})]^2 + \\ & [d_2 \sin \theta_2 - (y_j - y_{j+1})]^2 \} + \\ & \sum_{k=S_2}^{s_3} \{ [d_3 \cos \theta_3 - (x_k - x_{k+1})]^2 + \\ & [d_3 \sin \theta_3 - (y_k - y_{k+1})]^2 \} + \\ & \sum_{l=S_3}^{s_4} \{ [d_4 \cos \theta_4 - (x_l - x_{l+1})]^2 + \\ & [d_4 \sin \theta_4 - (y_l - y_{l+1})]^2 \} + \\ & \sum_{m=S_4}^{s_5} \{ [d_5 \cos \theta_5 - (x_m - x_{m+1})]^2 + \\ & [d_5 \sin \theta_5 - (y_m - y_{m+1})]^2 \} + \\ & \sum_{i=S_5}^n \{ [d_6 \cos \theta_6 - (x_i - x_{i+1})]^2 + \\ & [d_6 \sin \theta_6 - (y_i - y_{i+1})]^2 + \\ & [d_6 \cos \theta_6 - (x_n - x_1)]^2 + \\ & [d_6 \sin \theta_6 - (y_n - y_1)]^2 \} \end{aligned} \quad (7)$$

其中,  $d_i$  指第  $i$  条边上相邻机器人之间的距离;  $\theta_i$  指第  $i$  条边与  $X$  轴正方向的夹角。通过调整  $f$  中的  $d_i$  和  $\theta_i$  可以得到任意形状的六边形。正六边形为  $\theta_i = \theta_{i-1} + \pi/3, i = 2, 3, 4, 5, 6$ 。

## 2 队列变换策略

四个基本队形的变换总体过程为: 线形  $\leftrightarrow$  三角形  $\leftrightarrow$  圆形  $\leftrightarrow$  六边形。

### 2.1 线形与三角形的变换

线形变三角形: 根据机器人个数  $n$  计算出  $S_1$  和  $S_2$ ,  $S_1 \sim S_2$  的机器人位置不变;  $1 \sim S_1 - 1$  的机器人以

$S_1$  为参考点排成直线, 斜率  $\theta_1 = \theta + 2\pi/3$ ,  $\theta$  为原直线的斜率; 目标函数如式 8 所示。编号  $S_2 + 1 \sim S_1$  的机器人以  $S_2$  为参考点排成直线, 直线斜率  $\theta_2 = \theta + \pi/3$ , 目标函数如式 9 所示。

$$\begin{aligned} f(x_{s_2+1}, y_{s_2+1}, x_{s_2+2}, y_{s_2+2}, \dots, x_s, y_s) = & \sum_{i=S_2+1}^s \{ [(i - S_2) \times d \cos \theta_2 - (x_i - x_{s_2})]^2 + \\ & [(i - S_2) \times d \sin \theta_2 - (y_i - y_{s_2})]^2 \} \end{aligned} \quad (8)$$

$$\begin{aligned} f(x_1, y_1, x_2, y_2, \dots, x_{s_1-1}, y_{s_1-1}) = & \sum_{i=1}^{s_1-1} \{ [(s_1 - i) \times d \cos \theta_1 - (x_i - x_{s_1})]^2 + \\ & [(s_1 - i) \times d \sin \theta_1 - (y_i - y_{s_1})]^2 \} \end{aligned} \quad (9)$$

三角形变线形:  $S_1 \sim S_2$  的机器人保持位置不变, 以  $S_1$  和  $S_2$  所在直线为另外两条边的斜率形成直线形。使用前面提到的直线形编队进行优化。

### 2.2 三角形与圆形的变换

三角形变圆形: 将  $1, S_1, S_2$  三个机器人的重心  $o$  作为极点, 圆的半径  $r$  可根据需要选择, 取所有的机器人到重心的平均距离作为  $r$ 。机器人的速度和位置迭代公式如式 10 和式 11 所示, 其中  $d_i$  为机器人到重心的距离。

$$V_{i+1} = g \times (X_i - O) \times (r - d_i) \quad (10)$$

$$X_{i+1} = X_i + V_{i+1} \times \Delta t \quad (11)$$

圆形变三角形: 根据  $x = r \times \cos \theta$  和  $y = r \times \sin \theta$  将极坐标系转化为直角坐标系, 通过  $n$  计算  $S_1$  和  $S_2$ , 则  $1, S_1, S_2$  成为三角形的三个顶点。再用式 12 和式 13 将两顶点之间的机器人运动到两顶点所在直线上, 其中  $o_1, o_2$  为两顶点的位置,  $n_1$  为两顶点间机器人的个数。

$$V_{i+1} = g \times [(o_1 - \frac{o_1 - o_2}{n_1 + 1} \times (i + 1)) - X_i] \quad (12)$$

$$X_{i+1} = X_i + V_{i+1} \times \Delta t \quad (13)$$

### 2.3 圆形与六边形的变换

圆形变六边形: 根据  $n_1$  计算出  $S_1 \sim S_5$ , 则  $1, S_1 \sim S_5$  变成六边形的六个顶点。用式 14 和式 15 将两顶点之间的机器人运动到两顶点所连成的线段上, 其中  $o_1, o_2$  为两顶点的位置。

$$V_{i+1} = g \times ((o_1 - \frac{o_1 - o_2}{n_1 + 1} \times (i + 1)) - X_i) \quad (14)$$

$$X_{i+1} = X_i + V_{i+1} \times \Delta t \quad (15)$$

六边形变圆形: 极点为  $1, S_1 \sim S_5$  六个机器人的重心  $o$ , 根据式 16 和式 17 更新机器人的速度和位置, 其中圆心半径  $r$  为所有机器人到重心的平均距离,  $d_i$  为机器人到重心的距离。

$$V_{i+1} = g \times (X_i - o) \times (r - d_i) \quad (16)$$

$$X_{i+1} = X_i + V_{i+1} \times \Delta t \quad (17)$$

3 仿真实验结果

3.1 不同形状编队仿真

图 2(a)是线形仿真,  $n = 7$ ,  $\theta_1 = 5\pi/6$ ,  $d = 10$  时, 迭代了 528 次。图中星号表示初始位置时机器人的团队重心。图 2(b)是三角形仿真,  $n_1 = 12$ ,  $\theta_1 = \pi/3$ , 边长为 60 时, 迭代 1 844 次的结果。图 2(c)是圆形仿真,  $n = 10$ ,  $r = 15$  时, 迭代 2 076 次的结果。图 2(d)是

六边形仿真,  $n = 20$ , 边长为 40,  $\theta_1 = \pi/3$  时, 迭代4 101 次的结果。

表 1 表示在不同机器人个数下对四种队形编队仿真时, 算法执行 50 次的平均迭代次数、最好迭代次数和最坏迭代次数, 以及迭代失败次数和迭代次数的方差。

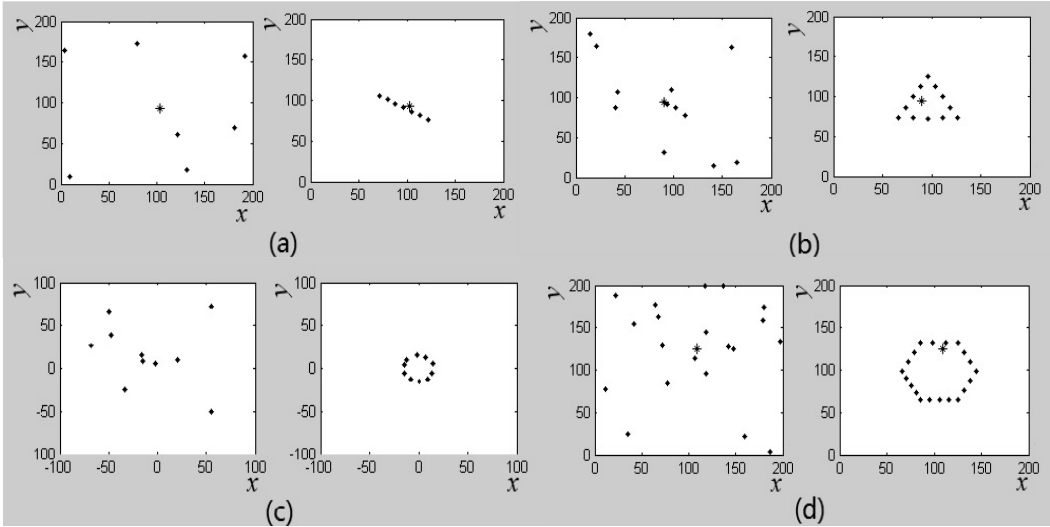


图 2 线形、三角形、圆形、六边形仿真

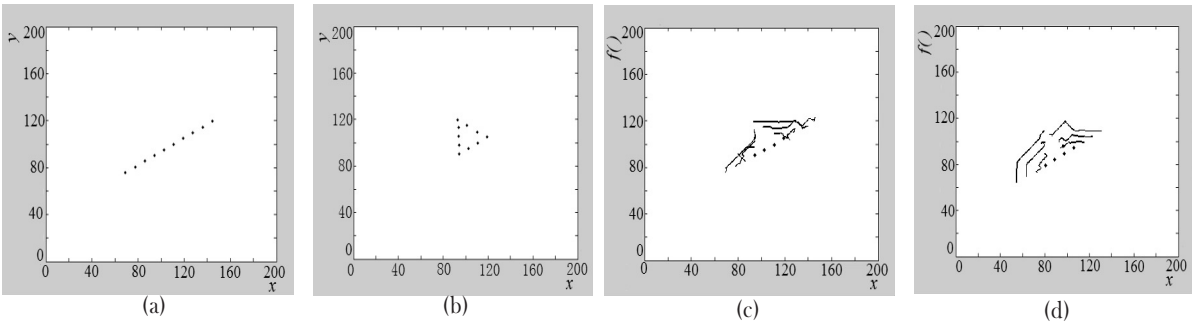
表 1 线形、三角形、圆形、六边形编队仿真比较

形状		算法执行 50 次 平均迭代次数	50 次中最好 迭代次数	50 次中最坏 迭代次数	所有迭代 次数的方差
线形	$\theta = 4 * \pi/3, n = 7$	449.440 0	278	608	75.339 5
	$\theta = 4 * \pi/3, n = 10$	1.023 3e+3	805	1 173	76.937 6
三角形	$\theta_1 = \pi/3, n = 8$	960.300 0	330	1 491	259.940 3
	$\theta_1 = \pi/6, n = 12$	1 864	1 234	2 379	240.864 1
圆形	$r = 15, n = 10$	2.233 8e+3	1 908	3 337	266.249 4
	$r = 30, n = 20$	3.228 5e+3	3 032	3 906	163.555 5
六边形	$\theta_1 = \pi/3, n = 15$	2.628 1e+3	2 088	3 004	208.547 5
	$\theta_1 = \pi/3, n = 20$	4.281 7e+3	3 876	4 624	167.311 4

3.2 四种形状队形变换仿真

图 3 机器人数量为 10,  $s_1 = 4$ ,  $s_2 = 7$ 。(c)、(d)的点为  $s_1$  与  $s_2$  之间的机器人。图(a)的线形到图(b)的

三角形, 迭代次数为 95, 运动轨迹为(c)。从图(b)的三角形到图(a)的线形, 迭代次数为 88, 运动轨迹为图(d)。



万方数据

图 3 线形和三角形变换

图 4 中的机器人数量为 20 个,迭代次数为 100,仿真中机器人是直接运动到目标点的。

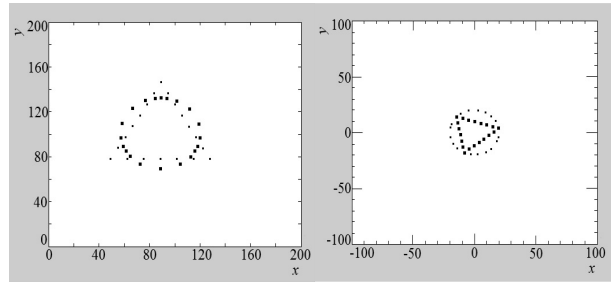


图 4 三角形与圆形的变换

图 5 中机器人数量为 18 个,固定迭代次数为 100。图 5 是圆形与六边形的变换。

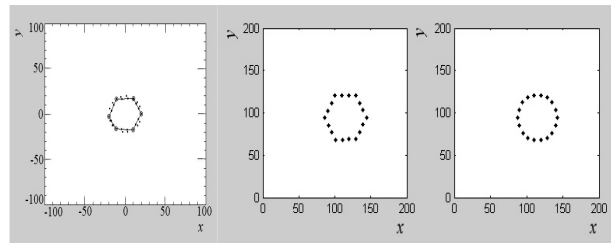


图 5 圆形与六边形的变换

从仿真结果可以看出,利用该计算方法能快速地实现多机器人编队和队形变换,自组织编队形成的目标队形重心与机器人初始化时的重心距离不远,从而体现了 PSO 算法优化的魅力。

## 4 结束语

提出了一种基于 PSO 的分布式编队控制算法。仿真结果表明,与集中式控制相比,分布式控制方式可以独立考虑每个机器人的特性,具有较强的灵活性、鲁棒性等优点。而采用 PSO 算法可以很快地达到适应值,收敛速度更快,迭代次数更少,有效提高了编队效率。但是文中研究的多机器人编队是静态的,如何让机器人动态编队完成跟踪任务还需要进一步研究;并且文中只考虑二维平面的机器人编队,编队过程中没有考虑障碍物,因此以后可以加入障碍物并从二维扩展到三维<sup>[15]</sup>进行研究。

## 参考文献:

[1] BAHCECI E, SOYSAL O, SAHIN E. A review: pattern formation and adaptation in multi-robot system[R]. Pittsburgh: Carnegie Mellon University, 2003.

[2] 韩学东,洪炳熔,孟 伟. 多机器人任意队形分布式控制研究[J]. 机器人, 2003, 25(1): 66-72.

[3] 高 翔,苏 青. 基于双层模糊逻辑的多机器人路径规划与避碰[J]. 计算机技术与发展, 2014, 24(11): 79-82.

[4] 杨 帆. 多移动机器人编队控制与协作运输研究[D]. 上海:华东理工大学, 2011.

[5] 刘 磊. 多移动机器人编队及协调控制研究[D]. 武汉:华中科技大学, 2009.

[6] 何锦璇. 考虑障碍环境下的多机器人编队控制研究[J]. 计算机技术与发展, 2013, 23(11): 30-33.

[7] SMET F D, AEYELS D. Cluster transitions in a multi-agent clustering model[C]//Proceedings of the joint 48th IEEE conference on decision and control and 28th Chinese control conference. Shanghai, China: IEEE, 2009: 4778-4784.

[8] 仇国庆,李芳彦,吴 健. 基于多智能体遗传算法的多机器人混合式编队控制[J]. 青岛科技大学学报:自然科学版, 2017, 38(2): 107-111.

[9] LEWIS M, TAN A K H. High precision formation control of mobile robots using virtual structures[J]. Autonomous Robots, 1997, 4(4): 387-403.

[10] 苏治宝,陆际联. 多移动机器人队形控制的研究方法[J]. 机器人, 2003, 25(1): 88-91.

[11] REN W, BEARD R W, ATKINS E M. Information consensus in multivehicle cooperative control: collective group behavior through local interaction[J]. IEEE Control Systems Magazine, 2007, 27(2): 71-82.

[12] 周浦城,洪炳熔,王月海. 动态环境下多机器人合作追捕研究[J]. 机器人, 2005, 27(4): 289-295.

[13] LAFFERRIERE G, WILLIAM A, CAUGHMAN J, et al. Decentralized control of vehicle formations[J]. Systems and Control Letters, 2005, 54(9): 899-910.

[14] 曾建潮,崔志华. 微粒群算法[M]. 北京:科学出版社, 2000.

[15] 郝爱锋. 基于改进粒子群算法的多机器人编队控制研究[D]. 南京:南京大学, 2017.