doi:10.3969/j.issn.1005-3026.2014.03.010

一种面向多智能体群集的避障算法

赵 海,刘 倩,邵士亮,李大舟
(东北大学 信息科学与工程学院,辽宁 沈阳 110819)

摘 要:多智能体群集中的避障问题是研究的难点问题,每个智能体需要安全避开障碍物并朝着目标点前进.根据现有的基于人工势场函数的群集算法,提出一种改进的具有避障能力的群集算法.在该算法中,将障碍物等效成虚拟智能体进行避障.智能体感知到障碍物后,不是立即采取避障措施,而是将智能体的速度方向和目标点考虑在内,根据智能体不同的速度方向和目标点的位置,采取不同的避障措施.经理论分析与实验验证,表明所提出的算法能够有效地躲避障碍,并且在避开障碍物后更快地达到群集.

关 键 词:多智能体;群集;势场函数;避障;切线
中图分类号:TP 18 **文献标志码**:A **文章编号**:1005-3026(2014)03-0347-04

A Multi-agent Flocking Oriented Obstacle Avoidance Algorithm

ZHAO Hai, *LIU Qian*, *SHAO Shi-liang*, *LI Da-zhou*
(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: LIU Qian, E-mail: 874858509@qq.com)

Abstract: The problem of obstacle avoidance is important in multi-agent flocking. Each agent should avoid obstacles safely, and then moves toward the target. Based on the existing artificial potential field flocking algorithm, an improved algorithm with obstacle avoidance capability was presented. In this algorithm, the obstacle was equivalent to a virtual agent for obstacle avoidance. Obstacle avoidance were not taken immediately when the agent perceived obstacles, but take the speed direction of the agent and the target point into consideration. According to different speed directions and position of the target, different obstacle avoidance measures will be taken. Through theoretical analysis and experimental verification, obstacles could be avoided efficiently based on the proposed algorithm, which can make the flocking faster.

Key words: multi-agent; flocking; potential function; obstacle avoidance; tangent

智能体群集[1]是一群有着共同目标的并且可以相互通信的智能体的集体行为.几十年来,这种有局部通信范围的多智能体的群集行为吸引了不同学科的科学家研究[2].这些智能体可以是鸟、鱼、企鹅、蚂蚁和蜜蜂等.多智能体群集的工程应用包括特定环境下的大规模移动传感,无人机群体协作完成军事任务,比如侦查、监视和战斗.群集是自组织网中动态智能体完成协调任务的例子.

避障问题在多机器人和多车辆系统中是重要的.智能体的目标是靠近自己的目标位置并且在此过程中不会碰到障碍物.目前,多智能体避障控制问题的主要解决方法[3-4]有栅格法、神经网络方法、人工势场法等.其中,人工势场法是在多智能体运动中比较常用的方法,对于传统的势场法,智能体很容易陷入陷阱区域,所以出现了很多改进的势场法.文献[1]中,Olfati-Saber 等利用改进的一种势场函数,使智能体能够避开障碍物朝着目标点移动,改善了势场陷阱的局限.但是该算法中智能体只要感知到障碍物就进行避障,而没有考虑智能体速度的方向,并且下一时刻智能体的速度方向也不能确定.针对这些问题,本文将智能体的速度方向考虑在内,提出了改进的避障算法.

收稿日期:2013-06-17
基金项目:中央高校基本科研业务费专项资金资助项目(N100304002).
作者简介:赵 海(1959-),男,辽宁沈阳人,东北大学教授,博士生导师.

1 具有避障能力的群集算法

1.1 相关定义

定义 1 群集[1],当所有智能体在任意时间段 $t\in[t_0,t_f]$ 内,所有智能体的速度大小相同方向一致,并且两两之间距离稳定时,称为多智能体达到渐进群集. 本文采用 Reynolds 模型[5]规则.

定义 2 集体势函数[6],多个智能体的集体势函数 $\psi(z)$ 是一个非负函数 $V:R^{mn}\to R_{\geqslant 0}$,对于智能体 i 和 j 之间的距离 $\|q_j-q_i\|_\sigma$(本文采用向量的 σ 范数[1] $\|z\|_\sigma$ 表示两点间的距离)可微,所有智能体群集时,总势能无限接近 $\psi(z)$ 的局部极小点,反之亦然. 在本文中,势场函数的导数是作用函数,作用函数值决定了两个相邻的智能体间的作用力.

文献[1]将需要群集的智能体称为 α 智能体,群体的目标点称为 γ 智能体,在障碍物处等效的智能体称为 β 智能体,α 智能体和 β 智能体相互作用,达到避障的目的. 在本文中,继续使用这些概念.

1.2 有障碍物情况下的群集

多智能体系统的拓扑结构用邻接图 $G=(v,\varepsilon)$ 表示,其中 $v=\{1,2,\cdots,n\}$ 表示所有智能体的集合,$\varepsilon\in\{(i,j):i,j\in v,j\neq i\}$ 表示智能体间边的集合.

多智能体系统集体动力学公式为

$$\left.\begin{aligned}\dot{q}_i&=\boldsymbol{v}_i,\\ \dot{\boldsymbol{v}}_i&=u_i.\end{aligned}\right\}\tag{1}$$

式中,$q_i,\boldsymbol{v}_i,u_i\in\mathbf{R}^m(m=2,3,i\in v)$ 分别为智能体 i 的位置、速度和系统输入,$u_i=u_i^\alpha+u_i^\beta+u_i^\gamma$ 表示智能体 i 和其邻居 α 智能体、所感知到的虚拟 β 智能体以及目标点 γ 智能体之间的相互作用关系. 令 $R>0$ 表示智能体的感知半径,可以用一个半径为 R 的球形表示智能体 i 的空间邻居集合,该集合表示为

$$N_i=\{i,j\in v:\|q_j-q_i\|<R\}.$$

作用函数 $\Phi(z)$ 是一个不均匀的反曲函数,定义 $\Phi(z)=\frac{1}{2}[(a+b)\sigma_1(z+c)+(a-b)]$. $0<a\leqslant b,c=|a-b|/\sqrt{4ab},\sigma_1(z)=z/\sqrt{1+z^2}$,得出 $\Phi(0)=0$,此函数决定相邻两个智能体间远离或接近. 智能体输入为

$$u_i^\alpha=c_1^\alpha\sum_{j\in N_i^\alpha}\Phi_\alpha(\|q_j-q_i\|_\sigma)\boldsymbol{n}_{i,j}+c_2^\alpha\sum_{j\in N_i^\alpha}a_{ij}(q)(\boldsymbol{v}_j-\boldsymbol{v}_i),$$

$$u_i^\beta=c_1^\beta\sum_{j\in N_i^\beta}\Phi_\beta(\|\hat{q}_{i,k}-q_i\|_\sigma)\hat{\boldsymbol{n}}_{i,k}+c_2^\beta\sum_{j\in N_i^\beta}b_{i,k}(q)(\hat{\boldsymbol{v}}_{i,k}-\boldsymbol{v}_i),$$

$$u_i^\gamma=-c_1^\gamma\sigma_1(q_i-q_\gamma)-c_2^\gamma(\boldsymbol{v}_i-\boldsymbol{v}_\gamma).$$

其中:c_η^v 是正常量,$\eta=1,2,v=\alpha,\beta,\gamma$;$(q_\gamma,p_\gamma)$ 是静态或者动态 γ 智能体的状态;$\boldsymbol{n}_{i,j}$ 和 $\hat{\boldsymbol{n}}_{i,k}$ 分别为 q_j,q_i 连线上和 $\hat{q}_{i,k},q_i$ 连线上的向量;u_i^α 和 u_i^β 中的第二项为 i 与邻居智能体的速度匹配项.

对于一个半径为 R_k,圆心为 y_k 的球形障碍物,虚拟智能体 β 的位置和速度通过下面公式给出:

$$\hat{q}_{i,k}=\mu q_i+(1-\mu)y_k,\hat{\boldsymbol{v}}_{i,k}=\mu P\boldsymbol{v}_i.$$

其中:$\mu=R_k/\|q_i-y_k\|$;$P=I-\boldsymbol{a}_k\boldsymbol{a}_k^{\mathrm{T}}$;$\boldsymbol{a}_k=(q_i-y_k)/\|q_i-y_k\|$. β 智能体的速度 $\hat{\boldsymbol{v}}_{i,k}$ 和 α 智能体与 β 智能体之间的连线垂直.

1.3 避障算法局限性分析

保持其他项不变,对 β 智能体与 α 智能体之间的作用机制进行分析. β 智能体对 α 智能体的作用包含 β 智能体对 α 智能体的排斥作用和 β 智能体的速度与当前 α 智能体速度匹配项. 该方法总结为通过障碍物对智能体施加一种排斥作用和改变当前速度的方向. 在此算法中只判断 α 智能体和障碍物之间的距离,只要距离小于感知距离后,就采取避障措施,而没有判断 α 智能体的速度方向. 当 α 智能体的速度方向在障碍物区域外,却感知到障碍物时,u_i^β 中的第二项将阻碍 α 智能体远离障碍物的运动. 并且智能体 α 和 β 智能体的速度匹配,将使 α 智能体的速度方向趋向于 β 智能体的速度方向,而智能体 α 最终的速度方向是不能确定的,不利于群集的形成和目标点的接近. 没有针对 α 智能体,β 智能体,γ 智能体它们的位置关系进行分情况考虑,所以需要根据不同的情况,处理 β 智能体与 α 智能体之间的相互作用关系. 本文提出一种改进的避障算法,使智能体在有效躲避障碍物的同时,能快速地接近目标点并快速地形成群集.

2 改进的避障算法

智能体感知半径为 R,障碍物危险半径为 r. 智能体到障碍物圆心的向量为 $\boldsymbol{d}_y$,到危险区域做切线,切向量分别为 $\boldsymbol{t}_1,\boldsymbol{t}_2$,如图1所示. α智能体

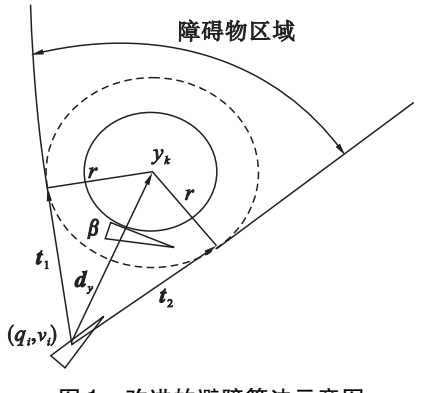


图 1 改进的避障算法示意图

Fig. 1 The improved obstacle avoidance algorithm

到 γ 智能体的向量为 $\boldsymbol{d}_\gamma$，智能体感知到障碍物后，判断速度方向，速度方向在障碍物区域内，即 $\det[\boldsymbol{t}_1,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{t}_2,\boldsymbol{v}_i]>0$[7] 时，采取避障措施. β 的速度由群集有没有形成以及目标点 γ 智能体的位置确定. 下面分情况讨论 β 智能体的速度.

2.1　群集未形成

群集未形成时，即智能体的速度与自己所有邻居的速度不相同时，α 智能体应以目标点为最终目标，尽可能朝向目标点运动. 所以躲避障碍物应以目标点 γ 智能体为基准.

1）γ 智能体在如图 1 所示的障碍物区域，下一时刻，α 智能体的速度应转向障碍物切线方向，这样可以在躲避障碍物的同时，尽可能小地偏离目标点. 此时，等效的 β 智能体的速度为

$$\boldsymbol{v}_\beta=\begin{cases}\|\boldsymbol{v}_i\|_\sigma\cdot\boldsymbol{t}_1/\|\boldsymbol{t}_1\|_\sigma, & \{\|\boldsymbol{d}_y\|_\sigma-r\leqslant R\}\cap\{\det[\boldsymbol{t}_1,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{d}_y,\boldsymbol{v}_i]>0\};\\ \|\boldsymbol{v}_i\|_\sigma\cdot\boldsymbol{t}_2/\|\boldsymbol{t}_2\|_\sigma, & \{\|\boldsymbol{d}_y\|_\sigma-r\leqslant R\}\cap\{\det[\boldsymbol{d}_y,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{t}_2,\boldsymbol{v}_i]>0\}.\end{cases}\tag{2}$$

2）γ 智能体在障碍物区域外时，下一时刻 α 智能体的速度应转向目标点方向，这样可以更好地朝向目标点. 此时，β 智能体的速度为

$$\boldsymbol{v}_\beta=\|\boldsymbol{v}_i\|_\sigma\cdot(q_i-q_\gamma)/\|(q_i-q_\gamma)\|_\sigma,\{\|\boldsymbol{d}_y\|_\sigma-r\leqslant R\}\cap\{\det[\boldsymbol{t}_1,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{t}_2,\boldsymbol{v}_i]>0]\}.\tag{3}$$

2.2　群集已形成

群集已经形成后，即智能体的速度和自己所有邻居的速度已经相同时，智能体应以尽量保持与邻居同速为目的，尽量不打乱已经形成的队形. 此时不需要再考虑目标点的位置，所以，遇到障碍物后，下一刻 α 智能体的速度应该转向障碍物切线方向. 此时，β 智能体的速度为

$$\boldsymbol{v}_\beta=\begin{cases}\|\boldsymbol{v}_i\|_\sigma\cdot\boldsymbol{t}_1/\|\boldsymbol{t}_1\|_\sigma, & \{\|\boldsymbol{d}_y\|_\sigma-r\leqslant R\}\cap\{\det[\boldsymbol{t}_1,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{d}_y,\boldsymbol{v}_i]>0\};\\ \|\boldsymbol{v}_i\|_\sigma\cdot\boldsymbol{t}_2/\|\boldsymbol{t}_2\|_\sigma, & \{\|\boldsymbol{d}_y\|_\sigma-r\leqslant R\}\cap\{\det[\boldsymbol{d}_y,\boldsymbol{v}_i]<0\wedge\det[\boldsymbol{t}_2,\boldsymbol{v}_i]>0\}.\end{cases}\tag{4}$$

2.3　改进后系统的输入

α 智能体 i 遇到障碍物后，应以躲避障碍为主要目标，所以输入项仅为相应的 β 智能体对 i 的作用项，而不考虑其同伴作用，因为同伴的输入项中包括与 α 智能体 i 之间的作用关系项.

没有遇到障碍物时，智能体 i 输入为 $u_i=u_i^\alpha+u_i^\gamma$；遇到障碍物时智能体 i 的输入为 $u_i=u_i^\beta$，$u_i^\beta=\boldsymbol{v}_\beta-\boldsymbol{v}_i$.

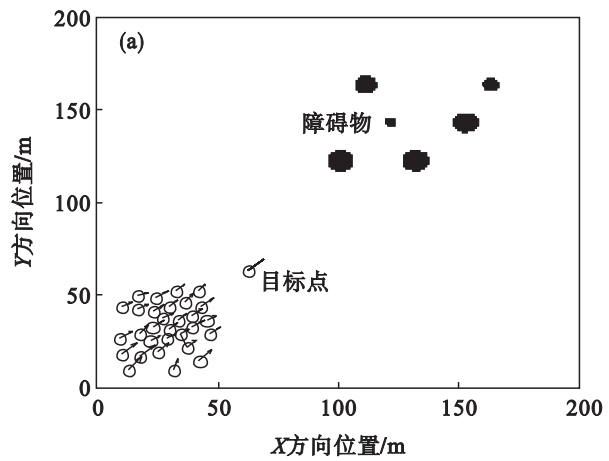
3　仿真结果

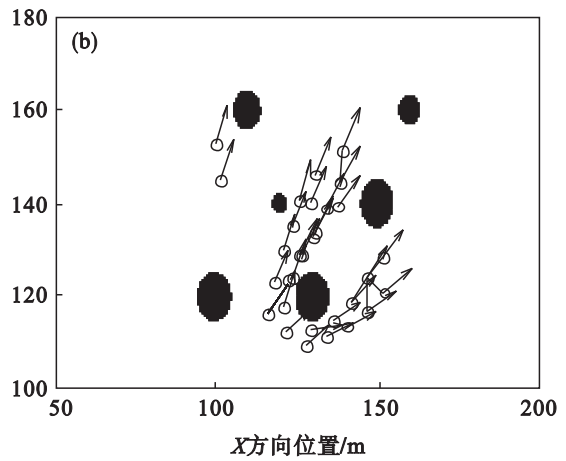
3.1　算法正确性验证

仿真环境为 Matlab，仿真参数设定为在二维空间内的 30 个将要群集的 α 智能体，初始位置横纵坐标在(0 ~ 50)内随机生成，目标点 γ 智能体为运动的，速度为 6.66 m/s. 每个智能体的感知半径 R 设定为 8.4 m，群集形成时，个体间的距离为 7 m，障碍物危险半径 r 为 9 m. 障碍物环境下，多智能体朝向特定目标点运动情况的仿真结果如图 2 所示，其中小的空心圆表示需要群集 α 智能体，小空心圆上的箭头代表 α 智能体的速度方向. 图 2a 中目标点即为 γ 智能体，实心圆表示障碍物. 当 α 智能体间的连线表示它们的相对位置达到稳定状态时，横坐标表示智能体在二维空间中运动的 x 方向的位置坐标，纵坐标表示 y 方向的位置坐标.

3.2　算法性能对比

文献[8]中提出一种衡量群集算法收敛速度的方法. 用群集行程，即群集已经达到的时刻所有智能体行程的均值[9]，来表示群集算法的收敛速度. 群集行程越小越好.





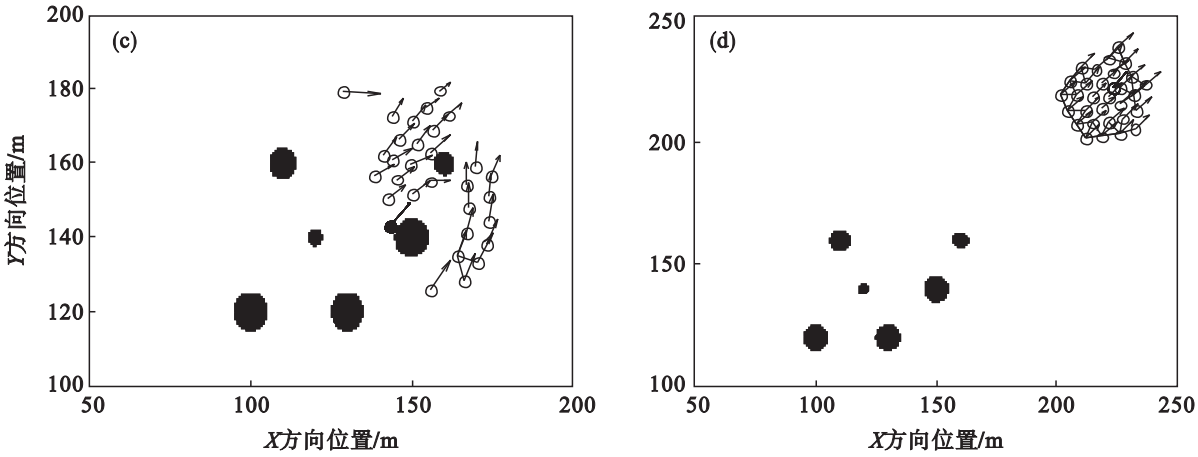


图 2 仿真结果图

Fig. 2 Simulation results

(a)—没有遇到障碍物时多智能体运动状态;(b)—采用本文的避障算法的避障过程图;(c)—采用本文的避障算法的避障过程图;(d)—多智能体躲避开障碍物后达到群集状态.

经分析本文提出的避障算法和 Olfati – Saber 算法的时间复杂度均为 $O(n^2)$. 群集行程对比如图 3 所示,可以看出,采用本文提出的避障算法,在没有提高复杂度的情况下,智能体能快速避开障碍物并达到群集.

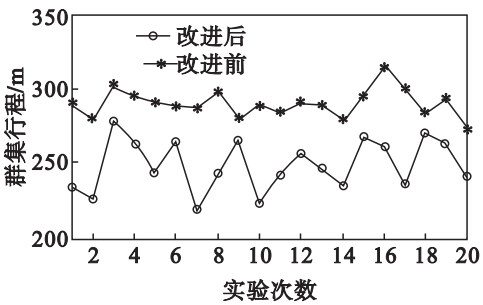


图 3 Olfati-Saber 算法和改进的避障算法群集行程对比图

Fig. 3 The flocking distance comparison of Olfati-Saber algorithm and the improved algorithm

4 结 论

1) 本文分析了现有避障算法,针对其中存在的不足进行改进,提出了改进的群集避障算法;

2) 经理论分析得出改进的算法在躲避障碍上与 Olfati – Saber 提出的避障算法相比具有更高的避障效率;

3) 在实验中使用群集行程作为评价指标,经验证该算法在没有提高算法复杂度的情况下,智能体能够更快地避开障碍物并完成群集.

参考文献:

[1] Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory[J]. *IEEE Transactions on Automatic Control*, 2006, 51(3): 401 – 420.

[2] Petter O, Fiorelli E, Leonard N E. Cooperative control of mobile sensor networks: adaptive gradient climbing in distributed environment[J]. *IEEE Transactions on Automatic Control*, 2004, 49(8): 1292 – 1302.

[3] Lu J H, Chen G R, Yu X H. Modelling, analysis and control of multi-agent systems: a brief overview[C]//IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE, 2011: 2103 – 2106.

[4] Cao Y C, Yu W W, Ren W. An overview of recent progress in the study of distributed multi-agent coordination[J]. *IEEE Transactions on Industrial Informatics*, 2013, 9(1): 427 – 438.

[5] Reynolds C. Flocks, herds, and schools: a distributed behavioral model[J]. *Computer Graphics*, 1987, 21(4): 25 – 34.

[6] Wen G H, Duan Z S, Su H S. A connectivity-preserving flocking algorithm for nonlinear multi-agent systems with bounded potential function [C]//Control Conference (CCC). Piscataway: IEEE, 2011: 6018 – 6024.

[7] Chang D E, Marsden J E. Gyroscopic forces and collision avoidance with convex obstacles[C]//Nonlinear Dynamics and Control. Berlin: Springer, 2003: 145 – 160.

[8] Xiao L, Boyd S. Fast linear iterations for distributed averaging [J]. *Systems & Control Letters*, 2004, 53(1): 65 – 78.

[9] Kim Y, Gu D W, Postlethwaite L. Spectral radius minimization for optimal average consensus and output feedback stabilization [J]. *Automatica*, 2009, 45(6): 1379 – 1386.