

引用格式:魏鹿鸣,马铁驹. 基于多智能体思想的辅助手部运动系统设计[J]. 北京化工大学学报(自然科学版),2020,47(2): 124-129.

WEI LuMing, MA TieJu. Design of a hand-assiting system based on a multi-agent model[J]. Journal of Beijing University of Chemical Technology (Natural Science), 2020,47(2):124-129.

# 基于多智能体思想的辅助手部运动系统设计

魏鹿鸣 马铁驹\*

(华东理工大学 理学院, 上海 200237)

**摘 要:**考虑运用多智能体思想,辅助手部活动不便患者实现日常的抓握、伸直等动作。于 Unity 引擎内设计并实现了一个基于多智能体蜂拥算法的辅助手部运动系统的仿真模拟,并对基础蜂拥算法进行了改进,使其更符合手指的运动方式。该系统包含 80 个智能体(智能体安装在类似手套的载体上,且与载体之间由弹簧关节连接)以及 5 个可随意变换位置的虚拟目标点,5 个目标点的位置分别代表患者左手 5 根手指意图到达的位置。辅助系统启动后,蜂拥算法控制的智能体开始自组织振动,通过弹簧带动手指逐渐运动至目标点,最终实现患者的运动意图。通过调整各目标点的位置,可充分辅助患者实现基本的抓握动作。该多智能体系统可根据实际情况灵活改变智能体的数目和大小,且具有良好的稳定性,为多智能体系统的应用提供了崭新的思路。

**关键词:**多智能体;蜂拥算法;辅助手部运动系统;Unity

**中图分类号:**TP273 **DOI:** 10.13543/j.bhxbzr.2020.02.017

## 引 言

多智能体系统(multi-agent system, MAS)目前主要应用于飞行编队、信号控制等领域,其思想源于自然界中的编队现象。多智能体系统的特殊性在于,每个智能体都能根据给出的全局目标进行自主规划,通过智能体之间的交流达到解决问题的目的,使得复杂系统的表达得到简化。哈佛大学自组织研究所设计了一个名为 Kilobot 的简单低成本机器人<sup>[1-2]</sup>,由若干个机器人组成的小型群体可以在平面上执行一系列简单的多智能体行为。Kilobot 机器人群体的面世使得多智能体系统在实际生活中的应用成为可能。

蜂拥算法(flocking algorithm)是一应用广泛且性能良好的多智能体算法。Reynolds<sup>[3]</sup>于 1987 年在其模拟程序 Boids 中首次模拟了多智能体的蜂拥行为。随后科研人员们就控制算法<sup>[4-5]</sup>、机器

人<sup>[6-7]</sup>、移动传感网络<sup>[8-9]</sup>等方面都进行了大量研究。其中,王沛磊等<sup>[4]</sup>提出了基于 Levy 飞行的蜂拥控制算法,采用 Levy 飞行策略对目标进行搜索,在移动智能体群体中加入搜索智能体,智能体根据周围邻居和感知到的目标信息调整自身运动;潘无为等<sup>[7]</sup>针对多自主式水下机器人(AUV)系统,设计了一种结合人工势场与虚拟结构的多 AUV 编队控制算法,可令 AUV 系统无碰撞地进行编队变换;娄柯等<sup>[8]</sup>构造了一个虚拟的领导者节点,可令其他网络各节点作为跟随者,跟踪领导者的运动轨迹,实现了网络拓扑结构连通性的保持。基于前人的成熟的研究成果可知,多智能体蜂拥算法能够良好地应用于实际生活。

现今有众多手部结构完整但活动不便的患者,已有研究人员为该类患者设计了不需要与人体神经对接的外骨骼系统<sup>[10]</sup>,它通过传感器感应人手的肌肉运动意图,外接机械系统通过杠杆产生目标需要的作用力,以此帮助患者进行正常的手部运动。

然而,传统的机械手系统结构固定,不易于根据患者手的大小不同灵活变换。由于多智能体系统具有易于更换且能够自主规划的特性,可考虑运用多智能体系统辅助患者的运动;同时,作为多智能体算法中的蜂拥算法可实现多智能体的自组织功能。因

收稿日期:2019-09-20

基金项目:国家自然科学基金(71571069)

第一作者:女,1995 年生,硕士生

\* 通信联系人

E-mail: tjma@ecust.edu.cn

此对蜂拥算法进行改进,使其能够适配于人体手指的运动,从而辅助患者进行日常的手部活动是一种可行的思路。

本文于 Unity 引擎内模拟了均匀安装在手指上的若干智能体单元组成的多智能体系统追踪目标点的运动行为,并带动手指以辅助手指弯曲的过程。基于多智能体原理制作辅助手部运动系统是一个崭新的思想,该系统可灵活改变智能体的大小和数目,且无需为每个智能体分别设置不同的算法或目标,为今后做出辅助系统的实体提供了可行的理论依据。

## 1 蜂拥算法

基础蜂拥算法由以下 3 个规则控制<sup>[3]</sup>:

(1) 分离 智能体与邻域内其他智能体避免碰撞;

(2) 聚合 智能体对邻域内其他智能体具有吸引力;

(3) 匹配 智能体与其他智能体保持速度相同。

采用 Olfati-Saber<sup>[11]</sup>设计的单虚拟领导者的蜂拥算法。该算法主要基于势能函数原理和一致性原理。在算法中,假设全部智能体都能接收到虚拟领导目标点的信息。

### 1.1 势能函数

基于势能函数的蜂拥算法的主要思想是在智能体周围模拟类似于原子之间的吸引力场和排斥力场,使得系统在稳定状态时能够实现基础蜂拥算法中的分离和聚合规则。文献[11]中提出的势能函数如式(1)所示。

$$\psi_{\alpha}(z) = \int_{d_{\alpha}}^z \phi_{\alpha}(s) ds \quad (1)$$

式中,

$$\phi_{\alpha}(z) = \rho_h(z/r_{\alpha})\phi(z-d_{\alpha}) \quad (2)$$

$$\phi(z) = \frac{1}{2}[(a+b)(z+k)/\sqrt{1+(z+k)^2} + (a-b)] \quad (3)$$

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ \frac{1}{2} \left\{ 1 + \cos \left[ \pi \frac{z-h}{1-h} \right] \right\}, & z \in [h, 1] \\ 0, & \text{其他} \end{cases} \quad (4)$$

式中,  $a, b, k, h$  为常数;  $r_{\alpha}(r_{\alpha} > 0)$  为传感器能够感知

的范围;  $d_{\alpha}(d_{\alpha} > 0)$  为智能体之间需要保持的距离。

### 1.2 一致性原理

多智能体系统中,智能体系统的状态方程式为

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t) \quad (5)$$

式中,  $\mathbf{x}_i(t) \in \mathbb{R}^m$  表示智能体  $i$  在  $t$  时刻的状态变量;  $\mathbf{u}_i(t) \in \mathbb{R}^m$  表示智能体  $i$  在  $t$  时刻的输入控制量。对于具有拓扑结构的网络, Olfati-Saber<sup>[11]</sup> 设计了线性一致性算法并验证了其有效性,该算法的表达式为

$$\mathbf{u}_i(t) = \mathbf{L}(t)\mathbf{x}(t) = \sum_{j \in N_i} \mathbf{a}_{ij}(t)[\mathbf{x}_j(t) - \mathbf{x}_i(t)] \quad (6)$$

式中,  $\mathbf{L}(t)$  为无权无向图的拉普拉斯矩阵,  $\mathbf{a}_{ij}(t)$  为网络时刻的邻接矩阵。

线性一致算法控制了智能体系统的稳定性,使得智能体的速度保持一致,可实现基础蜂拥算法中的匹配规则。

## 2 模型建立与仿真

Unity 是利用交互式图型化开发环境为首要方式的引擎,能较便捷地创作三维模型。本实验将在 Unity 引擎内进行模拟。其中,人体手部骨骼模型使用 Unity Assets Store 中的 Hand Physics Controller 模型,该模型是一节具有手腕的左手,手指的各个关节可自由活动,其中中指长度为 1 190 像素。假设在现实中系统外部有类似手套的载体,由于载体具有一定的弹性系数,故在此次模拟中通过弹簧关节 (spring joint) 来模拟实现载体的作用。

于 Unity 引擎中创建 80 个完全相同的球形智能体,智能体直径为 100 像素。将所有智能体通过弹簧关节均匀地固定在左手模型的 5 根手指上,每根手指分配到 16 个智能体(手指的上表面与下表面均放置 8 个),智能体之间间隔 150 像素。智能体的运动遵循蜂拥算法。为每根手指上的 16 个智能体分别设立 1 个目标点,目标点的位置可通过鼠标拖动随时更改。目标点所在位置即代表着患者的运动目的。

初始状态的手部模型如图 1 所示。

### 2.1 原始蜂拥算法及其改进

#### 2.1.1 原始蜂拥算法

由于单根手指上多智能体系统的智能体个数为 16,系统的运动方程<sup>[11]</sup>可表示为

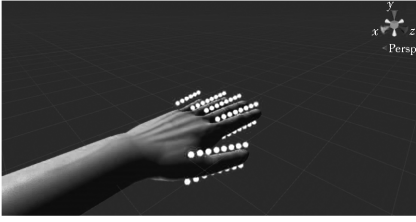


图1 智能体在手部分布情况

Fig. 1 Agent distribution on a hand

$$\begin{cases} \dot{\mathbf{q}}_i(t) = \mathbf{p}_i(t) \\ \dot{\mathbf{p}}_i(t) = \mathbf{u}_i(t) \end{cases} \quad i=1,2,\dots,16 \quad (7)$$

式中,  $\mathbf{q}_i(t) \in \mathbb{R}^m$  表示在  $t$  时刻智能体  $i$  的位置矢量;  $\mathbf{p}_i(t) \in \mathbb{R}^m$  表示在  $t$  时刻智能体  $i$  的速度矢量;  $\mathbf{u}_i(t) \in \mathbb{R}^m$  表示在  $t$  时刻智能体  $i$  的输入控制矢量。

系统虚拟目标点的运动方程<sup>[11]</sup>为

$$\begin{cases} \dot{\mathbf{q}}_\gamma(t) = \mathbf{p}_\gamma(t) \\ \dot{\mathbf{p}}_\gamma(t) = \mathbf{u}_\gamma(t) \end{cases} \quad (8)$$

式中,  $\mathbf{q}_\gamma(t) \in \mathbb{R}^m$  表示在  $t$  时刻虚拟目标点的位置矢量;  $\mathbf{p}_\gamma(t) \in \mathbb{R}^m$  表示在  $t$  时刻虚拟目标点的速度矢量;  $\mathbf{u}_\gamma(t) \in \mathbb{R}^m$  表示在  $t$  时刻虚拟目标点的输入控制矢量。式(8)经变换可以写作<sup>[11]</sup>

$$\mathbf{u}_i(t) = \mathbf{u}_i^\alpha(t) + \mathbf{u}_i^\beta(t) + \mathbf{u}_i^\gamma(t) \quad (9)$$

式中,

$$\mathbf{u}_i^\alpha(t) = \sum_{j \in N_i} \phi_\alpha [\|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|_\sigma] \times \frac{\mathbf{q}_j(t) - \mathbf{q}_i(t)}{\sqrt{1 + \sigma \|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|^2}} \quad (10)$$

$$\mathbf{u}_i^\beta(t) = \sum_{j \in N_i} \mathbf{a}_{ij}(t) [\mathbf{p}_j(t) - \mathbf{p}_i(t)] \quad (11)$$

$$\mathbf{u}_i^\gamma(t) = -c_1 [\mathbf{q}_i(t) - \mathbf{q}_\gamma(t)] - c_2 [\mathbf{p}_i(t) - \mathbf{p}_\gamma(t)] \quad (12)$$

$$\|z\|_\sigma = \frac{\sqrt{1 + \sigma \|z\|^2} - 1}{\sigma} \quad (13)$$

式中,  $c_1$ 、 $c_2$ 、 $\sigma$  为常数项;  $\mathbf{u}_i^\alpha(t)$  与  $\mathbf{u}_i^\beta(t)$  均为控制势能函数, 其中  $\mathbf{u}_i^\alpha(t)$  实现基础蜂拥算法中的分离规则和聚合规则,  $\mathbf{u}_i^\beta(t)$  控制速度一致, 实现匹配规则,  $\mathbf{u}_i^\gamma(t)$  为多智能体系统的虚拟领导者导航反馈量, 可诱导整个系统都跟随虚拟领导者运动, 最终实现蜂拥效果。

### 2.1.2 算法改进

原始蜂拥算法主要解决开放空间内的多智能体目标追踪问题, 但安装在手指表面的智能体受到手部结构的限制, 无法在空间中自由移动。正常手指弯曲时, 手指尖端的弯曲幅度明显大于手指根部。

针对手指这一特性作出算法改进, 将手指根部的位置矢量表示为  $\mathbf{q}_{\text{root}}$ 。首先将式(11)中的速度匹配项改进为

$$\mathbf{u}_i^\beta(t) = \sum_{j \in N_i} \mathbf{a}_{ij}(t) \left[ \mathbf{p}_j(t) \frac{\mathbf{q}_i(t) - \mathbf{q}_{\text{root}}}{\mathbf{q}_j(t) - \mathbf{q}_{\text{root}}} - \mathbf{p}_i(t) \right] \quad (14)$$

由于原文中已知所有控制点的原速度矢量  $\mathbf{p}_i(t)$  和原位置矢量  $\mathbf{q}_i(t)$ , 通过  $\mathbf{p}_i(t)$  及  $\mathbf{q}_i(t)$  可计算出即时的加速度矢量  $\mathbf{u}_i(t)$ 。现在加入一个远离系数  $\omega$ , 通过智能体到手指根部的距离之比计算出智能体当前状态的实时速度矢量  $\mathbf{p}'_i(t)$ 。以手指中点的速度为基准, 用  $\mathbf{q}_{\text{mid}}$  表示手指中点的位置矢量, 则改进后智能体的速度矢量  $\mathbf{p}'_i(t)$  可以表示为

$$\mathbf{p}'_i(t) = \mathbf{p}_i(t) + \omega \mathbf{u}_i(t) \Delta t \frac{|\mathbf{q}_i(t) - \mathbf{q}_{\text{root}}|}{|\mathbf{q}_{\text{mid}} - \mathbf{q}_{\text{root}}|} \quad (15)$$

可通过调整  $\omega$  的值来改变手指的运动速度。在本文实验中,  $\omega$  的值设置为 1。

## 2.2 仿真实验结果

启动基于改进蜂拥算法的多智能体系统后, 白色的球形智能体群体便会跟随目标点作蜂拥算法控制的自组织运动, 将手逐渐带动至目标点。在系统中, 目标点用白色五角星标示(图2)。

通过移动虚拟目标点的位置, 可令所有手指弯曲。在手指运动的可达范围内, 目标点越往远离手的方向移动, 手指弯曲程度越大。移动目标点后, 5 根手指的弯曲示意如图 2 所示。

由图 2 可见, 5 根手指都能通过移动目标点实现不同程度的弯曲, 并最终达到某一阈值。通过 5 根手指的共同作用, 可令整个左手实现抓握物体等动作。

## 3 实验结果分析

### 3.1 结果分析

以中指上的 16 个智能体为例, 将目标点设置在中指可达到的位置(图 3(a)), 此时目标点的坐标为  $(-300, -450, 0)$ 。取时间间隔  $\Delta t = 0.1$  s 来记录 16 个智能体向目标点运动过程中的即时位置和即时速度。中指开始运动和结束运动时的状态如图 3 所示。

由于中指弯曲主要是在  $xy$  平面上发生位置变化, 在  $z$  轴上几乎没有位置变化, 故分析时只需提取出智能体在  $x$  轴及  $y$  轴上的即时位置情况。

由图 4 可以看出, 16 个智能体相对手指的位置

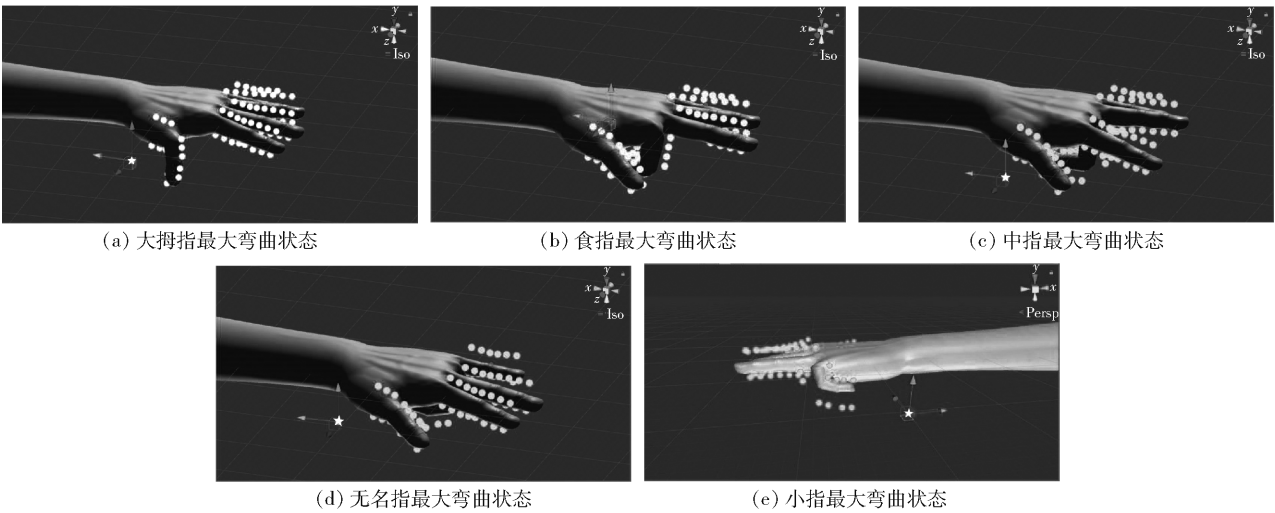


图 2 5 根手指的弯曲示意图  
Fig. 2 Bending states of the five fingers

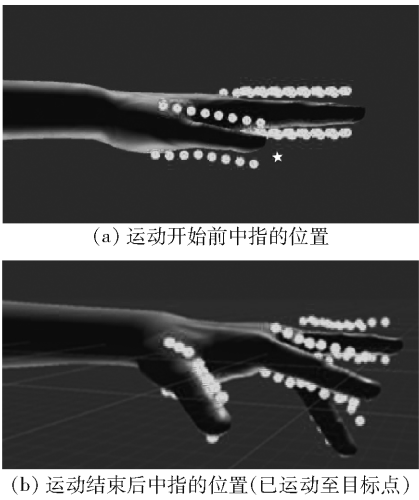


图 3 中指开始运动和结束运动时的状态  
Fig. 3 The position of the middle finger before and after the motion

不同,故到达目标点的时间也不尽相同;智能体的移动轨迹都较为平滑、连续,不易伤害到人体,故认为今后将该系统做成实体是可行的。

随后,提取出智能体速度在  $x$ 、 $y$ 、 $z$  轴上的分量。从图 5 中可以看出,智能体在  $x$  轴上的速度分量相对均衡,在  $t=2\text{ s}$  及  $t=6\text{ s}$  之间呈现出比较大的速度变化,在  $t=8\text{ s}$  时手指本身不再位移,此时认为系统达到了稳态。判断出系统达到稳态后,智能体将在  $2\text{ s}$  的延迟后自动停止运动。

由于中指主要向  $y$  轴的负半轴弯曲,故图 6 中智能体速度在  $y$  轴的分量于  $t=2\text{ s}$  时呈现出加速度向上的态势,原因在于系统阻止中指的迅速向下运动,使中指较为平滑地进行向下弯曲的动作;同样在

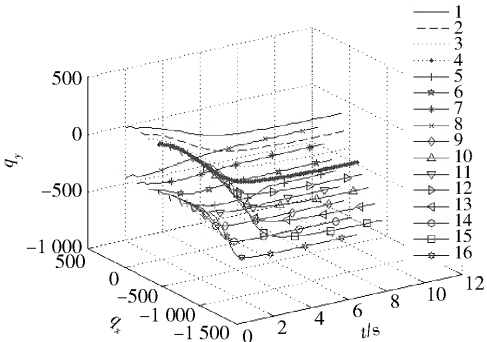


图 4 智能体在  $xy$  平面的位置随时间变化情况  
Fig. 4 Variation of the agent in the  $xy$  plane with time

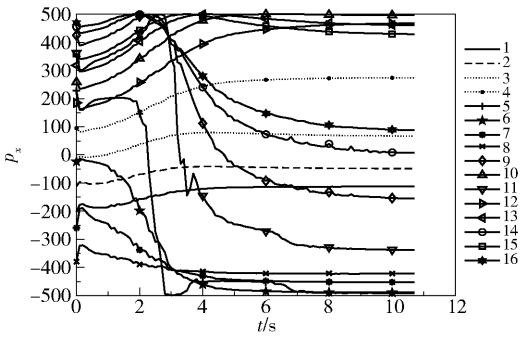


图 5 智能体速度在  $x$  轴的分量  
Fig. 5 The component of the speed of the agent on the  $x$ -axis  
 $t=8\text{ s}$  时系统大致达到了稳态。

$z$  轴方向上手指没有任何位移,故图 7 中智能体在  $z$  轴上的速度分量图像基本上是围绕  $p_z=0$  进行轻微震动。但图中有两条曲线(5 号曲线与 11 号曲线)代表的智能体速度相对突出。5 号曲线对应位于中指关节上方的智能体,而 11 号曲线对应位于



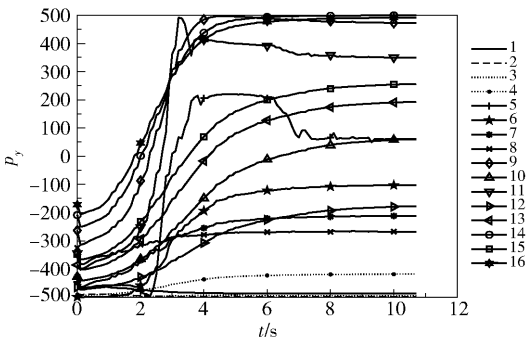


图6 智能体速度在y轴的分量

Fig.6 The component of the speed of the agent on the y-axis

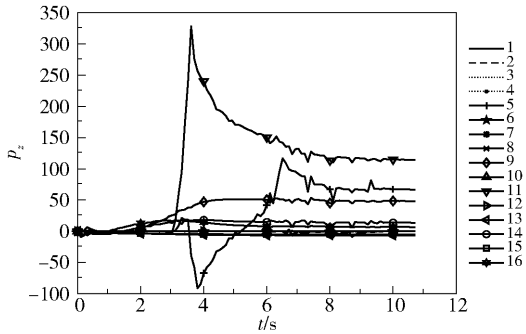


图7 智能体速度在z轴的分量

Fig.7 The component of the speed of the agent on the z-axis

中指关节下方的智能体。在手指关节处的智能体随着手部运动有一定的偏移量是正常表现,具体如何减少该偏移量还需今后进一步的研究和分析。

3.2 灵敏度测试

选取编号从1到9共9个位置的目标点(图8)来测试系统达到稳定所需时间。测试结果见表1。

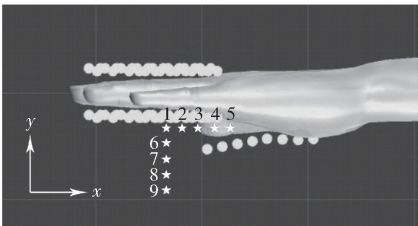


图8 灵敏度测试

Fig.8 Sensitivity test

从表1中可以看出,算法改进后较改进前系统达到稳定所需时间明显缩短,从而可认为改进后的蜂拥算法可以更好地匹配人体手指。同时,当y轴坐标固定在-300时,目标点越远离指尖,手指弯曲幅度越大,故系统达到稳态所需时间越长,此时手指完全弯曲所需时间约为18s;而当x坐标固定在-300时,由于目标点离指尖的远近变化并没有太多地影响到手指的弯曲程度,故运动时长也无明显

表1 目标点的坐标及其达到稳定所需时间

Table 1 The coordinates of the target point and the time it takes to stabilize

编号	x	y	稳定时间/s	
			改进前	改进后
1	-300	-300	10.8	8.4
2	-150	-300	14.2	9.8
3	0	-300	16.8	15.6
4	150	-300	18.1	17.3
5	300	-300	19.9	15.5
6	-300	-450	11.5	7.7
7	-300	-600	11.4	8.1
8	-300	-750	10.1	7.6
9	-300	-900	13.7	7.9

变化,基本稳定在8s左右。可见该系统的运动时长主要与手指弯曲幅度有关。

4 结论

(1)使用改进的蜂拥算法建立并仿真了一个多智能体手部运动辅助系统,通过多个智能体的运动带动手指运动,最终实现患者的运动意图。

(2)本文智能体系统只需为所有智能体设置相同的算法,当设置好目标点后,所有智能体便可自适应调整自己的运动速度和运动方向,直至抵达目标点。故当单个智能体损坏时可随时更换,硬件上容错率较高。

(3)本文智能体的数目可在一定范围内改变,可根据不同大小的手设置不同数目的智能体,拥有较高的灵活性。

(4)基于本文模拟所用的参数,系统的运动时长约在7~18s之间,运动轨迹平滑、连续,可认为该系统具有一定的安全性。该系统可根据患者手的大小调整智能体的个数及振动强度,并且可随时更换出故障的智能体,操作灵活,为多智能体系统的应用提供了崭新的思路。

目前已有现成的传感器,及现成的可运用多智能体算法的小型智能体,故今后将该系统做成实体是可行的。今后需要考虑调整传感器的测定方式,智能体的运动方式,以及找到合适的智能体的安装载体(如软性手套等)。

参考文献:

[1] RUBENSTEIN M, CORNEJO A, NAGPAL R. Programmable self-assembly in a thousand-robot swarm [J]. Science, 2014, 345(6198): 795-799.

- [2] RUBENSTEIN M, AHLER C, NAGPAL R. Kilobot: a low cost scalable robot system for collective behaviors [C] // 2012 IEEE International Conference on Robotics and Automation. Saint Paul, 2012: 3293–3298.
- [3] REYNOLDS C W. Flocks, herds and schools: a distributed behavioral model [C] // ACM Siggraph Computer Graphics. New York: Association for Computing Machinery, 1987, 21(4): 25–34.
- [4] 王帅磊, 张金春, 曹彪, 等. 基于 Levy 飞行的蜂拥控制算法在目标跟踪中的应用[J]. 兵工自动化, 2016, 35(7): 35–40.  
WANG S L, ZHANG J C, CAO B, et al. Application in target tracking of flocking control algorithm based on Levy flight [J]. Ordnance Industry Automation, 2016, 35(7): 35–40. (in Chinese)
- [5] 钟午, 张伟, 康怀祺, 等. 一种改进的多领导者蜂拥算法及其性能分析[J]. 小型微型计算机系统, 2014, 35(9): 2116–2121.  
ZHONG W, ZHANG W, KANG H Q, et al. An improved flocking algorithm with multiple leaders and its performance analysis [J]. Journal of Chinese Computer Systems, 2014, 35(9): 2116–2121. (in Chinese)
- [6] JIA Y N, WANG L. Experimental implementation of distributed flocking algorithm for multiple robotic fish [J]. Control Engineering Practice, 2014, 30: 1–11.
- [7] 潘无为, 姜大鹏, 庞永杰, 等. 人工势场和虚拟结构相结合的多水下机器人编队控制[J]. 兵工学报, 2017, 38(2): 326–334.  
PAN W W, JIANG D P, PANG Y J, et al. A multi-AUV formation algorithm combining artificial potential field and virtual structure [J]. Acta Armamentarii, 2017, 38(2): 326–334. (in Chinese)
- [8] 娄柯, 张艳, 李浩. 移动传感器网络保持网络连通性控制[J]. 电子测量与仪器学报, 2016, 30(11): 1657–1663.  
LOU K, ZHANG Y, LI H. Connectivity preservation control of mobile sensor network [J]. Journal of Electronic Measurement and Instrumentation, 2016, 30(11): 1657–1663. (in Chinese)
- [9] 娄柯, 崔宝同, 李纹. 基于蜂拥控制的移动传感器网络目标跟踪算法[J]. 控制与决策, 2013, 28(11): 1637–1642, 1649.  
LOU K, CUI B T, LI W. Target tracking algorithm of mobile sensor networks based on flocking control [J]. Control and Decision, 2013, 28(11): 1637–1642, 1649. (in Chinese)
- [10] AGARWAL P, DESHPANDE A D. Subject-specific assist-as-needed controllers for a hand exoskeleton for rehabilitation [J]. IEEE Robotics and Automation Letters, 2018, 3(1): 508–515.
- [11] OLFATI-SABER R. Flocking for multi-agent dynamic systems: algorithms and theory [J]. IEEE Transactions on Automatic Control, 2006, 51(3): 401–420.

## Design of a hand-assisting system based on a multi-agent model

WEI LuMing MA TieJu\*

(College of Science, East China University of Science and Technology, Shanghai 200237, China)

**Abstract:** Multi-agent system (MAS) have many uses in fields such as flight formation and signal control. This paper investigates the use of MAS in rehabilitation medicine to help patients with hand defects to carry out daily activities. A simulation of the hand-assisting system based on the flocking algorithm was designed and implemented in the Unity engine. The basic flocking algorithm was improved to make it more suitable for finger movement. This hand-assisting system consists of 80 agents which are mounted on the carrier such as a glove with a spring joint between the carrier and the finger, and five virtual target points that can be freely changed. The positions of the five target points represent the locations where the five fingers are intended to arrive. After turning on the hand-assisting system, the agent begins self-organized vibration, following by gradual movement of the finger towards the target point through the spring joint, finally realizing the patient's intended movement. By adjusting the position of each target point, it is possible to fully assist the patient in achieving a basic grasping action. The MAS can change the number and size of the agents flexibly and has good stability. This work suggests a new application for MAS.

**Key words:** multi-agent; flocking algorithm; hand-assisting system; Unity