

控制与决策

Control and Decision

基于k-shell分解的多智能体牵制控制算法

何明, 许元云, 刘锦涛, 周波, 丁晓晖

引用本文:

何明, 许元云, 刘锦涛, 等. 基于k-shell分解的多智能体牵制控制算法[J]. *控制与决策*, 2020, 35(10): 2556–2560.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0173>

您可能感兴趣的其他文章

Articles you may be interested in

基于矩形波控制的多智能体系统静态多一致性与收敛性能优化

Stationary multi-consensus and convergence performance optimization of multi-agent systems via rectangular wave controller

控制与决策. 2020, 35(8): 2029–2036 <https://doi.org/10.13195/j.kzyjc.2018.1492>

符号网络下多智能体系统二分一致性的牵制控制问题

Leader-following bipartite consensus of multi-agent systems under signed networks

控制与决策. 2019, 34(8): 1695–1701 <https://doi.org/10.13195/j.kzyjc.2018.0035>

离散时变多智能体系统有限时间一致性迭代学习控制

Finite-time consensus iterative learning control of discrete time-varying multi-agent systems

控制与决策. 2019, 34(4): 891–896 <https://doi.org/10.13195/j.kzyjc.2017.1362>

事件触发机制下的二阶多智能体系统的一致性

Event-triggered consensus control of second-order multi-agent systems

控制与决策. 2017, 32(12): 2261–2267 <https://doi.org/10.13195/j.kzyjc.2016.1248>

基于事件触发的二阶多智能体领导跟随一致性

Leader-following consensus of second-order multi-agent systems via event-triggered control

控制与决策. 2016, 31(5): 835–841 <https://doi.org/10.13195/j.kzyjc.2015.0156>

基于 k -shell 分解的多智能体牵制控制算法

何明¹, 许元云^{1†}, 刘锦涛¹, 周波¹, 丁晓晖²

(1. 解放军陆军工程大学 指挥控制工程学院, 南京 210007; 2. 解放军 94860 部队, 南京 210000)

摘要: 针对多智能体网络在牵制控制过程中存在的网络分裂现象, 考虑到牵制节点选择对多智能体收敛速度的影响, 提出一种基于 k -shell 分解的牵制控制算法. 首先根据节点连通度划分子网; 然后提出基于 k -shell 分解的牵制节点选择方法; 最后完成多智能体的牵制控制. 理论推导证明, 采用该算法后整个智能体网络最终将形成一个子网. 分析对比 3 种牵制控制算法, 通过实验仿真结果验证所提出算法能够实现多智能体的一致性, 有利于提高多智能体的收敛速度.

关键词: 多智能体; 牵制控制; 一致性; k -shell 分解

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0173

开放科学(资源服务)标识码(OSID):



引用格式: 何明, 许元云, 刘锦涛, 等. 基于 k -shell 分解的多智能体牵制控制算法[J]. 控制与决策, 2020, 35(10): 2556-2560.

Multi-agent pinning control algorithm based on k -shell decomposition

HE Ming¹, XU Yuan-yun^{1†}, LIU Jin-tao¹, ZHOU Bo¹, DING Xiao-hui²

(1. Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China; 2. Unit 94860 of PLA, Nanjing 210000, China)

Abstract: Aiming at the network splitting phenomenon in the control process of multi-agent network, considering the influence of informed agent selection on the convergence speed of multi-agents, the pinning control algorithm based on k -shell decomposition is proposed. Firstly, the subnet is divided according to the node connectivity. Then, the method of selecting the informed agent based on k -shell decomposition is proposed. Finally, the pinning control of multi-agent is completed. Theoretical derivation proves that after the algorithm is adopted, the entire multi-agent network eventually form a connected graph. The experimental results verify that the proposed algorithm can achieve the consensus of multi-agent, and benefits to improve the convergence speed compared with three pinning control algorithms.

Keywords: multi-agent; pinning control; consensus; k -shell decomposition

0 引言

多智能体具有扩展性强、适应性强、运行成本低、复杂性低等优点, 已广泛应用于智能机器人、交通、柔性制造等诸多领域. 多智能体的表示一般使用 Boids 模型^[1]、Vicsek 模型^[2]、Couzin 模型^[3]等简化模型, 包括分离、聚集和速度匹配 3 个规则^[1]. 研究者在简化模型基础之上根据不同的限制条件或不同规则构建不同的控制算法以适应复杂的任务需求, 包括牵制控制算法^[4-5]、时滞控制算法^[6]、避障控制算法^[4,7]、非线性控制算法^[8]、输入饱和控制算法^[9]等.

牵制控制是用于复杂动态网络同步和共识的反馈控制策略. 具体而言, 一个或多个虚拟领导者被添加到网络中并定义其期望的轨迹, 虚拟领导者通过控

制输入控制网络节点的一小部分(牵制节点)完成整个网络的控制. Olfati-Saber^[4]在 3 个规则基础上提出使用 α -智能体、 β -智能体和 γ -智能体实现多智能体的一致性, 并以 α -智能体和 γ -智能体构建多智能体牵制控制的基本框架. Su 等^[5]证明了多智能体中少量智能体具有引导信息即可实现牵制控制.

由于多智能体网络在聚集过程中会发生复杂拓扑变化, 复杂网络的研究成果被引入到多智能体牵制控制中^[10-11]. 网络中最具影响力的节点是复杂网络中普遍关注的问题, 这种节点对网络功能具有重要意义. 为了量化节点在复杂网络中的重要性, 研究人员从不同角度提出了度中心性、介数中心性、接近度中心性等网络中心性度量. 文献[12]对部分中心性

收稿日期: 2019-02-17; 修回日期: 2019-05-08.

基金项目: 国家重点研发计划项目(2018YFC0806900); 中国博士后科学基金项目(2018M633757); 江苏省重点研发计划项目(BE2016904, BE2017616, BE2018754, BE2019762); 江苏省博士后科学基金项目(2019K185).

责任编辑: 关新平.

[†]通讯作者. E-mail: 718996172@qq.com.

进行了总结. 度中心性是最简单的测度, 它统计与节点相连的边的个数, Gao 等^[11] 通过划分子网, 选择度中心性完成多智能体牵制控制, 提出了自适应牵制控制算法 (dynamic pinning control algorithm, DPCA). 与特定网络功能结合时, 度中心性并不总是最佳选择, 多智能体网络由于其稀疏性可能存在大量度最大的节点, 文献^[11] 并未就度中心性相同的情况如何选择牵制节点进行深入研究. 牵制节点的一个主要任务是将虚拟领导者的信息传递给其他节点, Kitsak 等^[13] 提出使用经 k -shell 分解得到的重要性指标来衡量网络中节点的传播性能, 一些 k -shell 分解的重要改进包括混合度分解^[14]、基于引力观点的 k -shell 分解^[15] 等.

本文提出一种基于 k -shell 分解的牵制节点选择方法, 并以此构建了基于 k -shell 分解的多智能体牵制控制算法 (pinning control algorithm based on k -shell decomposition, PCAKD). 同时证明了采用本算法后所有智能体速度趋向虚拟领导者速度, 整个智能体网络最终会合并为一个子网, 且该子网牵制节点位置会趋向虚拟领导者位置. 通过实验验证所提出算法能够实现多智能体的一致性, 有利于提高多智能体的收敛速度.

1 多智能体图论基础

考虑 m 个智能体工作在 n 维欧几里得空间上, 其动力学方程可以写为

$$\begin{aligned} \dot{\mathbf{q}}_i &= \mathbf{p}_i, \quad \dot{\mathbf{p}}_i = \mathbf{u}_i, \\ i &= 1, 2, \dots, m. \end{aligned} \quad (1)$$

其中 \mathbf{q}_i 、 \mathbf{p}_i 、 $\mathbf{u}_i \in \mathbf{R}^n$ 分别为智能体 i 的位置向量、速度向量和加速度向量.

t 时刻多智能体构成的无向图 $\mathbf{G}(t)$ 由节点和边组成, 其中节点集合表示为 $V = \{1, 2, \dots, m\}$, 边的集合表示为

$$E(t) = \{(i, j) \in V \times V, j \in \mathcal{N}_i(t)\}.$$

设 r 是多智能体的感知半径, 期望距离为 d_α , 则智能体 i 邻域定义为

$$\mathcal{N}_i(t) = \{j \mid \|\mathbf{q}_i - \mathbf{q}_j\| < r, j \in V, j \neq i\}. \quad (2)$$

多智能体网络可以使用 Laplacian 矩阵表示, Laplacian 矩阵 \mathbf{L} 定义为

$$\mathbf{L} = \text{diag}\left(\sum_{j=1}^m a_{ij}\right) - \{a_{ij}\}. \quad (3)$$

其中仅当 $j \in \mathcal{N}_i$ 时 $a_{ij} = 1$, 否则 $a_{ij} = 0$. \mathbf{L} 满足如下平方和性质^[4]:

$$\mathbf{z}^T (\mathbf{L} \otimes \mathbf{I}_m) \mathbf{z} = \frac{1}{2} \sum_{j \in \mathcal{N}_i} a_{ij} (\mathbf{z}_j - \mathbf{z}_i)^2. \quad (4)$$

2 基于 k -shell 分解的多智能体牵制控制算法

2.1 牵制算法设计

多智能体网络动力学方程由式 (1) 定义, 在拓扑切换过程中多智能体网络会发生边断裂和边重连的现象, 当一个子网没有牵制节点时, 多智能体网络可能会分裂, 进而无法完成多智能体的一致性. 为避免这类现象的出现, 首先每个时刻根据多智能体的连接情况将多智能体划分为多个子网, 记作 $G_1(t), G_2(t), \dots, G_{M(t)}(t)$, 其中 $M(t)$ 为 t 时刻子网的数量; 然后对每个子网根据基于 k -shell 分解的牵引点选择方法选择牵制节点接收引导信息; 最后结合控制输入完成多智能体的牵制控制.

2.2 牵制节点选择方法

牵制节点的一个主要工作是将虚拟领导者的速度位置信息通过智能体之间的相互作用传递给其他节点. 文献^[13] 提出使用 k -shell 分解得到的节点重要性指标 k_s 衡量节点的传播能力.

k -shell 分解是一个连续修剪节点的过程. 首先删除所有度为 1 的节点, 一些节点可能度仍然为 1, 因此继续迭代地修剪网络, 直至没有度为 1 的节点, 删除的节点以及相应的链接形成索引为 $k_s = 1$ 的 k -shell. 以类似方式迭代地删除下一个 k -shell, 并继续删除更高的 k -shell, 直到删除所有节点. 最终每个节点与唯一的 k_s 索引相关联, 并且整个多智能体网络被视为所有 k -shell 的并集. 由此产生的节点分类可能与度中心性分类非常不同.

为避免多智能体网络节点 k_s 大量相同的情况, 结合牵制节点与虚拟领导者的距离, 本文提出改进的 \hat{k}_s 指标, 具体数学定义如下:

$$\hat{k}_s(i) = \frac{k_s(i)}{\|\mathbf{q}_i - \mathbf{q}_\gamma\|}, \quad (5)$$

其中 $\mathbf{q}_\gamma \in \mathbf{R}^n$ 为 t 时刻虚拟领导者的位置向量.

基于改进 k -shell 分解的牵制节点选择方法步骤如下.

step 1: 当 $t = 0$ 时, 遍历整个子网, 将 k_s 值最大的节点作为子网的牵制节点.

step 2: 当 $t > 0$ 时, 遍历整个子网, 如果子网中有两个及两个以上牵制节点, 则选择 \hat{k}_s 最大的牵制节点作为牵制节点; 如果子网中只有一个牵制节点, 则将该节点作为牵制节点; 如果子网中没有牵制节点, 则选择 k_s 值最大的节点作为牵制节点.

注 1 本文方法选用的节点与虚拟领导者的距离只涉及牵引节点与虚拟领导者的距离, 可以通过后文的公式计算得出.

2.3 控制输入和一致性证明

设 t 时刻控制输入为

$$\mathbf{u}_i(t) = \mathbf{f}_i^\alpha(t) + h_i(t)\mathbf{f}_i^\gamma(t). \quad (6)$$

其中: \mathbf{f}_i^α 、 \mathbf{f}_i^γ 采用文献[4]的定义, \mathbf{f}_i^α 实现多智能体的3个规则, \mathbf{f}_i^γ 接收虚拟领导者信息, 其数学表示式为

$$\mathbf{f}_i^\alpha(t) = \sum_{j \in \mathcal{N}_i(t)} \nabla_{\mathbf{q}_i(t)} \psi_\alpha(\|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|_\sigma) + \sum_{j \in \mathcal{N}_i(t)} a_{ij}(\mathbf{q}(t))(\mathbf{p}_j(t) - \mathbf{p}_i(t)), \quad (7)$$

$$\mathbf{f}_i^\gamma(t) = -c_1^\gamma(\mathbf{q}_i(t) - \mathbf{q}_\gamma(t)) - c_2^\gamma(\mathbf{p}_i(t) - \mathbf{p}_\gamma(t)), \quad (8)$$

$\mathbf{p}_\gamma \in \mathbf{R}^n$ 为 t 时刻虚拟领导者的速度向量. $h_i(t)$ 为与牵制节点的选择和时间相关的参数, t 时刻 $h_i(t)$ 的数学表示为

$$h_i(t) = \begin{cases} 1, & \text{节点 } i \text{ 是牵制节点;} \\ 0, & \text{节点 } i \text{ 不是牵制节点.} \end{cases} \quad (9)$$

定理1 考虑一个多智能体运动方程如式(1)所示, 控制输入如式(6)所示, 初始能量 Q_0 为一个有限值, 则有:

- 1) 所有智能体最终会趋向一致^[5];
- 2) 所有智能体速度会趋向虚拟领导者速度 $\mathbf{p}_\gamma(t)$ (对文献[5]速度一致性证明的改进);
- 3) 智能体之间不会发生碰撞^[5];
- 4) 整个智能体网络最终会形成一个子网, 且该子网牵制节点位置会趋向虚拟领导者位置 $\mathbf{q}_\gamma(t)$ (对文献[11]位置结论的拓展).

证明 设 $\tilde{\mathbf{q}}_i(t) = \mathbf{q}_i(t) - \mathbf{q}_\gamma(t)$, $\tilde{\mathbf{p}}_i(t) = \mathbf{p}_i(t) - \mathbf{p}_\gamma(t)$, $\mathbf{q}_{ij}(t) = \mathbf{q}_i(t) - \mathbf{q}_j(t)$, 多智能体的总能量由总势能和总动能构成. 由式(6)~(8), 总能量为

$$Q(t) = \frac{1}{2} \sum_{i=1}^m (U_i(t) + \tilde{\mathbf{p}}_i(t)^\top \tilde{\mathbf{p}}_i(t)). \quad (10)$$

其中

$$U_i(t) = \sum_{j=1, j \neq i}^m \psi_\alpha(\|\mathbf{q}_{ij}(t)\|_\sigma) + h_i(t)c_1^\gamma \tilde{\mathbf{q}}_i(t)^\top \tilde{\mathbf{q}}_i(t). \quad (11)$$

$Q(t)$ 对时间 t 求导, 得到

$$\dot{Q}(t) = -\tilde{\mathbf{p}}(t)^\top (\mathbf{L}(t) \otimes \mathbf{I}_m) \tilde{\mathbf{p}}(t) - c_2^\gamma \sum_{i=1}^m h_i(t) \tilde{\mathbf{p}}_i(t)^\top \tilde{\mathbf{p}}_i(t). \quad (12)$$

1) $\tilde{\mathbf{p}}_i(t)$ 和 $\tilde{\mathbf{q}}_i(t)$ 对应的集合 Ω 数学表示为

$$\Omega = \{[\tilde{\mathbf{q}}(t)^\top, \tilde{\mathbf{p}}(t)^\top]^\top \in \mathbf{R}^{2mn} | Q(t) \leq Q_0\}. \quad (13)$$

由文献[5], Ω 会趋向最大的不变集 $\bar{\Omega}$, 有

$$\bar{\Omega} = \{[\tilde{\mathbf{q}}(t)^\top, \tilde{\mathbf{p}}(t)^\top]^\top \in \mathbf{R}^{2mn} | \dot{Q}(t) = 0\}. \quad (14)$$

2) 由式(4)和(12), 有

$$\dot{Q}(t) = - \underbrace{\sum_{j \in \mathcal{N}_i} a_{ij}(\tilde{\mathbf{p}}_i(t) - \tilde{\mathbf{p}}_j(t))^2}_{\dot{Q}_1(t)} - \underbrace{\sum_{i=1}^m h_i(t) \tilde{\mathbf{p}}_i(t)^\top \tilde{\mathbf{p}}_i(t)}_{\dot{Q}_2(t)}, \quad (15)$$

其中 $\dot{Q}_1(t)$ 、 $\dot{Q}_2(t) \leq 0$.

$\dot{Q}(t) = 0$ 成立的充分必要条件是 $\dot{Q}_1(t) = \dot{Q}_2(t) = 0$. 由 $\dot{Q}_1(t) = 0$, 对于子网 $G_{M_k(t)}$, 有

$$\tilde{\mathbf{p}}_i(t) = \tilde{\mathbf{p}}_j(t), \quad \forall i, j \in G_{M_k(t)}. \quad (16)$$

假设 \mathbf{p}_1^i 是子网 i 的牵制节点的速度向量, 由 $\dot{Q}_2(t) = 0$, 对于牵制节点 \mathbf{p}_1^i , 有

$$\mathbf{p}_1^i(t) = \mathbf{p}_\gamma(t). \quad (17)$$

由式(16)和(17), 可得

$$\mathbf{p}_1(t) = \dots = \mathbf{p}_m(t) = \mathbf{p}_\gamma(t). \quad (18)$$

3) 当 Q_0 为一个有限值时, 由文献[5]可知, 多智能体不会发生碰撞.

4) 由 $\mathbf{p}_i(t) = \mathbf{p}_\gamma(t)$, 当 i 不是牵制节点时, 由式(6)和(7)可得

$$\mathbf{u}_i(t) = \sum_{j \in \mathcal{N}_i(t)} \nabla_{\mathbf{q}_i(t)} \psi_\alpha(\|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|_\sigma) = 0. \quad (19)$$

当 i 是牵制节点时, 由式(6)~(8)可得

$$\mathbf{u}_i(t) = \sum_{j \in \mathcal{N}_i(t)} \nabla_{\mathbf{q}_i(t)} \psi_\alpha(\|\mathbf{q}_j(t) - \mathbf{q}_i(t)\|_\sigma) - c_1^\gamma(\mathbf{q}_i(t) - \mathbf{q}_\gamma(t)) = 0. \quad (20)$$

由对称性可知

$$\mathbf{f}_i(t) = -c_1^\gamma(\mathbf{q}_i(t) - \mathbf{q}_\gamma(t)) = 0, \quad (21)$$

即

$$\mathbf{q}_i(t) = \mathbf{q}_\gamma(t). \quad (22)$$

假设最终存在多个子网. 不失一般性, 假设存在 i, j 两个牵制节点, 则有

$$\mathbf{q}_i(t) = \mathbf{q}_j(t) = \mathbf{q}_\gamma(t). \quad (23)$$

由定理1的3)可知智能体不会发生碰撞, 即

$$\mathbf{q}_i(t) \neq \mathbf{q}_j(t). \quad (24)$$

式(23)与(24)矛盾, 因此整个智能体网络最终会合并成一个子网, 同时由式(23), 牵制节点的位置会趋向虚拟领导者的位置. \square

3 模拟仿真

实验1 本实验研究智能体基于PCAKD的速度和位置变化. 一组智能体初始速度和位置如图1(a)所示, 其中蓝点表示多智能体, 旁边标注黑色圆圈表示该智能体为牵制节点, 五角星表示虚拟领导者. 所有智能体的感知半径为 $r = 6$, 期望距离为 $d_\alpha = 5$, 控制输入参数 $c_1^\gamma = 0.5, c_2^\gamma = 2$. 虚拟领导者的位置为 $[25, 25]$, 速度为 $[0.5, 0.5]$. 图1(b)显示智能体向着虚拟领导者聚集. 由图1(c)可见, 牵制节点和虚拟领导者位置趋于重合, 验证了定理1的4).

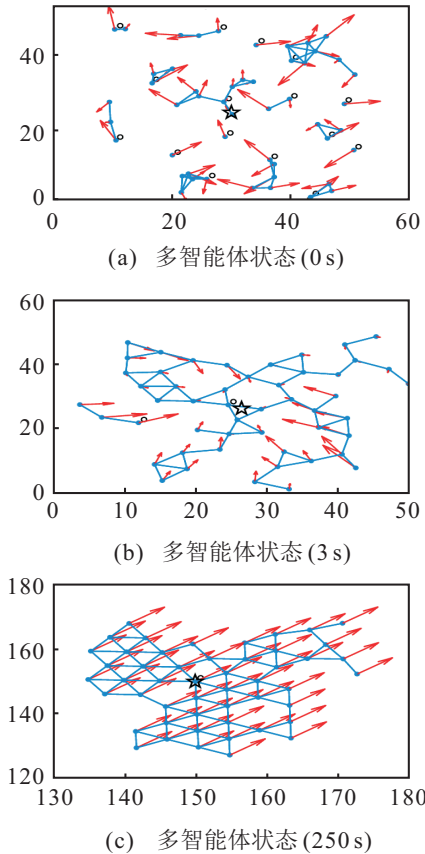


图1 基于PCAKD的多智能体状态

设定智能体 i 速度差异为

$$v_i = \frac{1}{2}(\mathbf{p}_i^x + \mathbf{p}_i^y - \mathbf{p}_\gamma^x - \mathbf{p}_\gamma^y). \quad (25)$$

其中 $\mathbf{p}_i^x, \mathbf{p}_i^y, \mathbf{p}_\gamma^x, \mathbf{p}_\gamma^y$ 分别为 x 轴、 y 轴方向的智能体 i 和虚拟领导者的速度分量. 基于PCAKD的多智能体速度差异结果如图2所示. 由图2可见, 多智能体的速度趋近于虚拟领导者的速度, 验证了定理1的2).

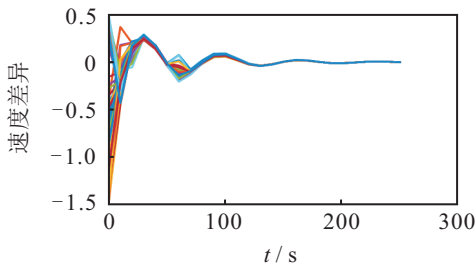


图2 基于PCAKD的多智能体速度差异

实验2 本实验研究PCAKD中参数 c_1^γ 和 c_2^γ 对多智能体蜂拥收敛速度的影响. 一组智能体初始速度和位置如图1(a)所示. 设置终止条件为 $\|\mathbf{p}_i - \mathbf{p}_\gamma\| \leq 0.0001 (i=1, 2, \dots, m)$ 且 $\sum_{i=1}^m h_i(t) = 1$. 图3(a)为 c_1^γ 对多智能体收敛速度的影响, 其中 $c_2^\gamma = 2$; 图3(b)为 c_2^γ 对多智能体收敛速度的影响, 其中 $c_1^\gamma = 0.5$, 智能体网络的其他参数设置同实验1. 由图3(a)可见, c_1^γ 与多智能体收敛速度不相关, 这主要是因为 c_1^γ 影响着多智能体的聚集速度, 聚集过程的拓扑变化对收敛速度的影响不是相关的. 由图3(b)可见, c_2^γ 越大, 多智能体的收敛速度总体越快, 但收敛速度有上界.

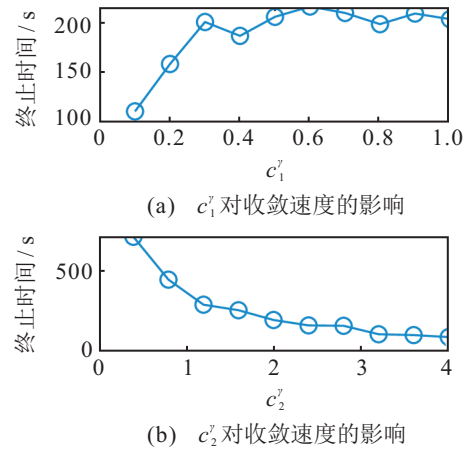


图3 参数 c_1^γ, c_2^γ 对多智能体收敛速度的影响

实验3 本实验主要比较基于不同中心性的牵制控制算法对多智能体收敛速度的影响. 50个智能体初始位置随机分布在 $[0, 50] \times [0, 50]$ 的空间内, 初始速度随机分布在 $[-1, 1] \times [-1, 1]$ 区间内, 智能体网络的其他参数设置同实验1. 设置终止条件为 $\|\mathbf{p}_i - \mathbf{p}_\gamma\| \leq 0.0001$ 且 $\sum_{i=1}^m h_i(t) = 1$. 重复10次实验, 结果如图4所示. 由图4可见, 相比于基于度中心性、介数中心性、接近度中心性的牵制控制算法, PCAKD在多数情况下能够更快实现一致性. 同时注意到接近度中心性也表现出较好的效果, 表明多智能体的收敛与牵制节点和普通节点的距离也有部分关系. 第4组实验250s时多智能体状态如图5所示, 牵制节点的选择偏离接近度中心性, 可能是结果不如接近度中心性和介数中心性的原因之一.

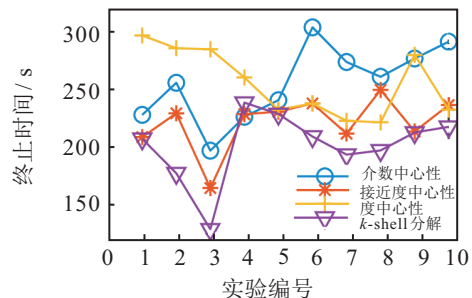


图4 不同中心性牵制控制算法对多智能体收敛速度影响

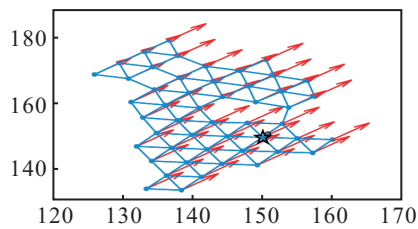


图5 第4组实验250s时多智能体状态

4 结论

本文提出了基于 k -shell 分解的多智能体牵制控制算法,通过实验验证了所提出的算法能够实现多智能体的一致性,有利于提高多智能体收敛速度.考虑到在多智能体控制算法中其他因素对收敛速度的影响,下一步继续就牵引节点的选择进行深入研究. PCAKD 在实际过程中需要得到整个子网的拓扑信息,加上感知智能体需要时间,下一步将对多智能体如何快速准确地获知子网拓扑信息开展研究,并在本算法基础上研究有时延的牵制控制算法.

参考文献(References)

- [1] Reynolds C W. Flocks, herds and schools: A distributed behavioral model[J]. ACM SIGGRAPH Computer Graphics, 1987, 21(4): 25-34.
- [2] Vicsek T, Czirók A, Ben-Jacob E, et al. Novel type of phase transition in a system of self-driven particles[J]. Physical Review Letters, 1995, 75(6): 1226-1229.
- [3] Couzin I D, Krause J, James R, et al. Collective memory and spatial sorting in animal groups[J]. Journal of Theoretical Biology, 2002, 218(1): 1-11.
- [4] Olfati-Saber R. Flocking for multi-agent dynamic systems: Algorithms and theory[J]. IEEE Transactions on Automatic Control, 2006, 51(3): 401-420.
- [5] Su H S, Wang X F, Lin Z L. Flocking of multi-agents with a virtual leader[J]. IEEE Transactions on Automatic Control, 2009, 54(2): 293-307.
- [6] Wang X, Wang L, Wu J H. Impacts of time delay on flocking dynamics of a two-agent flock model[J]. Communications in Nonlinear Science and Numerical Simulation, 2019, 70: 80-88.
- [7] Wang J L, Zhao H, Bi Y G, et al. An improved fast flocking algorithm with obstacle avoidance for multiagent dynamic systems[J]. Journal of Applied Mathematics,

2014, 15: 394-403.

- [8] Jiang C R, Du H B, Zhu W W, et al. Synchronization of nonlinear networked agents under event-triggered control[J]. Information Sciences, 2018, 459: 317-326.
- [9] Zhang Z C, Zuo Z Q, Wang Y J. Finite-time consensus of neutrally stable multi-agent systems in the presence of input saturation[J]. Journal of the Franklin Institute, 2019, 356(2): 894-907.
- [10] 陈世明, 邱昀, 刘俊恺, 等. 基于社区划分的多智能体网络快速蜂拥控制[J]. 控制与决策, 2018, 33(8): 1523-1526.
(Chen S M, Qiu Y, Liu J K, et al. Fast flocking algorithm of multi-agent network via community division[J]. Control and Decision, 2018, 33(8): 1523-1526.)
- [11] Gao J Y, Xu X, Ding N, et al. Flocking motion of multi-agent system by dynamic pinning control[J]. IET Control Theory & Applications, 2017, 11(5): 714-722.
- [12] Newman M. Networks[M]. New York: Oxford University Press, 2018: 159-176.
- [13] Kitsak M, Gallos L K, Havlin S, et al. Identification of influential spreaders in complex networks[J]. Nature Physics, 2010, 6(11): 888-893.
- [14] Liu J G, Ren Z M, Guo Q. Ranking the spreading influence in complex networks[J]. Physica A: Statistical Mechanics and its Applications, 2013, 392(18): 4154-4159.
- [15] Wang J Y, Hou X N, Li K Z, et al. A novel weight neighborhood centrality algorithm for identifying influential spreaders in complex networks[J]. Physica A: Statistical Mechanics and its Applications, 2017, 475: 88-105.

作者简介

何明(1978-), 男, 教授, 博士生导师, 从事互联网与无人化指控等研究, E-mail: paper_review@126.com;

许元云(1988-), 男, 硕士生, 从事无人机指控的研究, E-mail: 718996172@qq.com;

刘锦涛(1981-), 男, 博士, 工程师, 从事多智能体协同控制的研究, E-mail: top1944@163.com;

周波(1982-), 男, 博士, 从事复杂网络建模的研究;

丁晓晖(1976-), 男, 硕士, 从事自动化控制、复杂网络建模的研究.

(责任编辑: 郑晓蕾)