



FMM与改进GBNN模型相结合的多AUV实时围捕算法

陈铭治, 朱大奇

引用本文:

陈铭治, 朱大奇. FMM与改进GBNN模型相结合的多AUV实时围捕算法[J]. 控制与决策, 2020, 35(12): 2845–2854.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0393>

您可能感兴趣的其他文章

Articles you may be interested in

[融合长短时记忆机制的机械臂多场景快速运动规划](#)

Multi-scene rapid motion planning combining with long and short time memory mechanisms for manipulators

控制与决策. 2020, 35(12): 2968–2976 <https://doi.org/10.13195/j.kzyjc.2018.1387>

[基于改进萤火虫算法的区域交通信号配时优化](#)

Timing optimization of regional traffic signals based on improved firefly algorithm

控制与决策. 2020, 35(12): 2829–2834 <https://doi.org/10.13195/j.kzyjc.2019.1835>

[一种高匹配性的多层代价地图生成算法](#)

A high matching layered costmap generation algorithm

控制与决策. 2020, 35(12): 2883–2888 <https://doi.org/10.13195/j.kzyjc.2018.0721>

[战术级兵棋实体作战行动智能决策方法](#)

Intelligent decision-making method of tactical-level wargames

控制与决策. 2020, 35(12): 2977–2985 <https://doi.org/10.13195/j.kzyjc.2019.0504>

[可持续逆向物流网络设计研究进展及趋势](#)

Progress and prospects of sustainable reverse logistics network design

控制与决策. 2020, 35(11): 2561–2577 <https://doi.org/10.13195/j.kzyjc.2019.1175>

FMM与改进GBNN模型相结合的多AUV实时围捕算法

陈铭治¹, 朱大奇^{2†}

(1. 上海海事大学 商船学院, 上海 201306; 2. 上海海事大学 物流工程学院, 上海 201306)

摘要: 多自主水下机器人(AUV)实时围捕是一个综合的研究课题,包括联盟生成和目标追捕等阶段. 首先,基于快速行进算法(FMM)预估围捕时间,有效形成多AUV的动态围捕联盟;然后,在追捕阶段,AUV需要立即跟踪智能逃逸机器人以防止其逃跑. 为了实现这一目标,在GBNN(Glasius biological inspired neural network)模型中使用反比例函数替换指数函数计算神经元连接权值,加入额外的衰减项,并提出两点加快神经元活性传播的改进措施,使其适用于实时追捕路径规划. 仿真研究表明,围捕联盟形成机制和反比例权值GBNN模型实时路径规划策略都显示出其优越性. 在水下环境的多AUV协作围捕中,所提出的围捕控制算法可以提高围捕效率,减少AUV所花费的追捕距离和逃逸机器人的逃逸距离.

关键词: 多AUV协作; 实时围捕; 快速行进算法(FMM); GBNN

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0393

开放科学(资源服务)标识码(OSID):



引用格式: 陈铭治, 朱大奇. FMM与改进GBNN模型相结合的多AUV实时围捕算法[J]. 控制与决策, 2020, 35(12): 2845-2854.

Multi-AUV real-time hunting control based on FMM and improved GBNN model

CHEN Ming-zhi¹, ZHU Da-qi^{2†}

(1. Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China; 2. Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China)

Abstract: Multi-AUV real-time hunting is a comprehensive research topic, including team collaboration and pursuit of targets. In this paper, the hunting time for an evader is first estimated using a fast marching algorithm, and then a dynamic hunting alliance is formed between AUVs. Then, in the hunting phase, the AUV needs to track the evader immediately to prevent it from escaping. In order to achieve this goal, an additional attenuation term is added in the GBNN (Glasius biological inspired neural network) model, the inverse function instead of exponential function is also applied in calculating the connection weights of neurons, and two improvements to accelerate the propagation of neuronal activity are proposed, which make the GBNN model suitable for real-time path planning. Simulation studies show that the hunting alliance formation mechanism and the improved GBNN model both have their advantages. In the underwater multi-AUV cooperative hunting, the proposed hunting control algorithm improves the hunting efficiency, and reduces the hunting distance of the AUVs and the escaping distance of the evader.

Keywords: multi-AUV cooperation; real-time hunting; fast marching method; GBNN

0 引言

自主水下机器人(autonomous underwater vehicle, AUV)是一种可以在水下环境自主完成作业的机器人^[1],在深海勘测、水下设备维护、军事等领域得到了广泛的应用^[2-5].但是,单个AUV的作业能力通常有限,因此,多AUV协同工作系统受到广泛关注.多

AUV系统通过多个AUV之间的协调与合作,在多样性、健壮性、可靠性和效率等方面,比单个AUV有较大的优势.近年来,多AUV系统的研究越来越热^[6-7],包括水下协作搜索^[8-9]、编队控制^[10]、协作导航^[11-13]和实时围捕^[7,14-16]等方面的研究.其中,多个AUV实时围捕是一个有趣且综合的问题,包括两个阶段:

收稿日期: 2019-04-02; 修回日期: 2019-08-21.

基金项目: 国家自然科学基金项目(U1706224, 91748117, 51575336); 国家重点研发项目(2017YFC0306302); 上海市自然科学基金项目(19ZR1422600).

责任编委: 方勇纯.

[†]通讯作者. E-mail: zdq367@aliyun.com.

任务分配形成围捕联盟和实时路径规划直到成功完成围捕^[17-18]. 研究人员对实时围捕问题展开了广泛的研究. Wu等^[16,19]结合人工势场和虚拟包围圈进行围捕控制,提出了实现快速围捕的极限包围圈. Ishiwaka等^[20]提出了基于强化学习(RL)的围捕控制方法,可以优化围捕机器人的行动(速度和方向),以快速围捕逃逸机器人. 但在研究中,围捕机器人的速度可达逃逸机器人的3倍以上,围捕难度较低. 胡俊等^[21]提出了基于动态预测的围捕控制算法,实现了多机器人围捕控制. 但是,逃逸机器人在感知到围捕者后,为避免被围捕而会多次改变自身的运动方向,此时预测方法可能失效. Ni等^[13]使用生物启发神经网络(BNN)模型解决多机器人实时围捕控制问题,所提出的方法可以使机器人有效地完成实时围捕,但是,所有机器人在追踪阶段共享同样的神经元活性来追捕逃逸机器人,在水下环境的通信速率比较慢,很难实现.

上述研究都集中在二维(2-D)地面围捕的问题上,关于AUV在三维(3-D)水下环境的围捕研究相对较少. Nguyen等^[22]和Williams^[23]利用多个AUV寻找水雷,并提出了全覆盖方法. 由于围捕目标是静态的,没有智能,这种情况下的围捕难度降低. 王宏健等^[24]使用蚁群重规划方法和人工势场的避碰策略完成了水下机器人对目标的围捕,但是,该研究没有讨论围捕任务的分配问题,围捕团队构建方法有待解决. Huang等^[25]通过共享神经元活性,用生物启发神经网络(BNN)模型解决多AUV的合作围捕问题,他们提出了一种基于距离协商的方法来分配围捕任务,该算法可以处理多种情况,但有时AUV之间会遇到冲突. 为了克服这个冲突,文献[26-28]提出了一种位置预测方法,与人工势场法相比,该算法能够以较小的航行距离成功捕获逃逸机器人. 但是,该方法假定围捕AUV比逃逸机器人快很多(速度是逃逸机器人的3~6倍),大大降低了围捕的难度^[25-28]. 围捕是控制一组AUV包围逃逸机器人,AUV的协作非常重要. 多个AUV需要协调合作、共同努力提高围捕效率. 为促进AUV的协作并提高围捕效率,Chen等^[7]提出了基于时间竞争的围捕联盟生成方法. Ni等^[29]基于脊髓神经系统和遗传算法完成了水下目标围捕. 但是,联盟生成策略都没有考虑障碍物对围捕过程的影响,所形成的围捕联盟也可能不尽合理.

在实时围捕控制中,AUV必须能够跟踪快速移动的逃逸机器人,自行规划实时、高效的路径. 在快速变化的环境中,完成实时路径规划对于多AUV的围

捕控制至关重要. 研究人员提出了许多神经网络模型以有效地完成机器人的路径规划. Ageev等^[30]提出了基于多层和前馈神经网络的路径规划;Xia等^[31]提出了用于生成最短路径的递归神经网络. 虽然这两种算法对于路径规划都是有效的,但仅适用于静态环境. Ni等^[32]提出了生物启发式神经网络(BNN)模型,用于动态环境下的实时路径规划,并且已将BNN模型用于各种机器人的路径导航中;Glasius等^[33]提出了另一种生物启发式神经网络(GBNN)模型,用于机器人轨迹规划和避障. GBNN模型求解速度快,并且在大多数环境中对路径规划都很高效. 但据报道,该模型很难完成快速变化环境中的实时路径规划^[32]. 由于围捕过程中,逃逸机器人经常调整位置和方向,围捕过程属于快速变化环境下的路径规划. 为了使GBNN模型适用于快速变化环境下的路径规划,本文将反比例函数替换原来的GBNN模型中指数函数并提出改进措施,通过理论分析和仿真实验表明了改进后的模型适用于实时围捕控制.

本文重点研究水下环境多AUV实时围捕智能逃逸机器人的问题. 三维水下环境中的围捕与地面二维围捕相比,主要的区别和难点在于:1)水下环境难以实现大数据量的通信,许多围捕方法经过协商形成联盟并且在追踪阶段共享路径信息,这需要较大的数据量,在水下环境中很难实现;2)三维环境中如果不共享有效的追捕路径信息,则对于神经网络的路径规划方法,环境中有很多神经元,需要适当计算复杂度的路径规划算法以保证计算的实时性;3)三维环境中智能逃逸机器人的逃跑方向更多,更难以完成最终的围捕.

鉴于水下环境的特殊性,本文首先基于快速行进算法(fast marching method,FMM)生成围捕联盟. 该围捕联盟机制可估算障碍物对围捕过程的影响,有效地为多AUV分配围捕任务,形成围捕联盟. AUV之间不需要经过协商,只需获悉其他AUV和逃逸机器人的位置,通过FMM算法,预估最少时间的围捕团队,即可自组织地形成围捕联盟. 其次,提出一种反比例权值GBNN(Glasius biological inspired neural network)模型,并采用两点计算改进措施以适应动态的环境. 反比例权值GBNN模型在广阔的三维水下环境中具有较高的计算效率,并且不需要学习过程. 所有AUV之间不需要共享神经活性值,即每个AUV都有自己的神经活性地图,并自行规划路径. 而且,改进的模型具有良好的动态性能,能正确实时地到达逃逸机器人的围捕位置,有效地阻断它的逃跑方

向. 仿真研究表明,所提出的围捕任务分配和路径规划方法可以很好地应对智能逃逸机器人,提高围捕效率.

1 问题描述

本文应用基于离散化环境的栅格地图描述围捕问题,如图1所示. 黑色块状区域代表障碍物,而空白区域表示空闲空间. 问题表述为: n 个 AUV ($AUV_1, AUV_2, \dots, AUV_n$) 围捕 m 个逃逸机器人 ($Ev_1, Ev_2, \dots, Ev_m$). 当 AUV 紧密包围逃逸机器人,并在四周均匀分布时,围捕任务完成. 二维环境需要 4 个 AUV 才能完成对单个逃逸机器人围捕任务,而三维水下环境则需要 6 个 AUV. 如果有障碍物帮助完成围捕任务,则围捕 AUV 可以相应地减少,如图 1(b) 所示. 多 AUV 围捕智能逃逸机器人时,逃逸机器人会采取逃跑策略对抗 AUV 的围捕. 当逃逸机器人感知环境安全时,它采取随机方向的航行策略. 探测到围捕的 AUV 后,它将采取逃跑策略尽量逃跑. 在两种不同的情况下,需要采取各不相同的逃逸策略.

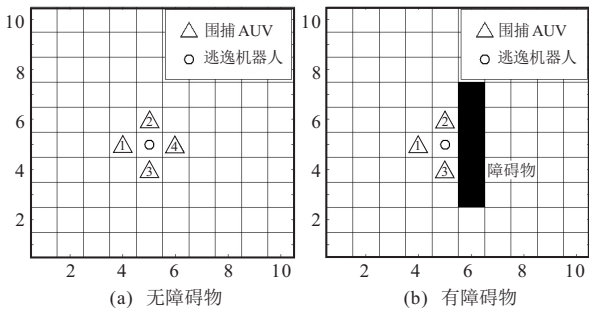


图1 二维环境下的成功围捕

1) 当 AUV 位于逃逸机器人的同一侧时,围捕包围圈还未形成,如图 2(a) 所示. 在这种情况下,逃逸机器人采取与 AUV 相反的方向运动以防被捕. 逃逸机器人通过改变目标点来改变运动方向,即

$$e_t = e_c + \sum_{i=1}^n (e_c - w_i) / r. \quad (1)$$

其中: e_t 是逃逸机器人的目标点的位置坐标,通过式 (1) 改变其目标点可使逃逸机器人朝着 AUV 的反方

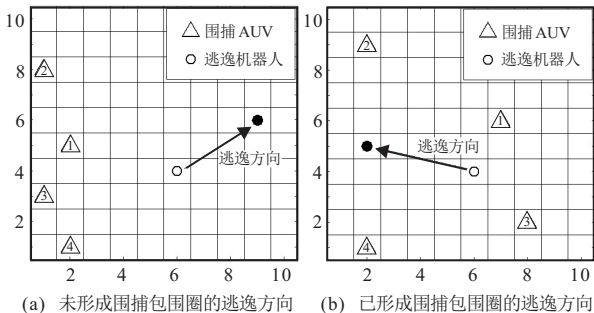


图2 二维环境下的逃逸策略

向运动; e_c 是逃逸机器人当前的位置坐标; n 是逃逸机器人感知范围内的 AUV 的数量; w_i 是第 i 个 AUV 的位置坐标. 为避免目标点的坐标值超过仿真环境的大小,引入 $r > 1$ 以减少逃逸机器人目标位置的剧烈变化,并将其控制在仿真环境中.

2) 如果多个 AUV 分布在逃逸机器人的 4 个方向,则 AUV 已经形成了围捕包围圈. 在这种情况下,逃逸机器人应该逃往距离最远的两个相邻 AUV 的中点,如图 2(b) 所示.

对于 3-D 环境,将 AUV 和逃逸机器人的位置投影到 3 个坐标平面 (X - Y 、 X - Z 和 Y - Z),而后检查多个 AUV 是否已经形成围捕包围圈. 如果其中一个坐标平面上没有形成包围圈,则三维环境的围捕包围圈未能形成. 同样地,如果还未形成包围圈,则逃逸机器人往 AUV 的相反方向逃跑;否则,朝着距离最远的两个相邻 AUV 之间的中点逃跑.

2 水下机器人的围捕策略

围捕联盟生成和实时路径规划是围捕需要解决的两个主要问题. 首先,障碍物会影响 AUV 的追捕过程;其次,逃逸机器人有一定的逃逸智能,快速规划围捕路径,防止他们逃跑也是至关重要的.

2.1 围捕联盟生成策略

本文提出的围捕联盟生成策略基于预估的围捕时间. 为了计及障碍物对追捕过程的影响,需要通过快速行进算法估算围捕时间. 快速行进算法使用栅格地图,通过求解 Eikonal 方程计算地图上每个栅格距离源点的到达时间. 预估到达时间时,假设障碍物栅格中 AUV 的速度为 0.0005 m/s ,而自由栅格中的速度为 2 m/s . 快速行进算法使用的 Eikonal 方程如下所示:

$$|\nabla T|F = 1, T_0 = 0. \quad (2)$$

其中: 初始位置的到达时间 T_0 为 0, F 为 AUV 的速度, ∇T 为栅格到达时间的增量. 计算围捕预估时间时, AUV 在每个栅格的速度仅取决于位置坐标,则该方程为非线性一阶偏导方程. Adalsteinsson 等^[34] 提出了迎风策略以求解该方程. 在二维环境中, ∇T 的数值近似解由下式给出:

$$\begin{aligned} & \max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \\ & \max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2 \approx |\nabla T_{ij}|^2 = \frac{1}{F_{ij}^2}, \end{aligned} \quad (3)$$

其中: D_{ij}^{-} 和 D_{ij}^{+} 是位置坐标 (i, j) 上的后向和前向差分.

用一阶差分近似 ∇T , 则式 (3) 可转换为

$$\sum_{k=1}^2 \max\left(\frac{T_{ij} - T_k}{\Delta_k}, 0\right)^2 = \frac{1}{F_{ij}^2}; \quad (4)$$

$$\Delta_1 = \Delta_x, \Delta_2 = \Delta_y; \quad (5)$$

$$\begin{cases} T_1 = \min(T_{i-1,j}, T_{i+1,j}), \\ T_2 = \min(T_{i,j-1}, T_{i,j+1}). \end{cases} \quad (6)$$

方程(4)中 T_{ij} 的解分为3种情况: 1) $T_{ij} > \max(T_1, T_2)$ 时, 由方程 $\sum_{k=1}^2 \left(\frac{T_{ij} - T_k}{\Delta_k}\right)^2 = \frac{1}{F_{ij}^2}$ 可求解得到 T_{ij} ; 2) $T_2 > T_{ij} > T_1$ 时, $T_{ij} = T_1 + \Delta_1/F$; 3) $T_1 > T_{ij} > T_2$ 时, $T_{ij} = T_2 + \Delta_2/F$. 位置坐标 (i, j) 的到达时间 T_{ij} 取决于具有较小到达时间值的邻域栅格. 每个栅格点做如下标记:

1) 已知点 (known): 该位置计算出的到达时间将保持不变;

2) 窄带点 (narrow band): 该位置的到达时间稍后可能改变;

3) 远点 (far): 该位置的到达时间尚未计算.

FMM 算法求解过程中, 首先将机器人所在的初始位置标记为已知点 (known), 将已知点 (known) 的邻域栅格标记为窄带点 (narrow band), 算法计算过程如下.

step 1: LOOP: 在所有窄带点中, 提取具有最短到达时间的栅格点并标记为已知点;

step 2: 找到它的邻域点 (远点或者窄带点);

step 3: 求解方程(4), 更新邻域点的到达时间;

step 4: 回到 LOOP 循环执行, 直至所有栅格点的到达时间都为已知.

利用快速行进算法计算栅格地图上所有栅格点到达某个逃逸机器人的时间后, 通过每个 AUV 占领的栅格位置坐标, 可以获取每个 AUV 到达这个逃逸机器人的预估时间. 将每个 AUV 到达逃逸机器人的预估时间编制为矩阵 T_{we} , T_{we} 的第 i 行是第 i 个 AUV 通过快速行进算法预估的到达各个逃逸机器人的时间. 利用 T_{we} , 下面的计算步骤给出了形成多 AUV 的围捕联盟的策略.

算法的输入参数为: T_{we} (获取的预估围捕时间矩阵); TeamNum (AUV 所能形成的最大围捕团队个数); Num (2-D 围捕: Num = 4, 3-D 围捕: Num = 6); NumEvader (逃逸机器人的个数).

输出为: evIndex (逃逸机器人的围捕顺序, 是行向量); auvIndex (对各个逃逸机器人形成的围捕联盟的 AUV 编号, 是矩阵).

step 1: 初始化 $j = 1$, evIndex = $1 \times \text{TeamNum}$ 的零向量, auvIndex = $\text{TeamNum} \times \text{Num}$ 的零矩阵;

step 2: LOOP: 对 T_{we} 排序, 以便对前 Num 行求和, 即 $[\text{stT}, \text{winners}] = \text{Sort}(T_{we})$;

step 3: 对每个逃逸机器人, 获取 stT 前 Num 行, 即 For $i = 1$ to NumEvader, $\text{tmSdT}(:, i) = \text{stT}(1: \text{Num}, i)$;

step 4: 对 tmSdT 求和并排序, 有 $\text{tmTotTime} = \text{sum}(\text{tmSdT}), [:, \text{winnerIndex}] = \text{sort}(\text{tmTotTime})$;

step 5: 获取最容易捕获的逃逸机器人编号, 即 $\text{evIndex}(1, j) = \text{winnerIndex}(1, 1)$;

step 6: 分配尚无追捕任务且能以最小预估时间到达该逃逸机器人的 AUV 追捕该逃逸机器人, 有 $\text{auvIndex}(j, :) = \text{winners}(1: \text{Num}, \text{evIndex}(1, j))$;

step 7: 设置逃逸机器人到达时间为较大的正值, 以防止其再次竞争获胜, 即 $T_{we}(:, \text{evIndex}(1, j)) = \text{colBigValue}$;

step 8: j 增加 1, 如果 $j \leq \text{NumEvader} \& j \leq \text{TeamNum}$, 则回到 LOOP 循环执行.

AUV 首先围捕具有最小的预估围捕时间的逃逸机器人, 对该机器人形成 AUV 围捕联盟. 随着环境的变化, 时间最短的 AUV 团队是变动的, 因此联盟是动态的. 许多联盟生成策略只考虑 AUV 和逃逸机器人的位置和距离信息, 首先围捕距离最近的逃逸机器人, 而非分析围捕所需的时间. 障碍物可能阻碍 AUV 的行进, 仅由位置信息形成的基于距离的围捕联盟可能是低效的. 在本文联盟生成机制中, 使用快速行进算法估算围捕时间, 考虑了障碍物因素的影响, 这使得围捕完成速度更快, AUV 追捕的距离和逃逸机器人逃跑的距离都将变短.

2.2 实时围捕路径规划

本文改进了 GBNN 模型, 使其可以用于实时规划动态环境中的路径.

2.2.1 回顾 GBNN 模型

Glasius 等^[33] 提出了 Glasius 生物启发神经网络 (GBNN) 模型, 即

$$x_i(t+1) = f\left(\sum_{j \in S_i} (w_{ij} \cdot x_j(t)) + I_i\right). \quad (7)$$

$$w_{ij} = \begin{cases} e^{-\gamma \cdot \text{dist}(p_i - p_j)}, & \text{dist}(p_i - p_j) \leq R; \\ 0, & \text{dist}(p_i - p_j) > R. \end{cases} \quad (8)$$

$$I_i = \begin{cases} v, & i \text{ 是目标点}; \\ -v, & i \text{ 是障碍物}; \\ 0, & i \text{ 是空白区域}. \end{cases} \quad (9)$$

$$f(x) = \begin{cases} 1, & x \geq 1; \\ \beta \cdot x, & 0 < x < 1; \\ 0, & x \leq 0. \end{cases} \quad (10)$$

其中: $x_i(t+1)$ 是当前时间步的第 i 个栅格的神经活性值, $x_j(t)$ 是当前时间步往前一步的邻域栅格 j 的神经活性值, w_{ij} 是对称的神经元连接权重, $\text{dist}(p_i - p_j)$ 是从神经元 i 到 j 的欧氏距离, R 是如图3所示的接受域半径, 外部输入 I_i 表示关于目标点、自由空间和障碍物栅格的信息. 其他AUV、障碍物或逃逸机器人占据的栅格均属于当前AUV的障碍物, 当前AUV都难以通过. 式(10)的 $f(x)$ 是分段线性传递函数.

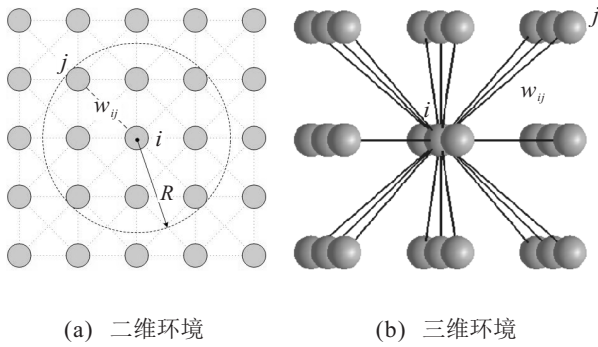


图3 GBNN模型

水下工作环境离散化后, 图3(a)和图3(b)显示了二维和三维环境的神经网络模型. 环境中每个栅格都设置一个神经元, 神经元与其邻域神经元相连, 其接受域是一个半径为 R 的圆形区域. 在GBNN模型中, 障碍物只有局部的作用, 但目标点的神经活性在整个环境中传播.

GBNN模型有两个参数 γ 和 β . 较小的 γ 可以使连接权重较大, 有利于目标神经活性值的快速传播; 相反, 较大的 β 可以加速目标神经活性值的传播. 然而, 参数 γ 和 β 的选取需要确保除目标神经元以外的神经元的活性都小于1. 有了这个限制, γ 有一个最小的阈值, 而 β 有一个最大的阈值. 对于二维环境, 参数的约束可由下式描述:

$$\begin{cases} 4\beta(e^{-\gamma} + e^{-2\gamma}) < 1, \\ 4(e^{-\gamma} + e^{-2\gamma}) < 1. \end{cases} \quad (11)$$

如果 $\beta = 1$, 式(11)的第1个不等式和第2个不等式相同, 则 $\gamma > (\ln(2\sqrt{2} + 2) \approx 1.5745)$. GBNN模型的参数可以根据环境维度固定. 由于参数有限, GBNN模型的时间衰减率并不理想. 在时间维度上的衰减率等于1, 这使得神经活动的变化可能滞后于快速变化的环境. 在空间维度上, GBNN的神经活性的传播需要许多时间步才能从目标点到达AUV的位置. 实时路径规划要求神经活性在一个时间步从目标点的位置传播到整个环境. 由于难以避免时间步延迟, GBNN模型不适用于实时路径规划.

2.2.2 改进的GBNN模型

为了使GBNN模型具有良好的动态环境下的实时路径规划性能, 改进GBNN模型, 在时间维度上实现更高的衰减率, 即

$$w_{ij} = \begin{cases} \frac{1}{\gamma \cdot \text{dist}(p_i - p_j)^2}, & \text{dist}(p_i - p_j) \leq R; \\ 0, & \text{dist}(p_i - p_j) > R. \end{cases} \quad (12)$$

$$x_i(t+1) = f\left(\sum_{j \in S_i} (w_{ij} \cdot x_j(t)) + I_i - \frac{1}{\gamma} x_i(t)\right). \quad (13)$$

改进的GBNN模型由式(13)表示, 与式(7)对比, 由于加入了额外的衰减项以获得更好的时间维度上的衰减率, 从而提高了神经元活性的更新速率. 加入衰减项前, GBNN在时间维度上的衰减率为1; 而加入衰减项后, 改进的GBNN在时间维度上的衰减率为 $1 + 1/\gamma$. 一些与指数函数具有类似属性的函数也可以用于计算神经元的连接权重. 如式(12)所示, 采用反比例函数替代指数函数. 反比例函数值域的变化快于指数函数. 此外, 如图4所示, 反比例函数具有比指数函数更高的值域. 反比例函数的值域在 $(0, \infty)$ 中, 而指数函数的值在 $(0, 1)$ 中. 反比例函数可以使神经元的连接更强, 适合于更大尺寸的环境.

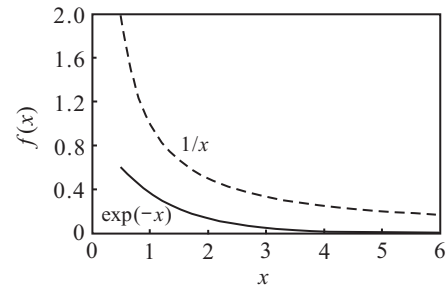


图4 反比例和指数函数对比

反比例权重GBNN模型用于二维路径规划时, 也要满足除目标神经元以外的神经元的活性都小于1, 参数的约束条件为

$$\begin{cases} \beta\left(3 \cdot \frac{1}{\gamma} + 4 \cdot \frac{1}{2\gamma}\right) < 1, \\ 3 \cdot \frac{1}{\gamma} + 4 \cdot \frac{1}{2\gamma} < 1. \end{cases} \quad (14)$$

如果 $\beta = 1$, 则 $\gamma > 5$. 本文所提出的具有额外衰减项的改进GBNN模型也可以针对特定应用进行参数固定. 此外, 提出如下两点活性值计算上的改进措施.

1) 在实时路径规划的要求下, 在空间维度上神经活性必须在环境中快速传播. 神经活性计算方向不明确的情况下, 由于初始时只有目标神经元的活性

值为1而其他神经元活性值全为0,神经元的活性在空间维度上传播需要把目标神经元的神经活性往外扩散,需要花费较多的延迟等待时间.确定神经元的计算方向,从目标神经元层层向外计算GBNN模型的神经活性值,可以在一个时间步传播目标神经元活性到整个环境中.即先计算目标神经元的邻域神经元的活性值,再计算目标点的邻域神经元的邻域栅格的神经活性值,层层扩散.

2) 另一方面,当目标在快速移动时,神经网络的神经元活性的时间衰减率可能低于环境变化,就会包含不正确的路径信息.当目标或障碍物快速移动时,所有神经元在下一个时间步被抑制,并且从目标点的位置开始向外重新计算所有栅格的神经活性值.

2.2.3 路径规划

通过改进的GBNN模型,在规划路径时AUV航行到邻域神经元中活性最大的栅格,即

$$\begin{aligned} \text{Path} &= P_n | x_{P_n} = \max\{x_i, i = 1, 2, \dots, k\}, \\ P_p &= P_c, P_c = P_n. \end{aligned} \quad (15)$$

其中: x_i 是AUV当前位置附近的栅格神经元活性值;对于二维环境, $k = 8$,而在三维环境中, $k = 26$; P_p 、 P_c 和 P_n 是AUV的前一个、当前和下一个位置.

本文提出的改进的反比例权值GBNN具有固定的计算方向,加快了传播速度,消除了路径规划的时延.额外的衰减项目也改善了GBNN模型在时间维度上的动态特性.即使逃逸机器人快速移动,该路径规划方法也可以确保障碍物的神经元活性小于空闲栅格,并且根据变化的环境快速地改变神经活性值.反比例GBNN模型在参数设定满足要求的情况下收敛,且AUV能够安全准确地到达围捕占位点,可适用于逃逸机器人方向和位置上快速变动的多AUV围捕实时追踪.

3 仿真研究

为了研究本文所提出的围捕控制算法的可行性,通过Matlab进行仿真实验.首先开展二维环境下围捕控制的实验.将围捕结果与BNN路径规划模型和基于位置信息的围捕联盟策略等算法进行比较.然后,通过实现三维环境下的合作围捕实验,验证算法在水下环境下围捕智能逃逸机器人的有效性.用于运行仿真程序的计算机配置为Intel(R)Core TM i7-6700HQ CPU @ 2.6 GHz和16 G内存,安装有Windows 10系统.为了考虑AUV和逃逸机器人在实际情况下的形状和动态特性,对障碍物放大两个栅格大小.在仿真结果图中,将障碍物栅格标记为灰色.

3.1 二维环境下的围捕仿真

本节考虑二维环境下的围捕.由8个AUV围捕2个逃逸机器人.设计的围捕环境包含 60×60 个栅格,每个栅格的边长为100 m,环境大小 $6000 \times 6000 \text{ m}^2$.逃逸机器人的感知范围是一个半径为5个栅格的圆形区域.当逃逸机器人检测到附近的围捕AUV后,采取逃逸策略.所有AUV的速度是逃逸机器人的两倍.围捕首先通过所提出的基于快速行进算法的联盟生成和反比例权值GBNN模型实现.然后,与基于位置信息的围捕联盟生成算法和BNN模型的实验结果进行对比,验证了所提出的围捕联盟生成算法能够加快围捕速度,提高围捕效率,并且反比例权值GBNN模型在实时围捕控制中具有良好动态性能.反比例权值GBNN模型计算复杂度低于BNN模型,所有AUV之间不需要共享相同的神经元活性.事实上,采用GBNN算法路径规划,每个AUV都可以由自己的独立神经元活性来完成路径规划.

3.1.1 所提出的围捕控制算法的围捕结果

仿真程序随机生成两个逃逸机器人 Ev_1 和 Ev_2 ,位于栅格(30, 15)和(35, 45),并且它们以随机方向移动.8个AUV在边界上,准备围捕逃逸机器人.在2D环境中,神经元要满足侧向连接的需要, $R = \sqrt{2}$,而 v 设置为一个远大于1的数值即可. β 和 γ 是两个对性能影响较大的参数,且受到不等式(14)的限制, β 的值可直接设为1,如前所述 $\gamma > 5$. γ 越小,神经元活性传播的范越围广,速度越快.因此 γ 的值设置为略大于5,满足除目标神经元活性为1、其他神经元活性小于1的收敛性要求.综上所述,改进的GBNN模型的参数设置为 $\beta = 1, v = 200, \gamma = 5.01, R = \sqrt{2}$.围捕过程如图5所示.在图5中,黑色块状区域表示障碍物,“+”、“▽”、“*”、“×”、“□”、“◇”、“△”、“★”分别表示AUV₁、AUV₂、AUV₃、AUV₄、AUV₅、AUV₆、AUV₇、AUV₈.逃逸机器人用圆圈表示.围捕的时间步随每个逃逸机器人的移动步数增加.

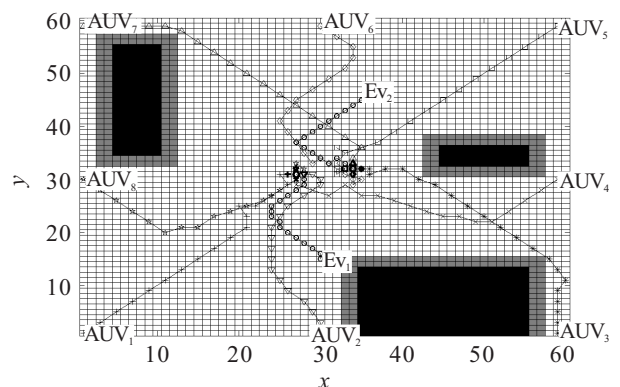


图5 采用所提出的围捕控制算法的2-D环境围捕结果

8个AUV被分成两个围捕联盟. 其中: 由AUV₁、AUV₂、AUV₄和AUV₈组成的围捕团队完成Ev₁的围捕任务; 剩下的AUV组成另一个围捕联盟, 完成Ev₂的围捕. 在围捕过程中, 逃逸机器人试图改变方向逃跑以对抗AUV的围捕. AUV₇在单元(3, 59)处遇到障碍物, 采用反比例权值GBNN模型安全避开障碍物, 并且所有的AUV可以跟随目标点的变化, 快速靠近他们的围捕占位点来完成围捕. Ev₁和Ev₂分别在单元(27, 31)和(34, 32)处被成功捕获, 程序运行了37个围捕时间步. 表1的第2列列出了采用本文提出的围捕控制算法完成围捕任务所花费的AUV的距离成本、逃逸机器人的逃跑距离和仿真时间.

表1 3种算法完成围捕的结果对比

	所提出的算法	BNN模型	位置联盟
AUV ₁ 的距离/m	4 341	4 583	4 859
AUV ₂ 的距离/m	3 532	4 562	3 979
AUV ₃ 的距离/m	4 581	5 822	4 018
AUV ₄ 的距离/m	3 962	4 806	4 262
AUV ₅ 的距离/m	4 548	5 601	5 013
AUV ₆ 的距离/m	4 310	4 357	4 644
AUV ₇ 的距离/m	4 658	5 971	4 583
AUV ₈ 的距离/m	3 822	4 199	4 353
AUV平均距离/m	4 219	4 988	4 464
Ev ₁ 的距离/m	2 255	2 904	2 297
Ev ₂ 的距离/m	2 454	2 638	2 456
仿真时间/s	7.7	1 093.5	9.2

3.1.2 BNN模型路径规划方法的围捕结果

在相同的初始环境设置下, 采用与本文相同的基于快速行进算法围捕联盟生成算法, 并应用BNN模型进行路径规划. BNN模型的参数设置为 $B = D = 1, A = 25, u = 1, R = 2$. 图6显示了完成的围捕结果. AUV组建了相同的围捕联盟, 但整个围捕过程花费了更多的围捕时间步和更多的运行时间. Ev₁和

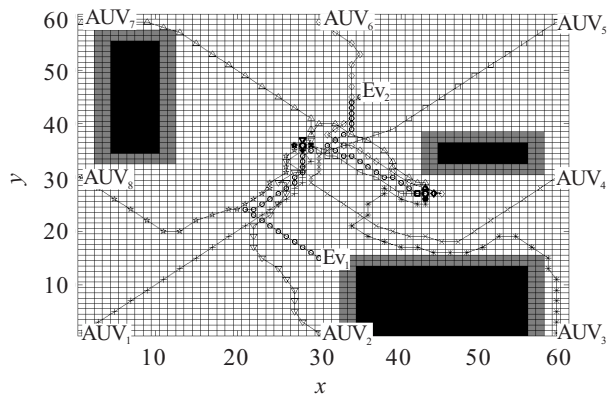


图6 采用BNN模型的2-D环境围捕结果

Ev₂分别在(28, 36)和(43, 27)被成功围捕, 程序运行50个围捕时间步. 由于BNN算法的动态性能略微滞后于快速移动的目标点, 规划的路径含有错误信息, 使得规划的路径变长, 逃逸机器人逃跑得更远. 表1的第3列列出了使用BNN模型完成围捕时AUV花费的距离成本、逃逸者的逃跑距离和仿真运行时间.

使用改进的GBNN模型完成围捕, AUV花费的平均距离为4 219 m, 而BNN模型的平均距离为4 988 m. 此前, BNN模型已被报道适用于动态环境下的实时路径规划. 然而, 反比例权值GBNN模型为AUV提供了更直接的方向和更短的路径. GBNN模型的技术改进并提高了其动态性能, 可以提供正确的路径信息. 更重要的是, 反比例权值GBNN模型不需要太多时间, 整个仿真过程只需要7.7 s的运行时间. BNN模型在每个时间步为环境的每个神经元求解微分方程, 需要1 093.5 s. 仿真中, 除了计算神经活性子程序不同外, 其他程序都是相同的. 由此可知BNN模型非常耗时. 在三维水下环境中存在许多神经元可以用来构建环境, 因此, 使用BNN模型完成路径规划, 其耗时将是难以想象的.

3.1.3 基于位置的围捕联盟生成算法的围捕结果

在文献[7, 25-28]中, 围捕联盟的生成都基于AUV和逃逸机器人的位置信息. 在基于位置信息的策略中, AUV以位置信息计算与逃逸机器人的欧氏距离, 根据邻域规则形成围捕联盟. 图7显示了相同环境下基于位置信息的联盟形成方法的围捕结果.

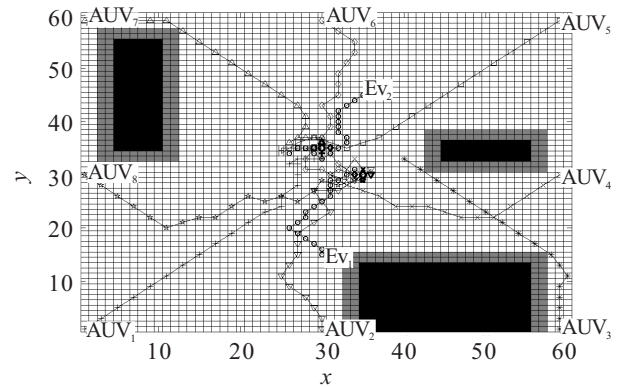


图7 采用位置信息联盟生成的2-D环境围捕结果

由于该策略未考虑障碍物对AUV行进过程的影响, AUV构建了不同的围捕团队. AUV₂、AUV₄、AUV₆和AUV₈组成联盟围捕Ev₁, 其余的AUV围捕Ev₂. Ev₁在第36围捕时间步于(35, 29)处被捕获, 之后AUV₆加入对Ev₂的围捕而AUV₃则停止追捕. Ev₂在(30, 35)处被成功围捕. 该程序运行了40个围捕时间步. 由于围捕联盟只依赖于AUV和逃逸者的位置信息, 在围捕过程中障碍物阻碍了一些AUV的前进,

而这一因素在分配围捕任务时并未考虑,从而导致逃逸者逃跑了更远的距离.表1的第4列列出了基于位置信息的围捕联盟生成算法完成围捕时,AUV花费的距离成本、逃逸者的逃跑距离和仿真运行时间.

采用本文所提出的基于快速行进算法的联盟生成机制,AUV以平均4 219 m的距离完成围捕;而基于位置信息的机制需要4 464 m.基于快速行进算法的联盟生成机制在分配任务时提前考虑了障碍物的影响,因此更加高效,更适合于存在障碍物环境下形成多个AUV的围捕联盟.

3.2 2-D环境下的蒙特卡洛围捕仿真

考虑与3.1节一样的仿真环境情况下,随机生成逃逸机器人的位置和运动方向,进行10次蒙特卡洛实验,分别使用本文所提出的围捕方法以及3.1节提到的BNN模型、位置距离信息联盟方法解决围捕问题.10次仿真实验后,计算3种不同算法下,围捕过程中AUV和逃逸机器人的平均距离以及平均仿真时间,如表2所示.

表2 3种算法完成围捕的平均数值对比

	所提出的算法	BNN模型	位置联盟
AUV ₁ 的平均距离/m	4 356	5 690	4 591
AUV ₂ 的平均距离/m	3 352	5 207	4 354
AUV ₃ 的平均距离/m	2 606	5 262	3 217
AUV ₄ 的平均距离/m	4 277	4 479	4 761
AUV ₅ 的平均距离/m	5 431	5 083	5 336
AUV ₆ 的平均距离/m	4 448	5 046	5 144
AUV ₇ 的平均距离/m	4 123	6 052	4 315
AUV ₈ 的平均距离/m	4 364	6 033	5 022
AUV平均距离/m	4 120	5 357	4 593
Ev ₁ 的平均距离/m	2 182	3 697	2 419
Ev ₂ 的平均距离/m	2 399	2 993	2 795
平均仿真时间/s	8.6	1 288.1	10.3

使用本文所提出的方法,在10次实验中,AUV平均距离为4 120 m;而位置信息联盟方法需要花费4 593 m,略高于本文提出的围捕联盟方法.在10次实验中,本文提出的围捕联盟方法有9次优于位置信息联盟方法,有1次实验所产生的围捕联盟劣于位置信息联盟方法,花费了略多的围捕距离.当环境中没有障碍物或者障碍物对路径规划影响较小时,本文的围捕联盟方法与位置信息联盟方法结果基本一致;而当障碍物较大时,较多地阻碍了AUV的围捕前行方向,本文所提出的联盟生成算法表现明显优于位置信息联盟方法.实验中,使用BNN模型的路径规划方法

则均劣于改进的GBNN方法,在围捕过程中,需要平均距离为5 357 m.此外,BNN模型的仿真实验平均完成1次实验需要1 288.1 s,远高于其他两种算法,高计算量可能使其无法胜任围捕所需的实时路径规划.

此外,针对2-D环境存在数目较多且分散的障碍物,也开展了10次随机的多AUV围捕仿真,以验证算法在不同类型的环境下具有普适性.本文所提出的算法每个AUV需要花费平均距离为4 233 m,而基于位置信息联盟方法AUV花费平均距离为4 368 m,BNN算法AUV平均花费距离为4 812 m.可见,本文算法在分散的多障碍物环境下也能提高围捕效率,加快完成围捕任务.

3.3 3-D环境的围捕

水下环境是一个三维环境,AUV和逃逸机器人有更多的运动方向.本节设计了三维环境下的围捕实验.围捕AUV位于仿真环境的8个角落,而逃逸机器人最初位于栅格(15,28,29).在实验中,AUV的速度是逃逸机器人的两倍.在三维环境中应用改进的反比例权值GBNN模型,为满足收敛和路径规划的条件,除目标神经元以外的其他神经元活性均小于1.3-D环境中,为满足神经元侧向连接的需要, $R = \sqrt{3}$. γ 和 β 受如下不等式方程的约束:

$$\begin{cases} \left(6 \cdot \frac{1}{2\gamma} + 12 \cdot \frac{1}{3\gamma} + 5 \cdot \frac{1}{\gamma}\right) \cdot \beta < 1, \\ 6 \cdot \frac{1}{2\gamma} + 12 \cdot \frac{1}{3\gamma} + 5 \cdot \frac{1}{\gamma} < 1. \end{cases} \quad (16)$$

若 $\beta = 1$,则 $\gamma > 12$.为更广并更快地传播神经元活性, γ 应尽可能小.因此,参数设置为 $\beta = 1, v = 200, \gamma = 12.000\ 1, R = \sqrt{3}$.

图8显示了三维环境中的围捕过程.围捕任务被分配到1、2、3、5、6和7号AUV.通过本文提出的改进的GBNN模型规划实时路径,在围捕过程中,AUV₁能够防止与障碍物发生碰撞.并且,只要逃逸机器人改变方向,围捕团队中的所有AUV都会跟随逃逸机器人改变方向,以尽快完成围捕.逃逸机器人在栅

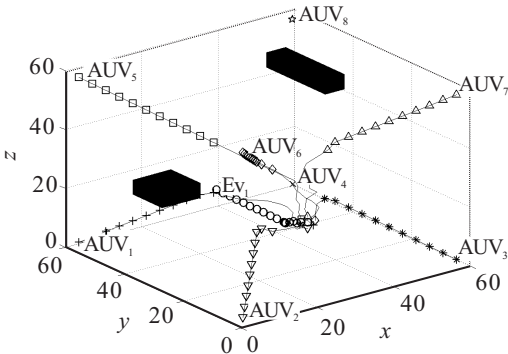


图8 采用所提出的围捕算法的3-D环境围捕结果

格单元(32, 19, 16)被6个AUV包围,它的逃跑距离为2961 m. AUV₁、AUV₂、AUV₃、AUV₅、AUV₆和AUV₇的追捕距离分别为6323、4499、4480、6816、6303和6201 m. 围捕Ev₁总共需要22个围捕时间步. 在每个时间步中,本文所提出的GBNN模型能够快速完成神经元活性的计算,实时规划路径并完成围捕,程序总共运行577.1 s.

在三维水下环境中,由于通信受限,AUV几乎不可能共享路径信息. 另外,三维环境的神经元数量很多,随着环境尺度增大按三次方增长,因此,本文所提出的反比例GBNN算法比BNN路径规划方法对每个神经元的活性计算效率高很多. 实验中更新一次环境(环境大小:6000 × 6000 × 6000 m,神经元个数为216000)中的所有神经元活性需要约4~5 s的时间,而BNN方法经测试需要超过30 s的时间. 对于水下机器人,其运行速度通常较慢(最快速度约4 m/s),改进的GBNN算法所需要的神经活性的更新时间完全可以满足规划路径的需要,AUV无需共享路径信息便可以直接自主规划路径. 而BNN算法计算速度太慢,难以满足路径规划的实时性要求. 基于快速行进算法的预估时间联盟生成机制和改进的反比例权值GBNN模型的策略适用于多AUV的实时协同围捕控制.

4 结 论

多AUV实时围捕智能逃逸机器人需要AUV互相协作,努力提高围捕效率. 本文针对多个AUV实时围捕智能逃逸机器人问题的联盟生成和实时路径规划进行了研究,提出了基于快速行进算法得出的预估围捕时间的联盟生成机制来构建高效的动态围捕联盟. 针对快速变化的环境,对GBNN模型进行了改进,使其适用于实时路径规划. 理论分析和仿真实验表明了该算法的有效性. 将所提出的策略与原始的基于位置信息的联盟生成算法和BNN模型进行了比较. 结果表明,本文策略更加有效,减少了围捕所需的时间,并减少了AUV花费的追捕距离和逃逸者的逃跑距离. 另外,该算法还同样适用于三维水下环境中的围捕.

参考文献(References)

- [1] 徐玉如, 庞永杰, 甘永, 等. 智能水下机器人技术展望[J]. 智能系统学报, 2006, 1(1): 9-16.
(Xu Y R, Pang Y J, Gan Y, et al. AUV state-of-the-art and prospect[J]. CAAI Transactions on Intelligent Systems, 2006, 1(1): 9-16.)
- [2] Krieg M, Mohseni K. Dynamic modeling and control of biologically inspired vortex ring thrusters for underwater robot locomotion[J]. IEEE Transactions on Robotics, 2010, 26(3): 542-554.
- [3] Blidberg D R. The development of autonomous underwater vehicles (AUV): A brief summary[C]. IEEE International Conference on Robotics and Automation. Seoul, 2001: 1-12.
- [4] 张荣敏, 陈原, 高军. 无鳍舵矢量推进水下机器人纵向稳定性研究[J]. 哈尔滨工程大学学报, 2017, 38(1): 133-139.
(Zhang R M, Chen Y, Gao J. Longitudinal handling stability of vectored thrust underwater vehicle without fin and rudder[J]. Journal of Harbin Engineering University, 2017, 38(1): 133-139.)
- [5] Joung T H, Lee J H, Nho I S, et al. A study on the design and manufacturing of a deep-sea unmanned underwater vehicle based on structural reliability analysis[J]. Ships and Offshore Structures, 2009, 4(1): 19-29.
- [6] 徐博, 邱立民, 杨建. 多AUV协同导航时间延迟误差机理分析与补偿算法[J]. 控制与决策, 2015, 30(1): 9-16.
(Xu B, Qiu L M, Yang J. Analysis of time delay and error compensation for multi-AUVs' cooperative navigation approach[J]. Control and Decision, 2015, 30(1): 9-16.)
- [7] Chen M Z, Zhu D Q. A novel cooperative hunting algorithm for inhomogeneous multiple autonomous underwater vehicles[J]. IEEE Access, 2018, 6(99): 7818-7828.
- [8] 李晔, 姜言清, 张国成, 等. 一种基于电子海图的欠驱动AUV区域搜索方案[J]. 机器人, 2014, 36(5): 609-618.
(Li Y, Jiang Y Q, Zhang G C, et al. An underactuated AUV-oriented region search method based on electronic chart[J]. Robot, 2014, 36(5): 609-618.)
- [9] Cao X, Zhu D Q, Yang S X. Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(11): 2364-2374.
- [10] 刘陆, 王丹, 彭周华. 基于PNDSC的欠驱动AUV编队控制器设计[J]. 控制与决策, 2015, 30(12): 2241-2246.
(Liu L, Wang D, Peng Z H. Formation controller design based on PNDSC for underactuated AUV[J]. Control and Decision, 2015, 30(12): 2241-2246.)
- [11] 李闻白, 刘明雍, 李虎雄, 等. 基于单领航者相对位置测量的多AUV协同导航系统定位性能分析[J]. 自动化学报, 2011, 37(6): 724-736.
(Li W B, Liu M Y, Li H X, et al. Localization performance analysis of cooperative navigation system for multiple AUVs based on relative position measurements with a single leader[J]. Acta Automatica Sinica, 2011, 37(6): 724-736.)
- [12] Paull L, Saeedi S, Seto M, et al. AUV navigation and localization: A review[J]. IEEE Journal of Oceanic Engineering, 2014, 39(1): 131-149.
- [13] Ni J J, Yang S X. Bioinspired neural network for real-time cooperative hunting by multirobots in unknown

- environments[J]. IEEE Transactions on Neural Networks, 2011, 22(12): 2062-2077.
- [14] Li J, Pan Q S, Hong B R, et al. Multi-robot cooperative pursuit based on association rule data mining[C]. International Conference on Fuzzy Systems and Knowledge Discovery. Tianjin: IEEE, 2009: 303-308.
- [15] 付勇, 汪浩杰. 一种多机器人围捕策略[J]. 华中科技大学学报: 自然科学版, 2008, 36(2): 26-29.
(Fu Y, Wang H J. A new capture strategy of multi-robots[J]. Journal of Huazhong University of Science & Technology: Nature Science Edition, 2008, 36(2): 26-29.)
- [16] Wu M, Huang F, Wang L, et al. A distributed multi-robot cooperative hunting algorithm based on limit-cycle[C]. International Asia Conference on Informatics in Control, Automation and Robotics. Bangkok: IEEE, 2010: 156-160.
- [17] Korf R E. A simple solution to pursuit games[C]. Working Papers of the Eleventh International Workshop on DAI. Glenn Arbor, 1992: 195-213.
- [18] Li J, Pan Q S, Hong B R. A new approach of multi-robot cooperative pursuit based on association rule data mining[J]. International Journal of Advanced Robotic Systems, 2009, 6(4): 329-336.
- [19] 伍明, 孙继银. 基于极限环的多机器人协作围捕算法[J]. 微计算机信息, 2010, 26(20): 13-15.
(Wu M, Sun J Y. A distributed multi-robot cooperative hunting algorithm based on limit-cycle[J]. Microcomputer Information, 2010, 26(20): 13-15.)
- [20] Ishiwaka Y, Sato T, Kakazu Y. An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning[J]. Robotics and Autonomous Systems, 2003, 43(4): 245-256.
- [21] 胡俊, 朱庆保. 基于动态预测目标轨迹和围捕点的多机器人围捕算法[J]. 电子学报, 2011, 39(11): 2480-2485.
(Hu J, Zhu Q B. A multi-robot hunting algorithm based on dynamic prediction for trajectory of the moving target and hunting points[J]. Acta Electronica Sinica, 2011, 39(11): 2480-2485.)
- [22] Nguyen B, Hopkin D. Modeling autonomous underwater vehicle (AUV) operations in mine hunting[C]. Oceans 2005-Europe. Brest: IEEE, 2005: 533-538.
- [23] Williams D P. On optimal AUV track-spacing for underwater mine detection[C]. IEEE International Conference on Robotics and Automation. Anchorage: IEEE, 2010: 4755-4762.
- [24] 王宏健, 熊伟, 陈子印, 等. 多自主水下航行器区域搜索与协同围捕方法研究[J]. 中国造船, 2010, 51(2): 117-125.
(Wang H J, Xiong W, Chen Z Y, et al. Research on methods of region searching and cooperative hunting for autonomous underwater vehicles[J]. Shipbuilding of China, 2010, 51(2): 117-125.)
- [25] Huang Z R, Zhu D Q. A cooperative hunting algorithm of multi-AUV in 3-D dynamic environment[C]. The 27th Chinese Control and Decision Conference. Qingdao: IEEE, 2015: 2571-2575.
- [26] Cao X, Huang Z, Zhu D. AUV cooperative hunting algorithm based on bio-inspired neural network for path conflict state[C]. IEEE International Conference on Information and Automation. Lijiang: IEEE, 2015: 1821-1826.
- [27] Lv R, Gan W, Sun B, et al. A multi-AUV hunting algorithm with ocean current effect[C]. IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems. Shenyang: IEEE, 2015: 869-874.
- [28] Zhu D Q, Lv R, Cao X, et al. Multi-AUV hunting algorithm based on bio-inspired neural network in unknown environments[J]. International Journal of Advanced Robotic Systems, 2015, 12(1): 1-12.
- [29] Ni J J, Yang L, Wu L Y, et al. An improved spinal neural system-based approach for heterogeneous AUVs cooperative hunting[J]. International Journal of Fuzzy Systems, 2018, 20(2): 672-686.
- [30] Ageev D A, Istratov A Y. Neural network implementation for the optimal path problem[J]. Journal of Computer & Systems Sciences International, 1998, 37: 118-125.
- [31] Xia Y S, Wang J. A discrete-time recurrent neural network for shortest-path routing[J]. IEEE Transactions on Automatic Control, 2000, 45(11): 2129-2134.
- [32] Ni J J, Wu L, Shi P, et al. A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles[J]. Computational Intelligence and Neuroscience, 2017, 2017: 1-16.
- [33] Glasius R, Komoda A, Gielen S. A biologically inspired neural net for trajectory formation and obstacle avoidance[J]. Biological Cybernetics, 1996, 74(6): 511-520.
- [34] Adalsteinsson D, Sethian J. A fast level set method for propagating interfaces[M]. Edinburgh: Academic Press Professional, Inc., 1995: 1-22.

作者简介

陈铭治(1987—), 男, 讲师, 博士生, 从事多AUV围捕的研究, E-mail: mingzhichen2008@163.com;

朱大奇(1964—), 男, 教授, 博士生导师, 从事水下机器人路径规划与跟踪控制、多水下机器人协作等研究, E-mail: zdq367@aliyun.com.

(责任编辑: 李君玲)