工程科学学报
Chinese Journal of Engineering

基于强化学习的多无人车协同围捕方法

苏牧青 王寅 濮锐敏 余萌

Cooperative encirclement method for multiple unmanned ground vehicles based on reinforcement learning

SU Muqing, WANG Yin, PU Ruimin, YU Meng

引用本文:
苏牧青, 王寅, 濮锐敏, 余萌. 基于强化学习的多无人车协同围捕方法[J]. 北科大：工程科学学报, 2024, 46(7): 1237−1250. doi: 10.13374/j.issn2095−9389.2023.09.15.004
SU Muqing, WANG Yin, PU Ruimin, YU Meng. Cooperative encirclement method for multiple unmanned ground vehicles based on reinforcement learning[J]. *Chinese Journal of Engineering*, 2024, 46(7): 1237−1250. doi: 10.13374/j.issn2095−9389.2023.09.15.004

在线阅读 View online: https://doi.org/10.13374/j.issn2095−9389.2023.09.15.004

工程科学学报, 第 46 卷, 第 7 期: 1237−1250, 2024 年 7 月
Chinese Journal of Engineering, Vol. 46, No. 7: 1237−1250, July 2024
https://doi.org/10.13374/j.issn2095-9389.2023.09.15.004; http://cje.ustb.edu.cn

基于强化学习的多无人车协同围捕方法

苏牧青[1)], 王　寅[1,2)✉], 濮锐敏[1)], 余　萌[1)]

1) 南京航空航天大学航天学院, 南京 211106　2) 南京航空航天大学航空航天结构力学及控制全国重点实验室, 南京 210016
✉通信作者, E-mail:yinwangee@nuaa.edu.cn

摘　要　本文面向无人车协同围捕问题开展研究, 提出了一种基于柔性执行者−评论家(SAC)算法框架的协同围捕算法. 针对多无人车之间的协同性差的问题, 在网络结构中加入长短期记忆 (LSTM) 构建记忆功能, 帮助无人车利用历史观测序列进行更稳健的决策; 针对网络结构中引入 LSTM 所导致的状态空间维度增大、效率低的问题, 提出引入注意力机制, 通过对状态空间进行注意力权重的计算和选择, 将注意力集中在与任务相关的关键状态上, 从而约束状态空间维度并保证网络的稳定性, 实现多无人车之间稳定高效的合作并提高算法的训练效率. 为解决协同围捕任务中奖励稀疏的问题, 提出通过混合奖励函数将奖励函数分为个体奖励和协同奖励, 通过引入个体奖励和协同奖励, 无人车在围捕过程中可以获得更频繁的奖励信号. 个体奖励通过引导无人车向目标靠近来激励其运动行为, 而协同奖励则激励群体无人车共同完成围捕任务, 从而进一步提高算法的收敛速度. 最后, 通过仿真和实验表明, 该方法具有更快的收敛速度, 相较于 SAC 算法, 围捕时间缩短 15.1%, 成功率提升 7.6%.

关键词　无人车; 协同围捕; 柔性执行者-评论家算法; 注意力机制; 奖励函数设计
分类号　TG142.71

Cooperative encirclement method for multiple unmanned ground vehicles based on reinforcement learning

SU Muqing[1)], *WANG Yin*[1,2)✉], *PU Ruimin*[1)], *YU Meng*[1)]

1) College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
2) State Key Laboratory of Mechanics and Control for Aerospace Structures, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
✉Corresponding author, E-mail: yinwangee@nuaa.edu.cn

ABSTRACT　Collaborative encirclement of multiple unmanned ground vehicles (UGVs) is a focal challenge in the realm of multiagent collaborative tasks, representing a fundamental issue in complex undertakings such as multiagent collaborative search and interception. Although optimization algorithms have yielded rich research outcomes in collaborative encirclement, challenges persist, including poor real-time computational efficiency and weak robustness. Reinforcement learning theory holds considerable promise for addressing multiagent sequential decision problems. This paper delves into the study of the collaborative encirclement of multiple UGVs based on deep reinforcement learning theory, focusing on the following key aspects: establishing a kinematic model for UGVs to describe the collaborative encirclement task, detailing the collaborative encirclement process, developing strategies for target UGV escape, and addressing challenges arising from the increasing number of UGVs, which results in a complex environment and issues such as algorithmic instability, dimension explosion, and poor convergence. This paper introduces a collaborative encirclement algorithm based on the soft actor–critic (SAC) framework. To address issues related to poor collaboration and weak generalization among multiple UGVs, long short-term memory is incorporated into the network structure, serving as a memory function for UGVs. This tactic aids in

收稿日期: 2023−09−15
基金项目: 航空科学基金资助项目 (ASFC-20175152); 南京航空航天大学实验技术研究与开发课题资助项目 (SYJS202311Z)

capturing and using information from historical observation sequences, effectively processing time–series data, making more accurate decisions, promoting mutual collaboration among UGVs, and enhancing system stability. To tackle the issue of increased state space dimensions and low training efficiency during collaborative encirclement, an attention mechanism is introduced to calculate and select attention weights in the state space, focusing attention on key states relevant to the task. This strategy helps constrain state space dimensions, ensuring network stability, achieving stable and efficient collaboration among multiple UGVs, and improving algorithm training efficiency. To address the problem of sparse rewards in collaborative encirclement tasks, a mixed reward function is proposed that divides the reward function into individual and collaborative rewards. Individual rewards guide UGVs toward the target, incentivizing their motion behavior, whereas collaborative rewards motivate a group of UGVs to collectively accomplish the encirclement task. This approach further guides UGVs to obtain more frequent reward signals, ultimately enhancing the algorithm convergence speed. Simulation and experimental results demonstrate that the proposed method achieves faster convergence than SAC, with a 15.1% reduction in encirclement time and a 7.6% improvement in success rate. Finally, the improved algorithm developed in this paper is deployed on a UGV platform, and real-world experiments in typical encirclement scenarios validate its feasibility and effectiveness in embedded systems.

KEY WORDS unmanned ground vehicles; cooperative encirclement; soft actor–critic algorithm; attention mechanism; reward function design

多无人车目标围捕指一群具有协作机制的无人车, 通过一定的编队优化策略成功地围捕目标[1]. 协作围捕不仅可以支持其他任务例如搜索[2]、拦截、队形转换和合作运输, 还可应用于自主无人机控制[3]、自主车辆跟踪控制[4]以及导弹和防御系统[5]等领域, 是多无人车协同领域的关键基础问题之一. 多无人车协同追捕的求解思路可以分为非学习类方法与学习类方法两类[6].

非学习类协同围捕的研究主要集中在微分对策领域, 将协同围捕问题转换为微分对策问题, 以实现无人车之间的协作与合作. Isaacs[7]将围捕问题视为一种非合作动态博弈, 并将该问题建模为一个具有交互决策和动态演化的系统, 运用微分博弈最小化一个目标函数从而求解. Dong 等[8]针对追逃博弈算法存在的计算量大、通用性差等问题, 提出了一种基于改进动态人工势场和微分博弈的混合算法. Sun 等[9]将多个追击者的运动过程分为两部分, 并通过差分博弈理论分别对多个追击者运动过程设计一对一制导律从而完成任务. 尽管微分对策可以用于解决协同围捕问题, 然而, 当涉及到多个追击者时, 问题的复杂性使目标函数求解变得异常困难, 并且由于状态空间和决策空间迅速扩展, 导致求解变得复杂且耗时[10].

图搜索方法[11]研究如何有效地使追踪者在连通图上找到逃避者, 相较于微分博弈法, 图搜索方法不依赖于目标函数的求解, 更注重路径规划和搜索策略的优化, 这使得图搜索方法在处理复杂环境和动态情况时更具鲁棒性, 能够快速地适应不同的追逃者行为和环境变化, 具有简单直观以及可扩展性强的优点. Kehagias 等[12]将围捕问题转换为图节点问题, 并基于此提出了迭代贪婪节点搜索算法, 通过室内环境搜索实验成功捕获逃逸者. 尽管这种方法能够在一定情况下成功捕获逃逸者, 但在动态、未知或复杂环境下, 构建准确的图结构可能变得复杂且耗时, 从而影响搜索的效率和准确性. 由于协同围捕与生物捕食过程的相似性, 因此常被用来探究团队合作的捕猎策略. 生物启发法相对于微分对策法和图搜索方法更加关注团队合作和集群行为, 能够从生物系统中获得一些集体智慧的启示, 具有更强的鲁棒性以及自适应能力. Janosov 等[13]受生物学中自然捕食系统的启发,提出了集群追捕策略. Wang 等[14]从生物学角度出发定义了围捕条件, 并基于运行成本以及相应的系数设计了最优控制策略. 生物启发法也存在着具有较差的适应性以及通用性的限制问题[15].

近年来, 强化学习理论的发展推动了决策规划等领域的研究. 许多算法如 Q 学习 (Q-learning), 深度确定性策略梯度 (DDPG), 近端策略优化算法 (PPO) 等已经被应用于解决各类复杂问题. 在基于强化学习的围捕逃逸问题中, Bilgin 和 Kadioglu-Urtis [16]采用 $Q(\lambda)$-learning 算法解决了在网格地图内对具有学习能力的单目标智能体的追捕问题, 并对不同的学习率、折扣因子和衰减率值进行了研究. Wang 等[17]采用一种具有通信功能的分布式协同追击策略, 通过集中式批评家和分布式行动者结构以及基于学习的通信机制来解决协同追击控制问题. Du 等[18]将蜂窝数据参数共享与课程学

习相结合提出了新的强化学习方法, 从而使围捕无人机能够共享观测信息并最终学到有效包围−围攻−收缩−捕获策略. 针对无人水面飞行器的追捕问题, Qu 等[15] 基于多智能体对等信用分配算法 (MA-POCA) 和集中训练分布式执行架构, 训练追击策略, 使多 USV(Unmanned surface vehicle) 具有自主避障和协同捕获的能力. De Souza 等[6] 使用双重延迟深度确定性策略梯度 (TD3) 以及鼓励良好编队的组奖励结构, 得到了良好的围捕控制策略. Zhang 等[19] 针对协调机器人群体追捕逃兵的问题, 将强化学习和以人工势场作为预定义规则相结合的提出了基于双层决斗深度 Q 网络的自适应合作算法 (DACOOP) 协调追击策略, 并验证了该算法在协同追击问题上的成功率高于 APF(Artificial potential field) 与 D3QN(Dueling double DQN). Hüttenrauch 等[20] 使用置信域策略优化 (TRPO) 解决多智能体围捕问题, 通过将均值特征嵌入 (Mean feature embeddings) 作为状态观测量, 有效解决了智能体感知信息高维度所导致的维度爆炸问题. Liu 等[21] 将双模糊系统与 Q-learning 相结合, 提出了模糊 Q 学习 (FQL) 和 Q 值表模糊推理系统 (QTFIS), 以此克服 Q 学习在处理连续空间问题时的困难, 并打破 Q 学习仅用于低维空间的约束, 从而有效地解决了无人机追踪围捕的问题. Zhang 等[22] 针对多航天器轨道追逃拦截问题提出了拦截策略, 包括深度强化学习 (DRL) 生成的捕获区域 (CZ) 嵌入策略, 以及基于障碍逆解神经网络 (BISNN) 的 CZ 内近似最优制导律. Santos 等[23] 设计了一个并行优化算法来最小化追捕逃逸问题中的捕获时间, 并通过 pac-dot 策略的修剪技术, 以减少状态、转换的数量, 从而减少了计算游戏所需的计算资源, 并提高了该技术的可扩展性. Wang 等[24] 在智能体深度确定性策略梯度 (MADDPG) 基础上, 提出了一种通过感应信息而非通信的围捕策略的协作方法, 并采用集中训练和分布式执行的方式使围捕智能体高度协作. 范之琳等[25] 将模型控制与强化学习原理相结合, 利用势能模型引导的改进多智能体强化学习算法进行围捕, 并在已有势能模型的基础上建立跟踪围捕和环航围捕两种围捕策略.

尽管强化学习在解决围捕问题中已取的丰富的成果, 但依然面临着挑战. 例如大部分研究认为, 将围捕智能体将目标智能体逼进围捕边界便可判定围捕成功, 具有较强的局限性. 多智能体环境中的状态信息通常涵盖了大量维度, 导致高维状态问题的出现. 这进一步增加了状态空间的复杂性和计算负担, 并可能导致在围捕过程中未能形成有效的围捕策略. 在解决高纬度空间训练问题时, 注意力机制由于其在模型中能够集中关注重要的信息, 从而提高模型性能的特性, 得到了广泛的关注和研究. 在神经网络中, 注意力机制使得模型能够根据输入的不同部分分配不同的权重, 从而在处理信息时能够更加灵活和精确[26]. Zhang 等[27] 为解决无人机在城市空域的路径规划与避障问题, 将注意力机制与 DDQN(Double deep Q network) 相结合, 从而减少多米诺骨牌冲突计数, 并最大限度地减少与参考路径的偏差. Peng 等[28] 为解决自动驾驶汽车需要对不确定性动作的行人作出反应的问题, 将图注意力网络与深度强化学习相结合, 提出了 DRL-GAT-SA(Deep reinforcement learning based on graph attention networks and simplex architecture), 该算法为车辆驾驶提供了安全性, 有效避免了碰撞.

本文针对无边界条件下的围捕问题进行研究,结合实际应用背景给无出边界条件下无人车围捕成功的判定条件; 针对围捕问题提出了一种基于柔性执行者−评论家 (SAC) 强化学习框架的求解方法, 通过 LSTM(Long short-term memory) 构建 Actor 与 Critic 网络结构以帮助无人车利用历史观测序列进行更稳健的决策; 在其 Critic 网络中引入了注意力机制, 旨在解决环境和无人车状态变化引发的高维度问题, 以及由此产生的不稳定性问题. 通过混合奖励函数将奖励函数分为个体奖励和协同奖励, 以最大化全局奖励和局部奖励, 从而指导无人车更有效地理解围捕任务. 并做出相应决策. 最后, 通过仿真与实验验证了本文改进算法有效性和实用性.

1 多无人车围捕问题描述

假设围捕无人车在一定区域内执行围捕任务, 目标无人车具有一定的机动能力, 可根据围捕无人车的位置速度信息逃逸, 从而远离围捕无人车.

在一个二维空间内, 有 $n(n \geqslant 3)$个围捕无人车 P_i对一个移动目标无人车T进行围捕, 任务场景如图 1 所示, 三个围捕无人车P_1、P_2、P_3与目标无人车T均在任务环境内的范围内, 并且通过搭载的传感器进行相互感知. V_{P_1}、V_{P_2}、V_{P_3}与V_{T}分别表示三辆围捕无人车与目标无人车的速度; 其速度为 $V_{\mathrm{P}_i} > V_{\mathrm{T}}(i = 1,2,3)$. 假定在初始时刻, 围捕无人车和目标无任何侧可互相感知到对方的存在, 当围捕

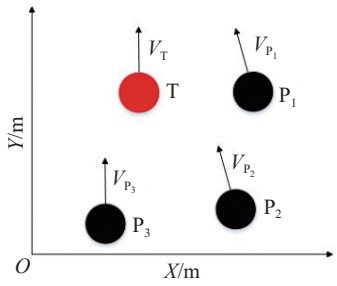

图 1　任务场景描述

Fig.1　Task scenario description

无人车侦测到目标无人车时, 将立即向目标所在位置进行移动. 其中围捕无人车的主要任务是在逐渐靠近目标无人车的过程中完成围捕, 目标无人车T探查到围捕无人车P_i接近后, 根据逃逸策略尽可能使自己远离围捕无人车.

1.1　无人车运动学模型

本文中无人车模型采用双轮差速模型, 假设无人车是一个刚体, 通过调整两个轮子的旋转速度来实现运动, 双轮差速模型如图 2 所示, M为无人车质心; w为无人车宽度. 机器人的运动运动学方程如式 (1) 所示[29]. 状态量为$\boldsymbol{q}=[x,y,\theta]^{\mathrm{T}}$, 其中$x$, y为当前位置, θ为航向角. 运动速度矢量$\boldsymbol{u}=[v,\omega]$, v为瞬时线速度, ω为瞬时角速度.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix}v\\ \omega\end{bmatrix}\tag{1}$$

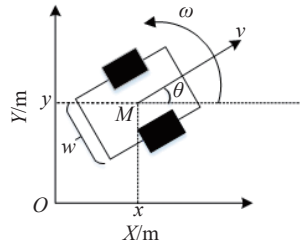

图 2　差分驱动运动学模型

Fig.2　Differential drive kinetic model

1.2　目标无人车逃逸策略

目标无人车从固定的起点运动到固定的终点, 在遇到围捕车辆时会采用一定的规则进行逃逸运动. 本文通过计算相邻围捕无人车之间的角度, 选择其夹角最大的方向进行运动, 运动方向如图 3 所示, 图中P_i、P_j、P_n分别表示编号为i、j、n的围捕无人车. 运动方向设定为由目标无人车所在位置指向围捕无人车P_i与P_j的连线中点$R(x_r,y_r)$, 逃逸角度定义为$\theta_{\mathrm{E}}=\arctan((y_r-y_{\mathrm{T}})/x_r-x_{\mathrm{T}})$. 逃逸运动策略如式 (2) 所示:

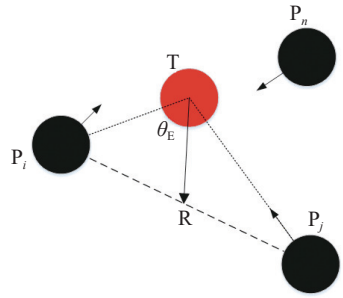

图 3　目标无人车逃逸方向示意图

Fig.3　Diagram of target agent escape direction

$$\begin{cases}\omega=\omega_0+\theta_{\mathrm{E}}, & v=v_0\cdot(1+d_{\mathrm{success}}/\min(d_{P_i}))\\ & 0.3\leqslant d_{P_i}<0.5\wedge 0.3\leqslant d_{P_j}<0.5(i\neq j)\\ v=v_0, & \omega=\omega_0\\ & \text{otherwise}\end{cases}\tag{2}$$

其中, v_0与ω_0表示当前时刻速度与角速度; d_{P_i}表示目标无人车与围捕无人车P_i之间的距离, $\min(d_{P_i})$表示三辆围捕无人车与目标无人车的最短距离; d_{success}表示最大成功围捕半径. 当目标无人车检测到与围捕无人车距离小于或等于 0.5 m 时, 目标车辆根据与终点距离、与围捕车辆距离等信息进行躲避. 如果没有逃逸需求, 目标车辆将继续以原有速度和角速度运动.

1.3　基于角度与距离判别围捕成功条件

理想的围捕包围圈是n个围捕无人车P_i能够均匀地分布在目标无人车T周围. 围捕无人车P_i与其顺时针方向无人车P_j以及目标无人车T所形成的夹角为$\theta_{\mathrm{T}j}$, 即相邻两个无人车的相对角度. 规定围捕成功需满足以下条件, 为了保证围捕无人车与目标无人车不发生碰撞, 且满足形成围捕半径: $d_{\min}<d_{P_i}<d_{\mathrm{success}}$; 其中, $d_{\min}$表示最小碰撞距离. 在本文中以三个围捕无人车为例, 为了保证围捕无人车能够尽可能地均匀分布在目标无人车的周围, 即$\theta_{\mathrm{T}j}$=120°, 在本文中允许存在一定的角度偏移, 偏差角可以设置为10°; 围捕角度条件为: 110° ⩽ $\theta_{\mathrm{T}j}$ ⩽ 130°. 综上, 围捕距离与角度约束条件如式 (3) 所示:

$$\begin{cases}\lim\limits_{t\to\infty} d_{\min}<d_{P_i}<d_{\mathrm{success}}(i\in\mathbf{N})\\ \lim\limits_{t\to\infty}\left|\theta_{\mathrm{T}j}-120°\right|\leqslant 10°(i,j\in\mathbf{N};i\neq j)\end{cases}\tag{3}$$

以三辆捕无人车P_1、P_2、P_3为例, 假定P_1、P_2、P_3以顺时针方向分布在T的周围, 且P_1在T的右上方; 理想围捕包围圈如图 4 所示, P_1、P_2、P_3与T的距离分别为d_{P_1}、d_{P_2}、d_{P_3}均在$d_{\min}\sim d_{\mathrm{success}}$之间; θ_{T}为目标无人车的航向角; α_1为围捕无人车P_1与目标无人车T的目标视线与x轴的夹角; 以目标无人

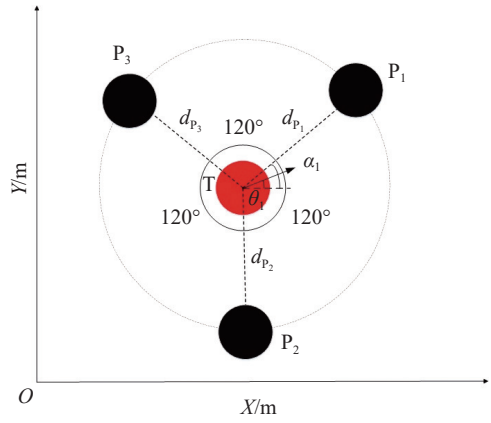

图 4 理想围捕圈

Fig.4 Ideal encirclement circle

车为角的顶点, 围捕无人车与其顺时针方向的围捕无人车所呈理想角度则为120°. P_1理想位置的坐标为$(x_T+d_{P_1}\cdot\cos\alpha_1, y_T+d_{P_1}\cdot\sin\alpha_1)$;根据T位置与围捕无人车与其顺时针方向的围捕无人车所呈的理想角度120°可知, P_2理想位置的坐标为$(x_T+d_{P_1}\cdot\cos(120°-\alpha_1), y_T-d_{P_1}\cdot\sin(120°-\alpha_1))$; P_3理想位置的坐标为$(x_T-d_{P_1}\cdot\cos(\alpha_1-60°), y_T-d_{P_1}\cdot\sin(\alpha_1-60°))$.

传统算法是基于规则和预定义策略的算法, 它们通过手动设计的方式来解决问题[8]. 包括搜索、优化和规划等技术, 然而传统算法存在一些劣势: 传统算法往往依赖于预定义的规则和策略, 对于复杂和不确定的场景很难适应; 难以处理由于无人车状态变化或者环境变化导致的高维度问题. 并且, 对于多围捕无人车的协同集群行为, 难以使用目标优化函数表达.

强化学习是在与环境的交互中主动学习, 通过最大化期望奖励来优化目标[15]: 强化学习在与环境交互的过程中, 无人车能够根据反馈信号不断调整策略, 适应环境的变化和不确定性; 并且能够处理高维、连续和复杂的状态空间. 为此, 本文针对具有逃逸能力的目标无人车围捕问题, 采用强化学习算法来克服传统算法的局限性.

2 基于 SAC 算法的多无人车围捕强化学习策略

2.1 算法设计

SAC 将策略的熵加入策略网络的学习目标中, 在最大化期望累计回报的同时尽可能探索新策略[30]. 在协同围捕中, 无人车围捕策略与时间序列相关, 不仅与当前时刻无人车状态有关, 还受到上一时刻无人车状态的影响. 而循环神经网络 (RNN) 能够处理与时间序列相关的信息. 因此, 本文在 SAC 中加入 RNN 来构建记忆功能, 从而进一步优化算法的性能. 通过整合过去的时间信息来改进决策, 增强在动态场景中的适应性.

LSTM 网络是一种深度神经网络, 其特点是能够提取序列特征并处理复杂的依赖关系. 作为 RNN 的一种形式, LSTM 网络包含外部递归 (从输出到隐藏层输入) 和内部递归 (LSTM 细胞之间). 在每个 LSTM 单元中, 一组门控单元负责调节信息的流动, 这使得网络能够根据输入序列保持或遗忘信息. LSTM 网络由输入层、多个隐藏层和输出层组成. 其关键特点在于隐藏层中的记忆单元, 这些单元通过门控机制控制信息的流动, 选择要记住或忘记的信息. LSTM 网络结构图如图 5 所示.

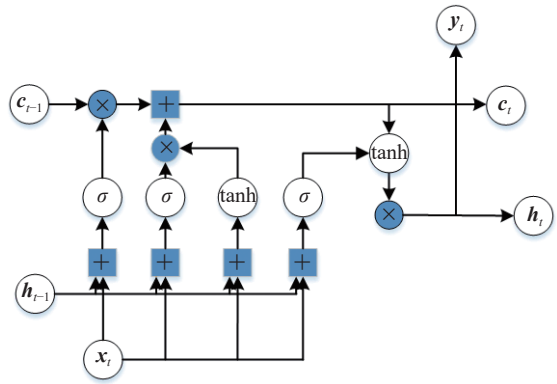

图 5 LSTM 网络结构图

Fig.5 Architecture diagram of LSTM network

遗忘门决定了上一步中哪些相关信息需要保留;输入门决定当前输入中哪些信息是重要的, 需要添加;输出门决定下一个隐藏状态应该是什么. 图中σ表示 Sigmoid 函数, tanh 表示 tanh 函数. Sigmoid 函数与 tanh 函数分别如式 (4) 与式 (5) 所示.

$$\sigma=\frac{1}{1+\mathrm{e}^{-x}} \tag{4}$$

$$\tanh=\frac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}} \tag{5}$$

LSTM 单元中, 遗忘门、输入门、单元状态、输出门和细胞输出的运算结果分别如式 (6)、(7)、(8)、(9) 与 (10) 所示.

$$\boldsymbol{f}_t=\sigma(\boldsymbol{W}_{\mathrm{f}}\cdot\boldsymbol{h}_{t-1}+\boldsymbol{W}_{\mathrm{f}}\cdot\boldsymbol{x}_t+\boldsymbol{b}_{\mathrm{f}}) \tag{6}$$

$$\boldsymbol{i}_t=\sigma(\boldsymbol{W}_{\mathrm{i}}\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{i}}) \tag{7}$$

$$\boldsymbol{c}_t=\boldsymbol{f}_t\odot\boldsymbol{f}_{t-1}+i_t\odot\tanh(\boldsymbol{W}_{\mathrm{xc}}\cdot\boldsymbol{x}_t+\boldsymbol{W}_{\mathrm{hc}}\cdot\boldsymbol{h}_{t-1}+\boldsymbol{b}_{\mathrm{c}}) \tag{8}$$

$$\boldsymbol{o}_t=\sigma(\boldsymbol{W}_{\mathrm{xo}}\cdot\boldsymbol{x}_t+\boldsymbol{W}_{\mathrm{ho}}\cdot\boldsymbol{h}_{t-1}+\boldsymbol{b}_{\mathrm{o}}) \tag{9}$$

$$\boldsymbol{h}_t=\boldsymbol{o}_t\odot\tanh(\boldsymbol{c}_t) \tag{10}$$

其中, $\boldsymbol{h}_{t-1}$和$\boldsymbol{c}_{t-1}$表示上个时刻的细胞输出与细胞

状态, $\boldsymbol{x}_t$、$\boldsymbol{h}_t$和$\boldsymbol{c}_t$分别表示当前时刻的输入状态、细胞输出以及细胞状态; $\boldsymbol{f}_t$、$\boldsymbol{i}_t$、$\boldsymbol{o}_t$分别表示遗忘门、输入门以及输出门的运算结果, $\boldsymbol{W}_{\rm xc}$表示当前时刻的输入状态与细胞状态之间的权重矩阵, $\boldsymbol{W}_{\rm hc}$表示当前时刻的细胞输出与细胞状态之间的权重矩阵, $\boldsymbol{W}_{\rm xo}$表示当前时刻的输入状态与输出门之间的权重矩阵, $\boldsymbol{W}_{\rm ho}$表示当前时刻的细胞输出与输出门之间的权重矩阵; $\boldsymbol{W}_{\rm f}$、$\boldsymbol{W}_{\rm i}$、$\boldsymbol{W}_{\rm o}$、$\boldsymbol{b}_{\rm f}$、$\boldsymbol{b}_{\rm i}$、$\boldsymbol{b}_{\rm o}$分别表示遗忘门、输入门以及输出门的权重与偏置矩阵, ⊙表示哈达玛积, 即相同维数矩阵间逐元素相乘.

当通过 LSTM 构建网络结构时, LSTM 能够提取序列中的长时间的序列信息, 会造成 Actor 与 Critic 网络输入信息维度的升高, 导致维度爆炸, 因此需要加入注意力机制对输入的信息进行筛选, 选择出对自己更为重要的关键信息. 在这种情况下, 通过注意力机制无人车可以更加关注与其协作的无人车的状态和行为, 从而更好地完成协同围捕任务.

注意力机制步骤如下: 假设输入向量为 $\boldsymbol{X}=[x_1,x_2,\cdots,x_m]$; 将向量$\boldsymbol{X}$映射到 Query 空间、Key 空间与 Value 空间分别得到新的向量$\boldsymbol{Q}$、$\boldsymbol{K}$和$\boldsymbol{V}$; 采用点积的形式,基于$\boldsymbol{Q}$、$\boldsymbol{K}$与$\boldsymbol{V}$计算相似度. 其中$\boldsymbol{Q}$、$\boldsymbol{K}$与$\boldsymbol{V}$计算分别如式 (11)、(12) 与 (13) 所示. $\boldsymbol{W}^{\boldsymbol{Q}}$、$\boldsymbol{W}^{\boldsymbol{K}}$、$\boldsymbol{W}^{\boldsymbol{V}}$代表对应的微分参数矩阵.

$$\boldsymbol{Q}=\boldsymbol{X}\cdot\boldsymbol{W}^{\boldsymbol{Q}}=[q_1,q_2,\cdots,q_m] \tag{11}$$

$$\boldsymbol{K}=\boldsymbol{X}\cdot\boldsymbol{W}^{\boldsymbol{K}}=[k_1,k_2,\cdots,k_m] \tag{12}$$

$$\boldsymbol{V}=\boldsymbol{X}\cdot\boldsymbol{W}^{\boldsymbol{V}}=[v_1,v_2,\cdots,v_m] \tag{13}$$

使用 softmax 计算每个位置的注意力分布, 并进行归一化处理, 如式 (14) 所示. 最后进行加权求和, 如式 (15) 所示.

$$\beta_m=\text{softmax}(q_m,k_m) \tag{14}$$

$$\text{Attention}=\sum_{m=1}^{z}\beta_m\boldsymbol{V}_m=\sum_{m=1}^{z}\text{soft}\max(\boldsymbol{Q}_m,\boldsymbol{K}_m)\boldsymbol{V}_m \tag{15}$$

其中, β_m表示注意力权重, 通过对查询向量和键向量的点积进行 softmax 运算得到, 表示每个值向量在输出中的重要性; q_m表示查询向量中的一个元素, 用于与键向量进行匹配, 以确定每个值向量的重要性; k_m表示键向量中的一个元素, 与查询向量匹配, 用于计算注意力权重. 注意力机制工作图如图 6 所示.

本文算法中 Actor 与 Critic 网络结构如图 7 所示.

本文设定多无人车围捕过程中, 状态变量与动作变量如下:

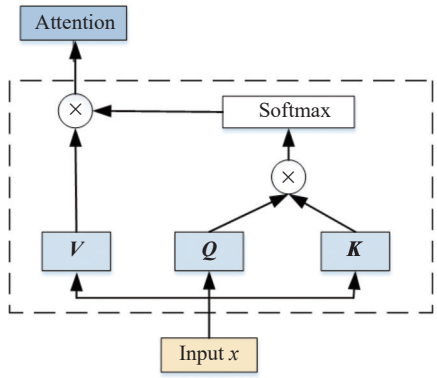

图 6　注意力机制

Fig.6　Attention mechanism

状态变量: 将围捕无人车与目标无人车之间的距离$d_{\rm P_1}$、$d_{\rm P_2}$、$d_{\rm P_3}$, 三辆围捕无人车之间的距离d_{12}、d_{13}、d_{23}, 围捕无人车与其顺时针方向无人车以及目标无人车所形成的夹角$\theta_{\rm 1T2}$、$\theta_{\rm 1T3}$与$\theta_{\rm 2T3}$, 围捕无人车与目标无人车速度之差$v_{\rm T1}$、$v_{\rm T2}$、$v_{\rm T3}$以及围捕无人车与目标无人车航向角θ_1、θ_2、θ_3、$\theta_{\rm T}$设置为状态变量.

$$\begin{aligned}\boldsymbol{S}=\{&d_{\rm P_1},d_{\rm P_2},d_{\rm P_3},d_{12},d_{13},d_{23},\theta_{\rm 1T2},\theta_{\rm 1T3},\theta_{\rm 2T3},\\&v_{\rm T1},v_{\rm T2},v_{\rm T3},\theta_1,\theta_2,\theta_3,\theta_{\rm T}\}\end{aligned} \tag{16}$$

动作变量: 将围捕无人车速度以及角速度作为动作变量, 速度范围为 [0 m·s^{-1}, 0.5 m·s^{-1}]、角速度范围为 [−1 rad·s^{-1}, 1 rad·s^{-1}].

2.2　奖励函数设置

奖励函数的设置在强化学习中是非常关键的, 它直接影响到无人车在学习过程中的表现和效果. 前人的研究中, 基于强化学习的协同围捕奖励函数设置一般是在围捕成功后给予正向奖励, 忽略了围捕过程中无人车运动所产生的奖励[31]. 即使考虑了围捕过程中的运动所产生的奖励, 该奖励函数仍然是以稀疏奖励形式设计的奖励函数, 如果奖励以稀疏频率出现, 无人车难以从稀疏奖励函数学习到最优策略[32]. 因此需要设计能够提供更密集反馈信号的内在奖励函数, 从而克服外部奖励的稀疏性. 本文提出了一种混合奖励函数的方法, 将奖励函数分解为个体奖励和协同奖励, 以最大化整体的奖励函数. 个体奖励通过引导无人车向目标靠近来激励其运动行为, 而协同奖励则激励群体无人车共同完成围捕任务.

2.2.1　距离奖励函数

在该奖励函数中设定当围捕无人车与目标无人车距离小于最大围捕成功距离 (本文仿真中设定为 0.3 m) 时给予奖励正向奖励, 距离奖励函数

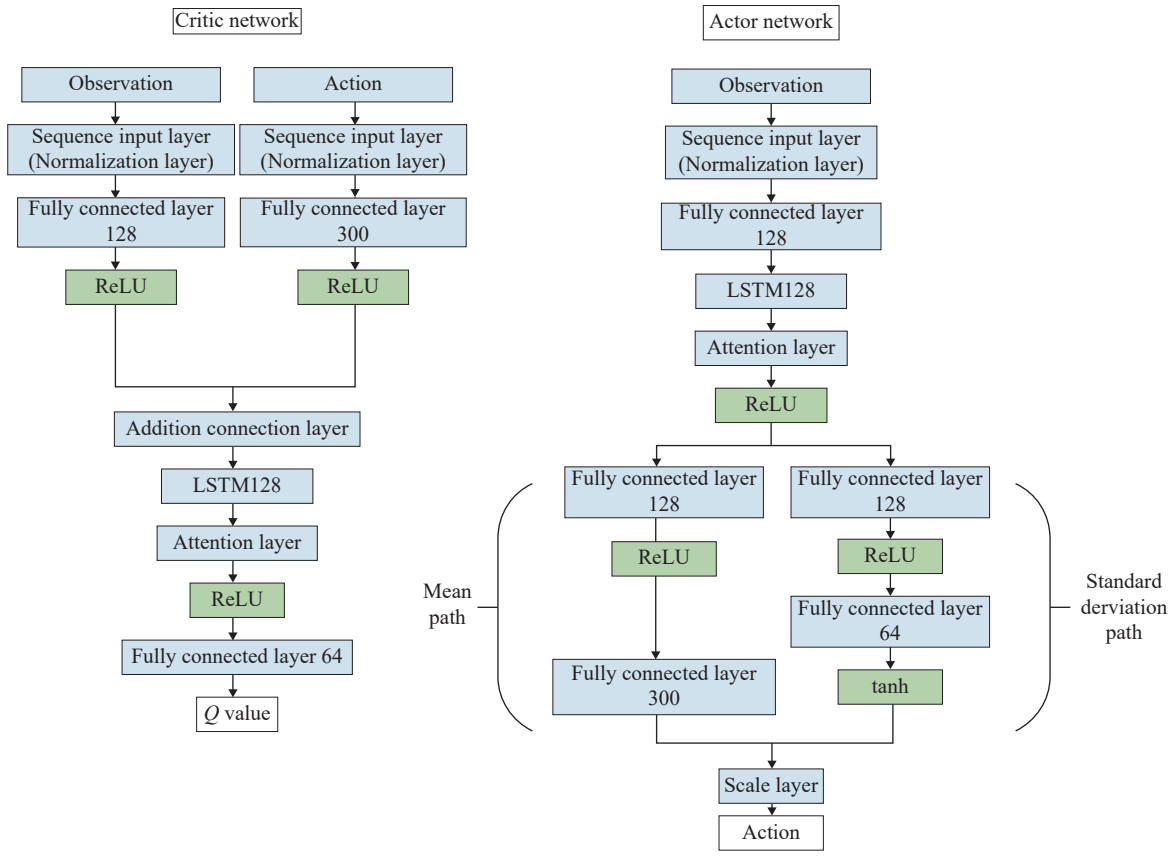

图 7 神经网络结构图

Fig.7 Neural network architecture diagram

如式 (17) 所示:

$$R_{\mathrm{P}_i}=\begin{cases}-(d_{\mathrm{P}_i}-0.232) & d_{\mathrm{P}_i}\leqslant 0.232\ \mathrm{m}\\ -(d_{\mathrm{P}_i}-0.3)\times(d_{\mathrm{P}_i}-0.232) & 0.232\ \mathrm{m}<d_{\mathrm{P}_i}\leqslant 0.3\ \mathrm{m}\\ \mathrm{e}^{(d_{\mathrm{P}_i}-0.3)}-1 & d_{\mathrm{P}_i}>0.3\ \mathrm{m}\end{cases} \tag{17}$$

2.2.2 角度奖励函数

在本文中围捕无人车数量为 3, 因此期望围捕角度范围为 [110°, 120°], 当$\theta_{i\mathrm{T}j}$在该范围内给予正向奖励; 当该角度大于 130°给予惩罚并在角度大于 110°时引入$(\mathrm{e}^{(\theta_{3\mathrm{T}1}-\theta_{2\mathrm{T}1})}-1)$这一项, 作为形状函数, 该函数可以计算形成的三个围捕角度之间的差值, 该值越小说明形成的三个围捕角差异越小, 从而引导围捕无人车形成围捕圈, 且在达到期望围捕角后, 该项比重变大. 角度奖励函数如式 (18) 所示:

$$R_{\mathrm{angle}\_i}=\begin{cases}-(\theta_{i\mathrm{T}j}-110°)^2 & 0\leqslant\theta_{i\mathrm{T}j}\leqslant 110°\\ -(\theta_{i\mathrm{T}j}-110°)\times(\theta_{i\mathrm{T}j}-130°)+ & \\ \quad 0.8\times(\mathrm{e}^{(\theta_{3\mathrm{T}1}-\theta_{2\mathrm{T}1})}-1) & \\ & 110°<\theta_{i\mathrm{T}j}\leqslant 130°\\ 0.3\times(\mathrm{e}^{(\theta_{3\mathrm{T}1}-\theta_{2\mathrm{T}1})}-1)+\mathrm{e}^{(\theta_{i\mathrm{T}j}-130°)}-1 & \\ & \theta_{i\mathrm{T}j}>130°\end{cases} \tag{18}$$

2.2.3 形成理想围捕圈奖励

当围捕无人车之间形成一个围捕圈时, 可以给予额外的奖励, 这样可以有效提高围捕无人车之间的协同合作, 增加围捕成功的概率, 更好地指导无人车的训练. 形成的围捕圈奖励如式 (19) 所示:

$$R_{\mathrm{circle}}=30\quad 0.232<d_{\mathrm{P}_i}<0.3;\\ 110°<\theta_{i\mathrm{T}j}\leqslant 130°(i,j\in 3;i\neq j) \tag{19}$$

其中, 目标无人车T在由三辆围捕无人车P_1、P_2和P_3构成的三角形之中.

惩罚是一种负向的奖励, 用于告诉无人车某个动作是不希望它执行的. 在本文中, 当无人车发生碰撞、围捕无人车与目标无人车过远以及围捕无人车超出边界时给予如下惩罚.

(1)碰撞惩罚.

本文中设定碰撞距离为 0.232 m; 围捕无人车之间距离为$d_{ij}(i\neq j)$, 当围捕无人车与目标无人车碰撞给予惩罚并停止训练, 碰撞惩罚函数如式 (20) 所示:

$$P_{\mathrm{collision}}=\quad -10\quad d_{\mathrm{P}_i}\leqslant 0.232\ \mathrm{m};\ d_{ij}\leqslant 0.232\ \mathrm{m} \tag{20}$$

(2)距离惩罚.

本文设定当围捕无人车与目标无人车距离大

于 2 m 时给予惩罚并停止训练, 距离惩罚函数如式 (21) 所示:

$$P_{\text{longdistance}} = \begin{cases} -10 & d_{\text{P}_i} \geqslant 2 \text{ m} \end{cases} \tag{21}$$

为了激励其更好地学习和适应任务并防止无人车陷入死循环或者无法完成任务的状态之中, 在奖励函数中加入负奖励项−0.1, 综上奖励函数设置如式 (22) 所示:

$$Re = 0.6\times\sum_{i=1}^{3} R_{\text{P}_i} + 0.4\times\sum_{i=1}^{3} R_{\text{angle\_}i} + R_{\text{circle}} + P_{\text{collision}} + P_{\text{longdistance}} - 0.1 \tag{22}$$

3　仿真与实验验证分析

为验证本文提出的基于循环神经网络的多无人车强化学习协同围捕算法的性能, 通过仿真实验将其与 SAC 进行比较. 仿真环境为: 操作系统 Win11、处理器 3.0 GHz Intel Core i9, 内存 32G, 用 pycharm3.10 对协同围捕过程进行仿真.

3.1　仿真验证与分析

为验证本文所提出算法的有效性和优越性, 在仿真中将本文所提算法和 SAC 算法进行了比较分析. 由于本文改进算法是在 SAC 框架下进行的改进, 因此仿真参数(表 1)与 SAC 算法一致, 无人车仿真参数如表 2 所示. 在二维连续空间建立一个无边界的多无人车围捕环境, 三辆围捕无人车与目标无人车初始位置分别设置为 (−0.05, 1.08)、(−0.02, 0.41)、(−0.03,0)、与 (0.53, 0.45). 仿真步长为 30 s, 每回合步数为 300 步.

表 1　SAC 与本文改进算法的相关参数

Table 1　Parameters of SAC and the improved algorithm in this paper

Discount factor	Entropy weight	Entropy weight learning rate	Entropy weight threshold	Experience replay	Policy update frequency	Critic update frequency
0.99	0.1	0.001	0.01	10000000	2	2

表 2　仿真环境无人车的相关参数

Table 2　Parameters of UGV in the simulation environment

Wheel radius/m	Robot radius/m	Target UGV initial speed/(m·s^{-1})	Target UGV maximum speed/(m·s^{-1})	Pursuing UGV initial speed/(m·s^{-1})	Pursuing UGV max speed/(m·s^{-1})	Angular speed range/(rad·s^{-1})
0.0335	0.116	0.1	0.3	0.2	0.5	−1−1

Notes: UGV is unmanned ground vehicle.

在以下场景中验证本文所提出方法的有效性: 三个围捕无人车对目标无人车进行围捕, 目标无人车以本文设计的规则进行逃逸, 当围捕无人车集群满足围捕的距离角度条件, 判定完成围捕任务. 针对本文改进算法、SAC 种算法进行对比验证, 训练回合次数为 30000 次. 不同算法训练的回合平均总奖励如图 8 所示.

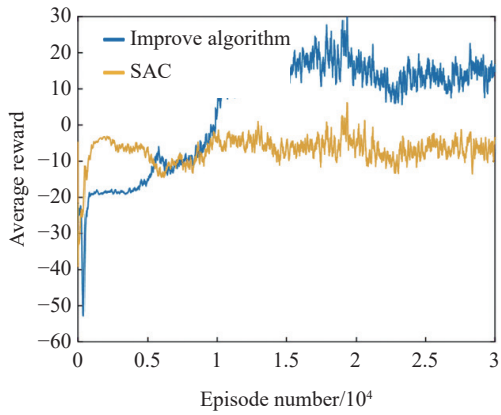

图 8　训练过程平均奖励曲线

Fig.8　Average reward curve during training

根据图 8 的结果, 使用本文改进算法能够获取更大的平均奖励, 收敛速度相较于 SAC 更快. 这是由于本文改进算法采用了 LSTM 结构, 并引入了注意力机制, 无人车能够从环境中获取更多的信息, 并从获得的信息中筛选出对自身会更加有利的信息, 从而做出更正确的决策, 其表现便是可以获取更高的平均奖励, 并且相较于 SAC 算法收敛速度更快.

为了更加直观地对两种算法进行性能比较与分析, 从两种算法测试结果中, 选取了一次多无人车围捕成功的过程来展示围捕效果. 两种围捕算法的轨迹比较如图 9 与图 10 所示. 图 9 表示基于本文改进算法的围捕轨迹图, 图 10 表示基于 SAC 的围捕轨迹图, 可以发现三辆围捕无人车从初始位置出发, 逐步靠近目标无人车并形成了围捕圈最终完成了围捕任务, 其黑色虚线所形成的三角形表示三辆围捕无人车组成的围捕队形.

由图 9(a) 可知, 三辆围捕无人车分别由初始位置 (−0.05, 1.08)、(−0.02, 0.41)、(−0.03, 0) 附近出

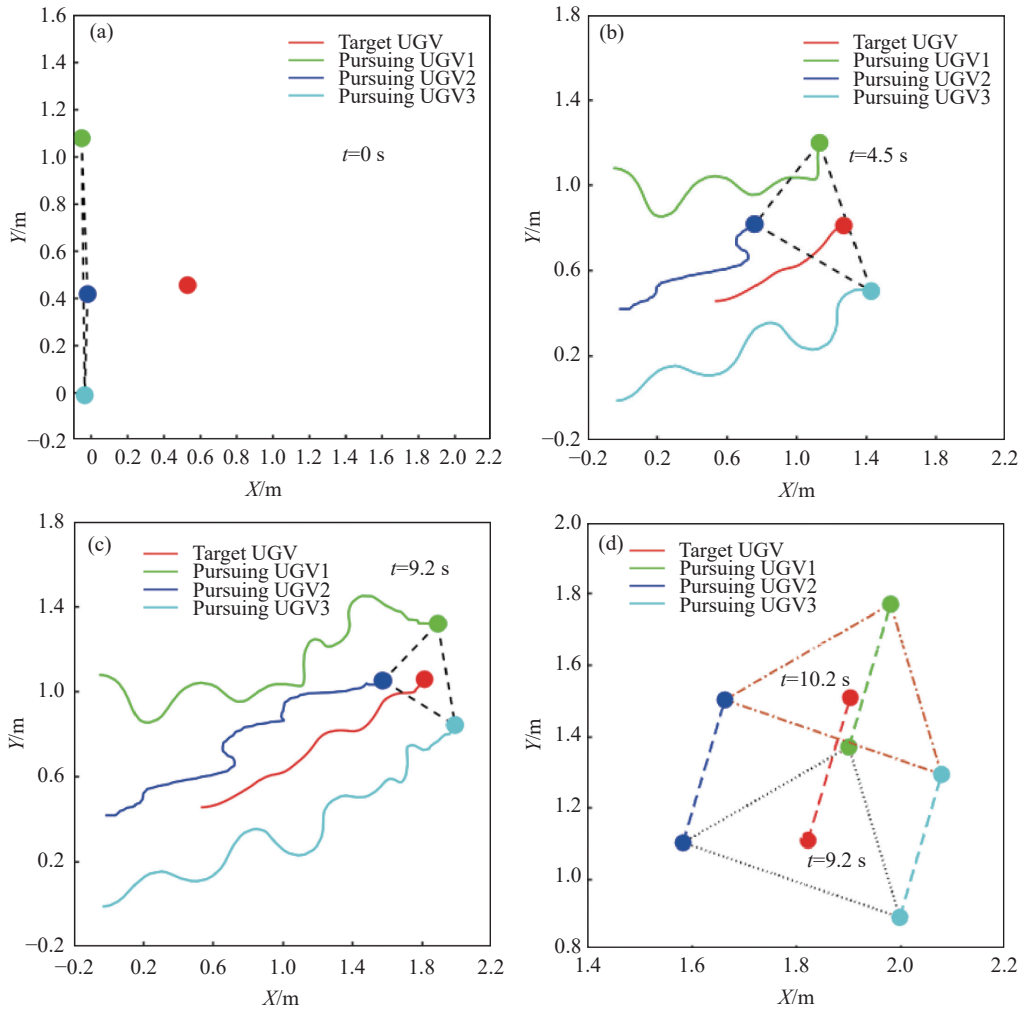

图 9 本文改进算法轨迹图. (a) t=0; (b) t=4.5 s; (c) t=9.2 s; (d) t=10.2 s

Fig.9 Trajectory diagram of the improved algorithm used in this paper: (a) t=0; (b) t=4.5 s; (c) t=9.2 s; (d) t=10.2 s

发, 向处于 (0.53, 0.45) 附近的目标无人车运动; 由图 9(b) 可知, 在 4.5 s 时三辆围捕无人车已对目标无人车形成围捕队形; 由图 9(c) 可知, 三辆围捕无人车在 9.2 s 成功完成围捕任务; 由图 9(d) 可知, 围捕无人车在完成围捕任务后能够保持一定的围捕队形并跟随目标无人车运动 1 s, 图中橙色虚线表示在保持围捕队形 1 s 后三辆围捕无人车所形成的围捕队形.

由图 10(a) 可知, 三辆围捕无人车分别由初始位置 (−0.05, 1.08)、(−0.02, 0.41)、(−0.03, 0) 附近出发向处于 (0.53, 0.45) 附近的目标无人车运动; 由图 10(b) 可知, 在 4.6 s 时三辆围捕无人车已对目标无人车形成围捕队形; 由图 10(c) 可知, 三辆围捕无人车在 10.8 s 成功完成围捕任务; 由图 10(d) 可知, 围捕无人车在完成围捕任务后能够保持一定的围捕队形并跟随目标无人车运动 1 s, 图中橙色虚线表示在保持围捕队形 1 s 后三辆围捕无人车所形成的围捕队形. 在该围捕场景下, 基于本文算法相较于 SAC 算法围捕时间缩短了 14.8%.

图 11 表示围捕无人车之间的相对角度的变化曲线. 图例中 Prusing UGV1-2 表示围捕无人车 1 与围捕无人车 2 之间的相对角度, Prusing UGV1-3 和 Prusing UGV2-3 以此类推. 其中, 黑色的虚线代表设置的角度参考线为 120°. 围捕无人车与相邻围捕无人车之间的相对角度越靠近 120°, 表面围捕效果越好. 图 11(a) 表示基于 SAC 的协同围捕算法各个围捕无人车之间的相对角度变化曲线; 图 11 (b) 为基于本文改进算法的各个围捕无人车之间的相对角度变化曲线. 由图 11 可知, 基于本文改进算法的围捕无人车的相对角度变化更加迅速, 在围捕成功后仍然可以保持较小角度震荡并跟随目标无人车运动, 证明了算法的稳定性.

图 12 表示围捕无人车与目标无人车距离变化曲线. 其中, 黑色的虚线代表设置的距离参考线为 0.3 m. 图 12(a) 为基于 SAC 算法的各个围捕无人车与目标无人车距离变化曲线; 图 12(b) 为基于本

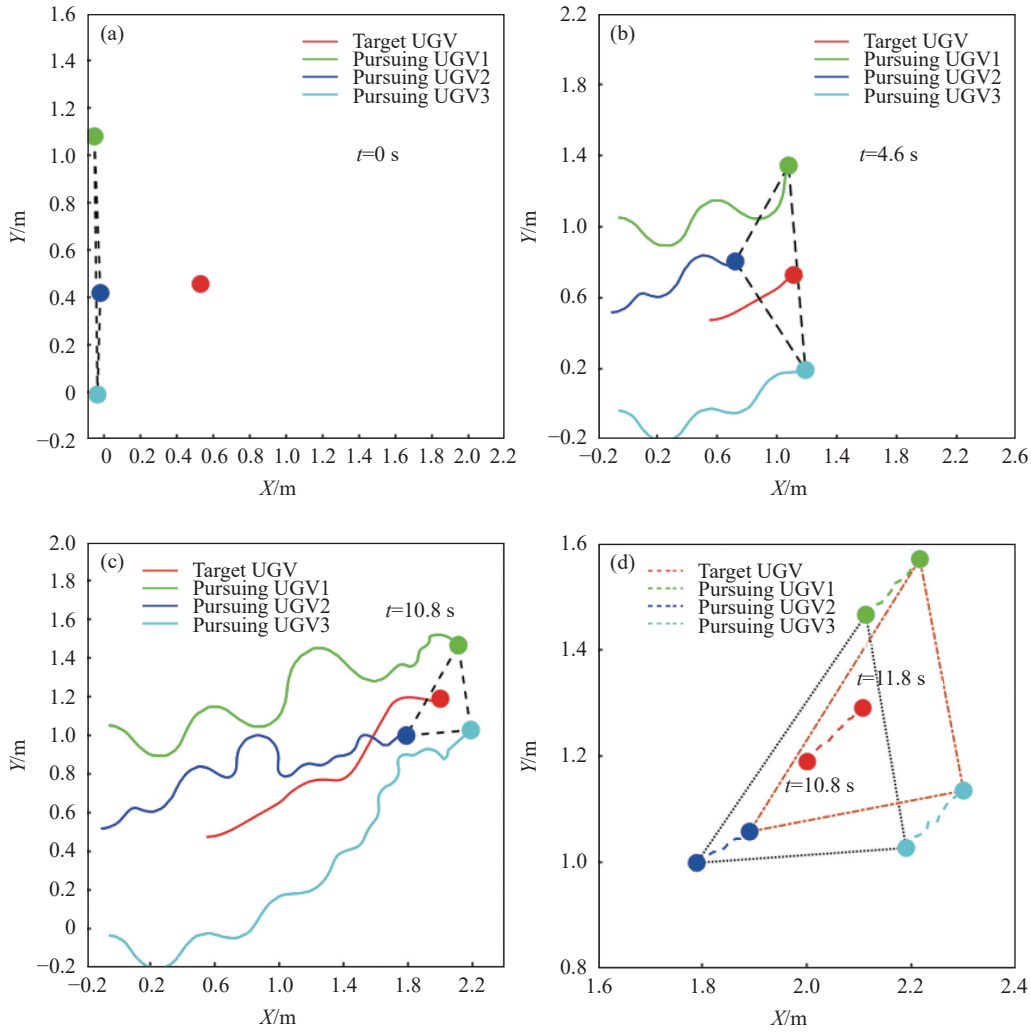

图 10　SAC 算法轨迹图. (a) t=0; (b) t=4.6 s; (c) t=10.8 s; (d) t=11.8 s

Fig.10　Trajectory diagram of SAC: (a) t=0; (b) t=4.6 s; (c) t=10.8 s; (d) t=11.8 s

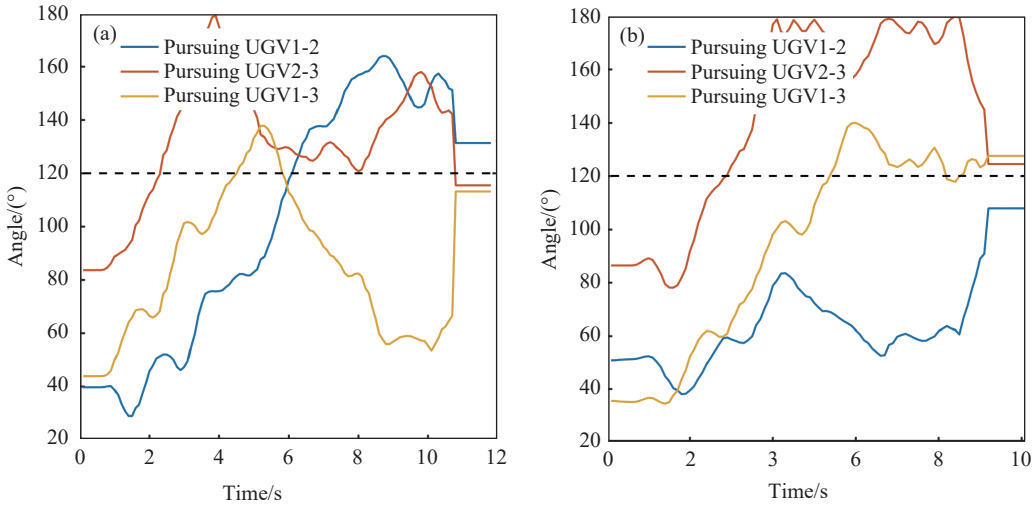

图 11　基于不同算法的围捕无人车之间相对角度的变化曲线. (a) SAC 算法; (b) 本文改进算法

Fig.11　Curve of relative angle variation between pursuing UGVs based on different algorithms: (a) SAC; (b) improved algorithm used in this paper

文改进的协同围捕算法的各个围捕无人车与目标无人车距离变化曲线. 从图 12 中可知, 围捕无人车与目标无人车距离随着围捕进程的增加而逐渐向最优围捕距离条件(0.3 m)收敛, 满足围捕距离

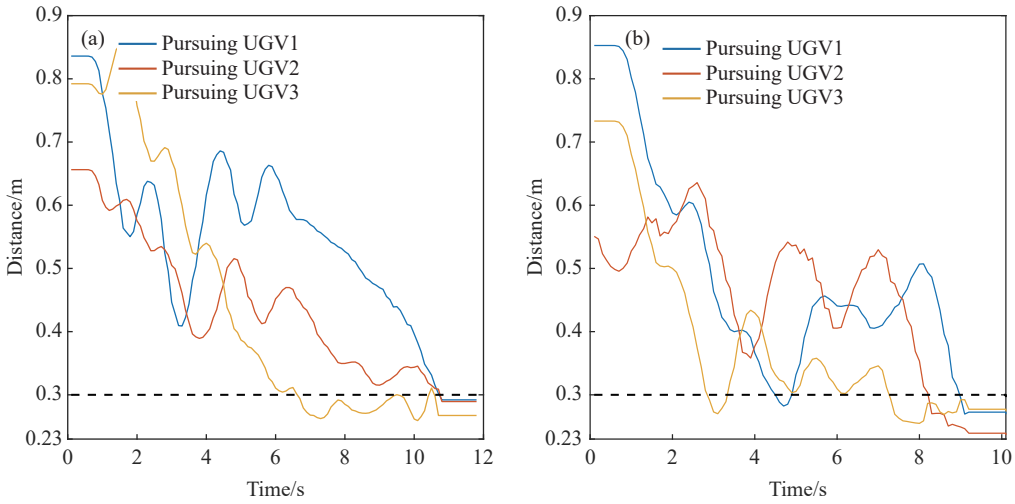

图 12 基于不同算法的围捕无人车与目标无人车距离变化曲线. (a) SAC 算法; (b) 本文改进算法

Fig.12 Distance variation curve between pursuing unmanned vehicles and target unmanned vehicles based on different algorithms: (a) SAC; (b) improved algorithm used in this paper

条件; 使用本文改进算法的围捕无人车与目标无人车距离变化更快. 围捕无人车围捕成功后, 在目标无人车运动的过程中, 采用本文挨近算法的三辆围捕无人车与目标无人车的平均距离分别为 0.27、0.23、0.27 m; 采用 SAC 算法的三辆围捕无人车与目标无人车的平均距离分别为 0.29、0.28、0.26 m. 相较于 SAC 算法,本文改进算法能够更加靠近目标无人车, 从而形成更紧凑的围捕队形.

为进一步验证本文所提出算法的泛化性和鲁棒性, 对前文两种算法进行了蒙特卡洛仿真, 重点考察目标无人车与围捕无人车在不同初始位置、不同最大速度的围捕能力. 基于蒙特卡洛仿真的无人车的相关参数设置如表 2 所示. 基于蒙特卡洛仿真的不同算法在该场景下围捕任务中的表现如表 3 所示. 本文改进的协同围捕算法, 平均消耗时间和围捕无人车行驶距离更短. 同时, 围捕队形保持时间占围捕总耗时的比率更高, 并且围捕成功率也更高. 这些结果充分表明了本文提出的改进算法在解决协同围捕问题上具有显著的优势.

3.2 实验验证与分析

为进一步验证本文所提方法的可行性和部署性, 将所提出的协同围捕算法部署到多无人车实验平台上. 无人车平台由无人车群、定位摄像头、上位机等组成, 定位系统可以通过定位摄像头识别车身二维码获取无人车车辆位置和方位角等信息. 上位机可以获取各无人车运动信息, 并通过无线网络向各无人车广播其他无人车运动状态. 无人车实物参数如表 4 所示.

图 13 表示围捕过程的状态图. 由图 13 可知三个围捕无人车从初始位置开始运动, 对目标无人车进行围捕, 并满足围捕的距离与角度条件, 从而完成围捕任务.

图 14 表示实物验证中围捕无人车与目标无人车的距离和相对角度变化曲线, 图 14(a) 表示三辆围捕无人车与目标无人车的距离变化曲线, 黑色虚线表示最大围捕距离, 即 0.3 m; 图 14(b) 表示三

表 3 基于蒙特卡洛仿真不同算法实验结果

Table 3 Experimental results of different algorithms based on Monte Carlo simulation

Algorithm	Average distance traveled per UGV/m	Total time/s	Formation circle consumption time for the first time/s	Success rate/%
SAC	5.8	11.2	5.2	78
Improved algorithm in this paper	4.3	9.5	4.8	84

表 4 无人车实物参数

Table 4 UGV material parameters

Shape and size/(mm×mm×mm)	Wheelbase/mm	Wheel diameter/mm	Drive mode	Speed range/($m \cdot s^{-1}$)
119.75×105.01×79.07	84.67	60.5	Dual-wheel differential	0–0.5

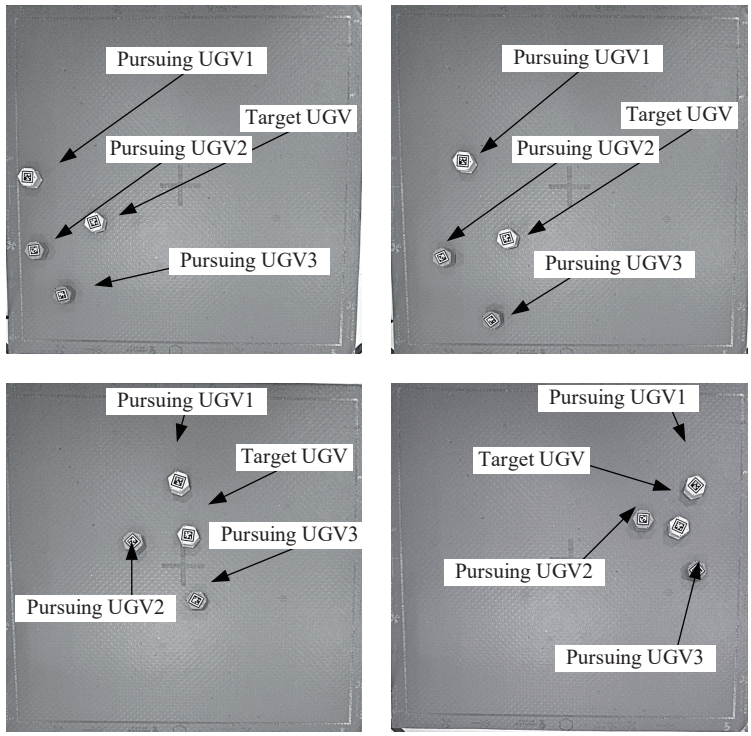

图 13　围捕过程状态图

Fig.13　State diagram of the capture process

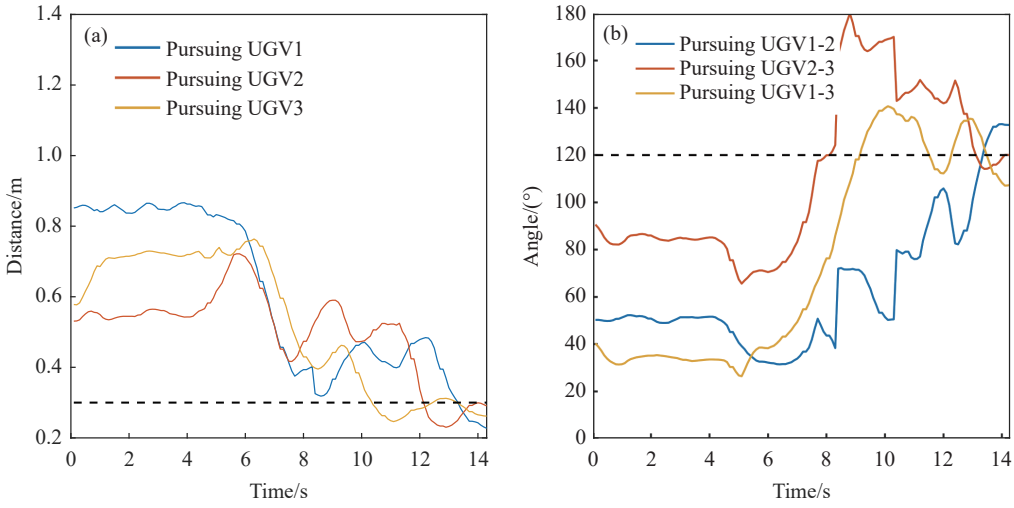

图 14　围捕过程的实物验证中的距离和相对角度的变化曲线. (a) 围捕无人车与目标无人车距离; (b) 围捕无人车之间的相对角度

Fig.14　Curve of distance and relative angle changes during the physical verification of the encirclement process: (a) distance between the pursuing UGV and target UGV; (b) relative angle between the pursuing UGVs

辆围捕无人之间与目标无人车的相对角度变化曲线, 黑色虚线表示最优围捕角度, 即 120°. 由图 14 可知, 围捕无人车与目标无人车距离随着围捕进程的增加而逐渐向 0.3 m 靠拢,围捕无人车的相对角度均向 120°趋近, 在围捕完成后围捕无人车仍然能够跟随目标无人车运动, 距离小于 0.3 m, 相对角度在 120°附近震荡 1.2 s, 证明了该算法的稳定性和迁移性.

4　结论

本文针对无人车协同围捕问题, 考虑实际任务需求以及约束条件, 建立了运动学模型, 并界定了直接围捕成功的条件. 在 SAC 的框架上, 基于 LSTM 构建 Actor 与 Critic 网络结构以促进无人车集群相互协作; 在 Actor 与 Critic 网络中引入注意力机制以解决网络结构中由于 LSTM 可能导致的高维度引起的网络不稳定的问题. 针对任务需求,

优化了强化学习的状态空间和动作空间, 提出了一种将个体奖励与协同奖励相结合的奖励函数构建策略. 构建了仿真环境并完成了协同围捕策略的训练. 仿真实验表明: 与 SAC 相比, 本文改进的协同围捕算法在训练过程中, 平均奖励更高, 收敛速度更快; 在无人车协同围捕任务中,可快速形成合围态势并完成协同围捕; 并且围捕车行驶路径更短、围捕消耗时间更短、围捕成功率更高. 本文所提算法在围捕时间上相较于 SAC 算法缩短了 15.1%, 成功率提升了 7.6%.

参 考 文 献

[1] Garcia E, Casbeer D W, Von Moll A, et al. Multiple pursuer multiple evader differential games. *IEEE Trans Autom Contr*, 2021, 66(5): 2345

[2] Yu D X, Chen C L P. Smooth transition in communication for swarm control with formation change. *IEEE Trans Ind Inform*, 2020, 16(11): 6962

[3] Camci E, Kayacan E. Game of drones: UAV pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning // *Proceedings of the* 2016 *IEEE International Conference on Fuzzy Systems* (*FUZZ-IEEE*). Vancouver, 2016: 618

[4] Vidal R, Rashid S, Sharp C, et al. Pursuit-evasion games with unmanned ground and aerial vehicles // *Proceedings* 2001 *ICRA IEEE International Conference on Robotics and Automation* (*Cat No.*01*CH*37164). Seoul, 2001: 2948

[5] Turetsky V, Shima T. Target evasion from a missile performing multiple switches in guidance law. *J Guid Contr Dyn*, 2016, 39(10): 2364

[6] De Souza C, Newbury R, Cosgun A, et al. Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robot Autom Lett*, 2021, 6(3): 4552

[7] Isaacs R. Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization. *Math Gaz*, 1967, 51(375): 80

[8] Dong J, Zhang X, Jia X M. Strategies of pursuit-evasion game based on improved potential field and differential game theory for mobile robots // 2012 *Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*. Harbin, 2012: 1452

[9] Sun Q, Chen Z P, Qi N M, et al. Pursuit and evasion conflict for three players based on differential game theory // *Proceedings of the* 2017 29*th Chinese Control and Decision Conference* (*CCDC*). Chongqing, 2017: 4527

[10] Hu Y Y, Lin X F, Zhang Y L, et al. Partition of cooperative acquisition space for multiple aircrafts based on differential game. *Ordnace Ind Autom*, 2023, 42(8): 55
(胡艳艳, 林旭锋, 张艳玲, 等. 基于微分对策的多飞行器协同捕获空间划分. 兵工自动化, 2023, 42(8): 55)

[11] Haslegrave J. An evasion game on a graph. *Discrete Math*, 2014, 314: 1

[12] Kehagias A, Hollinger G, Singh S. A graph search algorithm for indoor pursuit/evasion. *Math Comput Model*, 2009, 50(9-10): 1305

[13] Janosov M, Virágh C, Vásárhelyi G, et al. Group chasing tactics: How to catch a faster prey. *New J Phys*, 2017, 19(5): 053003

[14] Wang J N, Li G, Liang L, et al. A pursuit-evasion problem of multiple pursuers from the biological-inspired perspective // *Proceedings of the* 2021 40*th Chinese Control Conference* (*CCC*). Shanghai, 2021: 1596

[15] Qu X Q, Gan W H, Song D L, et al. Pursuit-evasion game strategy of USV based on deep reinforcement learning in complex multi-obstacle environment. *Ocean Eng*, 2023, 273: 114016

[16] Bilgin A T, Kadioglu-Urtis E. An approach to multi-agent pursuit evasion games using reinforcement learning // *Proceedings of the* 2015 *International Conference on Advanced Robotics* (*ICAR*). Istanbul, 2015: 164

[17] Wang Y D, Dong L, Sun C Y. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing*, 2020, 412: 101

[18] Du W B, Guo T, Chen J, et al. Cooperative pursuit of unauthorized UAVs in urban airspace via Multi-agent reinforcement learning. *Transp Res Part C Emerg Technol*, 2021, 128: 103122

[19] Zhang Z, Wang X H, Zhang Q R, et al. Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning // 2022 *International Conference on Robotics and Automation* (*ICRA*). Philadelphia, 2022: 8808

[20] Hüttenrauch M, Sosic A, Neumann G. Deep reinforcement learning for swarm systems. *J Mach Learn Res*, 2019, 20(54): 1

[21] Liu S Z, Hu X X, Dong K J. Adaptive double fuzzy systems based Q-learning for pursuit-evasion game. *IFAC PapersOnline*, 2022, 55(3): 251

[22] Zhang J R, Zhang K P, Zhang Y, et al. Near-optimal interception strategy for orbital pursuit-evasion using deep reinforcement learning. *Acta Astronaut*, 2022, 198: 9

[23] Santos R F D, Ramachandran R K, Vieira M A M, et al. Parallel multi-speed pursuit-evasion game algorithms. *Robot Auton Syst*, 2023, 163: 104382

[24] Wang S X, Wang B, Han Z M, et al. Local sensing based multi-agent pursuit-evasion with deep reinforcement learning // 2022 *China Automation Congress* (*CAC*). Xiamen, 2022: 6748

[25] Fan Z L, Yang H Y, Han Y L. Target round-up control for multi-agent systems based on reinforcement learning. *Acta Aeronaut Astronaut Sin*, 2023, 44(S1): 236
(范之琳, 杨洪勇, 韩艺琳. 基于强化学习的多智能体系统目标围捕控制. 航空学报, 2023, 44(S1): 236)

[26] Liu L Q, Xu X L. Self-attention mechanism at the token level: Gradient analysis and algorithm optimization. *Knowl Based Syst*, 2023, 277: 110784

[27] Zhang M C, Yan C, Dai W, et al. Tactical conflict resolution in urban airspace for unmanned aerial vehicles operations using attention-based deep reinforcement learning. *Green Energy Intell Transp*, 2023, 2(4): 100107

[28] Peng Y F, Tan G Z, Si H W, et al. DRL-GAT-SA: Deep reinforcement learning for autonomous driving planning based on graph attention networks and simplex architecture. *J Syst Archit*, 2022, 126: 102505

[29] Kim Y, Singh T. Energy-time optimal control of wheeled mobile robots. *J Frankl Inst*, 2022, 359(11): 5354

[30] Kathirgamanathan A, Mangin E, Finn D P. Development of a soft actor critic deep reinforcement learning approach for harnessing energy flexibility in a large office building. *Energy AI*, 2021, 5: 100101

[31] Xia J W, Zhu X F, Zhang J Q, et al. Research on cooperative hunting method of unmanned surface vehicle based on multi-agent reinforcement learning. *Contr Decis*, 2023, 38(5): 1438
(夏家伟, 朱旭芳, 张建强, 等. 基于多智能体强化学习的无人艇协同围捕方法. 控制与决策, 2023, 38(5): 1438)

[32] Yu L L, Huo S X, Wang Z J, et al. Hybrid attention-oriented experience replay for deep reinforcement learning and its application to a multi-robot cooperative hunting problem. *Neurocomputing*, 2023, 523: 44