

基于 MATD3 算法的多智能体避碰控制

郭雷^{1†}, 梁成庆²

(1. 河海大学 理学院, 江苏 南京 211100; 2. 河海大学 人工智能与自动化学院, 江苏 南京 211100)

摘 要:使用多智能体双延迟深度确定性策略梯度(Multi-agent Twin Delayed Deep Deterministic Policy Gradient, MATD3)算法研究了多无人机的避障和到达目标点问题,首先,利用 MATD3 算法的优越性提高训练效率。其次,基于人工势场法的思想设计了稠密碰撞奖励函数,使得智能体在没有找到最优解决方案时也能得到积极的反馈,加快学习速度。最后,在仿真实验阶段,通过设计的三组对比实验和泛化实验验证了算法的有效性。

关键词:多智能体;强化学习;人工势场法;避障

中图分类号:TP273

文献标识码:A

Multi-agent Collision Avoidance Control Based on MATD3 Algorithm

GUO Lei^{1†}, LIANG Chengqing²

(1. College of Science, Hohai University, Nanjing, Jiangsu 211100, China;

2. School of Artificial Intelligence and Automation, Hohai University, Nanjing, Jiangsu 211100, China)

Abstract: Multi-agent twin delayed deep deterministic policy gradient (MATD3) algorithm is used to study the obstacle avoidance and target reaching of UAVs. Firstly, the advantage of MATD3 algorithm is used to improve the training efficiency. Secondly, a dense reward function is designed based on the idea of artificial potential field, which can accelerate the learning speed and help the agent get positive feedback when the optimal solution is not found. In the experiment, the effectiveness of the algorithm is verified by the comparison experiment and generalization experiment.

Key words: multi-agent; reinforcement learning; artificial potential field; obstacle avoidance

在 2020 年亚阿战争^[1]中,亚美尼亚投入了大量的装甲部队,阿塞拜疆则投入了大量的攻击型无人机,以压倒性的优势取得了战争早期的胜利。这次战争最鲜明的特点就是无人机进攻战术的有效性,无人机集群作战必将在未来一体化联合作战中扮演独一无二的角色。除此之外,无人机在物资运输、精准农业、城市交通和应急救援等方面也有巨大潜力。

多无人机控制^[2]是指由一定数量的无人机组成,利用信息交互与反馈、激励与响应,实现相互间行为协同,适应动态环境,共同完成特定任务的自主式空中智能系统。多无人机协作首先要考虑的就是避障和无人机之间的避碰问题。传统避障算法包括人工势场法^[3]、遗传算法^[4]、可视图法^[5]、A*算法^[6]等。人工势场法的基本思想是在障碍物周围构建斥力势场,在目标点周围构建引力势

场,智能体在这两种势场的组合场中同时受到斥力和引力作用,合力方向指引着智能体的运动。其优点在于它是一种反馈控制策略,对控制和传感误差有一定的鲁棒性;缺点在于存在局部极小值问题,因此无法保证一定能找到问题的最优解。实际上,针对更加复杂的模型以及未知的地图环境,上述提及的算法具有一定的局限性。

强化学习是智能体通过不断与环境交互来优化智能体的行为策略,进而使智能体的期望收益最大化的一种自学习的机器学习算法。其概念最早在 20 世纪 50 年代被提出,随后 Bellman^[7]提出的经典马尔科夫决策模型以及 Watkins^[8]提出的 Q 学习算法让传统强化学习算法得到进一步推广。2015 年谷歌 DeepMind^[9]团队使用深度 Q 学习(Deep Q-Network, DQN)算法训练的单智能体在多个 Atari 游戏上的高端表现以及围棋智能体 AlphaGo 的成功,把强化学习的热度推向最高。2016 年 Lillicrap 等人^[10]提出的深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)算法是 Actor-Critic (A-C)框架下的一种确定性深度强化学习算法,可以完成连续动作空间上的任务。双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient, TD3)算法是由 Scott Fujimoto 等人^[11]在 DDPG 算法上改进得到的一种用于解决连续控制问题的深度强化学习算法。TD3 算法在 DDPG 算法的基础上增加一个 Critic 网络和一个目标 Critic 网络来缓解 Q 值高估问题。与 DDPG 算法相比,TD3 算法在连续控制任务中的表现获得了显著改善,受到了众多学者的关注。

为解决多无人机避障等多智能体控制问题,多智能体强化学习(Multi-Agent Reinforcement Learning, MARL)算法被提出。在 MARL 算法结构中“集中式训练、分布式执行”是当前常用的一种训练框架。在此框架下 Lowe^[12]团队提出了多智能体深度确定性策略梯度(MADDPG)算法来解决协作环境、竞争环境和混合环境下的多智能体问题;Yan 等人^[13]使用 MAPPO 算法实现了多智能体的避障和集群;Zhang 等人^[14]使用 MADDPG 算法实现了多智能体的链式编队、V 字编队和钻石编队飞行;Gutpa 等人^[15]使用 MATD3 算法实现了多智能体的编队飞行。但在上述研究中并未充分考虑智能体之间的避碰问题,也未在环境中设置

更多障碍物来验证算法的有效性。鉴于 MATD3 算法的优越性,其将被采用来研究考虑了多无人机间避碰的避障控制,以填补空白。

基于上述文献的启发,在 MATD3 算法的框架下,研究了考虑无人机之间避碰的多智能体控制问题,并做出以下改进:基于人工势场法的思想设计稠密奖励函数,使得智能体在没有找到最优解时也能得到积极的反馈,奖励收敛后更稳定。在仿真实验部分,设计了完整的实验方案,通过对比实验和泛化实验,验证了算法的有效性和优越性。

1 问题建模

1.1 问题描述

在建模阶段,简化的四旋翼无人机模型被用来描述多智能体在复杂环境中的避障行为。如图 1 所示,假设在任务场景中有 1 个目标点(橙色)、 N 架无人机组成的无人机群和 M 个障碍物(蓝色)。避障任务要求无人机群能够有效地穿越障碍物所在区域,穿越障碍物后在目标地点附近包围目标点。因此,每架无人机的目标是控制其速度和方向,避免其与障碍物和其他无人机的碰撞,最终完成在复杂环境中从出发点导航到目标点的任务。为更好地研究多智能体之间的避碰问题,无人机设定了初始位置,并且所有无人机都在同一高度飞行。

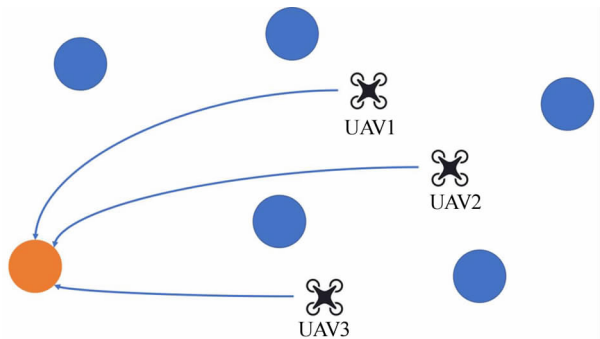


图 1 无人机避碰示意图

综上,多智能体系统的最优路径规划可以被描述为:

$$\min \sum_{i=1}^N D_i \text{ s. t. } \begin{cases} \|p_i - p_j\| > d_{a,s}, i, j = 1, 2, \dots, N \\ \|p_i - p_o\| > d_{ob,s}, i = 1, 2, \dots, N, \\ o = 1, 2, \dots, M \end{cases}$$

其中 D_i 为智能体 i 到目标点的移动距离; p_i 为智能体 i 的位置; d_{a_s} 为智能体之间的安全距离; p_o 为障碍物的位置; d_{ob_s} 为智能体和障碍物之间的安全距离。

1.2 智能体建模

在问题分析时,无人机 $i(i=1 \sim N)$ 被看作是一个智能体模型。智能体 i 的动力学模型如下:

$$\begin{cases} v_{i,x}^{t+1} = v_{i,x}^t + a_{i,x} \cdot t_{\text{step}} \\ v_{i,y}^{t+1} = v_{i,y}^t + a_{i,y} \cdot t_{\text{step}} \\ x_i^{t+1} = x_i^t + v_{i,x}^{t+1} \cdot t_{\text{step}} \\ y_i^{t+1} = y_i^t + v_{i,y}^{t+1} \cdot t_{\text{step}} \end{cases} \quad (1)$$

其中 t_{step} 为时间步长; $(v_{i,x}^t, v_{i,y}^t)$ 为智能体的速度; (x_i^t, y_i^t) 为智能体的位置; $(a_{i,x}, a_{i,y})$ 为智能体的水平加速度和竖直加速度,其控制智能体的运动。

假设 1:在飞行过程中,所有无人机都是在同一个组网下,无人机之间可以互相通信,每架无人机都可以获取相邻无人机的位置等信息。

2 MATD3 算法

多无人机在复杂环境中的避障行为被描述成一个基于部分可见马尔可夫决策过程的强化学习问题,并运用多智能体强化学习来解决这个问题。在强化学习的框架下,智能体通过状态、动作、奖励与环境进行交互,在交互过程中学习策略以达成长期奖励最大化或实现特定目标。多智能体强化学习的主要组成部分包括观测空间、动作空间、奖励函数以及算法框架。

2.1 观测空间

在数值仿真中,设置了目标位置 $p_d(x, y)$;智能体 i 可以观测到相邻智能体 j 的位置 $p_j(x_j, y_j)$;由于障碍物的位置是未知的且较远处的障碍物对智能体的运动影响较小,故取智能体 i 的观测范围内最近的三个障碍物的位置 $p_o(x, y)$ 。

对智能体 i 观测到的信息采取进一步处理,其观测空间由四个部分组成:

- (1)智能体 i 和目标点的相对位置;
- (2)智能体 i 和相邻智能体 j 的相对位置;
- (3)智能体 i 和观测到的三个障碍物的相对位置;
- (4)智能体 i 的位置 $p_i(x_i, y_i)$ 和速度 $(v_{i,x}, v_{i,y})$ 。

2.2 动作空间

智能体的动作空间是一个由加速度 (a_x, a_y) 组成的连续的空间。将得到的加速度 (a_x, a_y) 代入动力学模型,可以计算出速度和位移,控制智能体的运动。

2.3 奖励函数

奖励函数是强化学习的重要组成部分,根据人工势场法的思想设计的稠密集体奖励函数和碰撞奖励函数,可以加快智能体训练速度,奖励收敛后更加平滑和稳定。

2.3.1 集体奖励函数

集体奖励函数由所有智能体和目标点的距离定义,根据人工势场法的思想设计了此奖励函数。即使智能体没有找到问题的最优解决方案,奖励函数也能提供积极的反馈。集体奖励函数 $r_{\text{dis_all}}$ 如下:

$$r_{\text{dis_all}} = - \sum_{i=1}^N d_i^2 \quad (2)$$

其中 d_i 代表智能体 i 和目标点 $p_d(x, y)$ 之间的距离。

2.3.2 靠近目标奖励函数

在仿真中设置了目标边界 $d_{\text{arr_edge}}$,当智能体 i 和目标点的距离 d_i 小于 $d_{\text{arr_edge}}$ 时,就意味着接近目标点。靠近目标奖励函数 r_{arr} 如下:

$$r_{\text{arr}} = 7 \sin(1.25\pi \cdot d_i + 3\pi/8) + 7, d_i < d_{\text{arr_edge}} \quad (3)$$

注 1:此奖励函数在边界处和目标点的导数为 0,设计的初衷是帮助智能体快速靠近目标点和保证奖励函数在空间上的稠密性。

2.3.3 智能体间碰撞奖励函数

在仿真中设置了智能体之间的安全距离 d_{a_s} ,当智能体 i 和 j 之间的距离 $d_{i,j}$ 小于 d_{a_s} 之后就会有惩罚,此奖励函数同样是根据人工势场法的设计。奖励函数 r_{a_c} 如下:

$$r_{a_c} = -\alpha_{c_p} \cdot (1/d_{i,j} - 1/d_{a_s})^2, d_{i,j} < d_{a_s} \quad (4)$$

其中 α_{c_p} 是可调节的碰撞参数。

2.3.4 智能体和障碍物间碰撞奖励函数

在仿真中设置了智能体和障碍物之间的安全距离 d_{ob_s} ,当智能体 i 与障碍物之间的距离 d_{i_ob} 小于 d_{ob_s} 之后就会有惩罚,此奖励函数同样是根据人工势场法的设计。智能体和障碍物的碰撞奖励函数 r_{ob_c} 如下:

$$r_{ob_c} = -\alpha_{c_p} \cdot (1/d_{i_ob} - 1/d_{ob_s})^2, d_{i_ob} < d_{ob_s} \quad (5)$$

最后,总奖励函数如下:

$$R = r_{dis_all} + r_{arr} + r_{a_c} + r_{ob_c} \quad (6)$$

2.4 算法介绍

2.4.1 TD3 算法

TD3 算法更新框架如图 2 所示。TD3 算法与 DDPG 算法相比,有三个重要改进:(1)采用两套 Critic 网络,计算目标值时二者取其小,从而抑制 Q 值过估计;(2)计算目标值时,在下一个状态的动作上添加策略噪声,平滑策略期望;(3)在多次更新 Critic 网络后更新 Actor 网络,保证 Actor 网络的训练稳定性。

Critic 1 和 Critic 2 网络更新过程:在训练阶段,从经验回放池中采样数据 (O_t, a_t, r_t, O_{t+1}) ,利用目标 Actor 网络计算出状态 O_t 下的动作 $\pi(O_{t+1})$,然后基于目标策略平滑正则化,在动作 $\pi(O_{t+1})$ 上加入噪声 ϵ

$$\pi(O_{t+1}) = \pi(O_{t+1}) + \epsilon \quad (7)$$

其中 $\epsilon \sim \text{dip}(N(0,1), -c, c)$, c 为随机噪声参数。接着基于双重网络思想,取两套 Critic 网络中的较小值,计算目标值

$$m = r_t + \gamma \min_{i=1,2} Q_i(O_{t+1}, \pi(O_{t+1})) \quad (8)$$

其中 γ 是折扣因子。然后利用梯度下降法最小化评估值和目标值之间的误差:

$$L = 1/N \sum (Q_i(O_t, a_t) - m), i = 1, 2 \quad (9)$$

从而对 Critic 1 和 Critic 2 网络中的参数进行更新。

Actor 网络更新过程:在 Critic 1 和 Critic 2 网络更新 d 步之后,开始 Actor 网络更新,Actor 网络的目标是选出最佳动作,按 DPG 定理:

$$\nabla_{\phi} J(\phi) =$$

$$1/N \sum \nabla_a Q(O_t, a_t) \big|_{a=\pi_{\phi}(O)} \nabla_{\phi} \pi_{\phi}(O_t) \quad (10)$$

进行更新,更新方向为最大化 Q 值方向。

目标网络的更新过程:目标网络采用软更新的方式进行更新,将旧的目标网络参数和新的对应网络参数做加权,然后赋值给目标网络:

$$\begin{aligned} \theta'_i &= \tau \theta_i + (1 - \tau) \theta'_i, i = 1, 2 \\ \phi' &= \tau \phi + (1 - \tau) \phi' \end{aligned} \quad (11)$$

其中 θ' 是目标 Critic 网络参数; θ 是 Critic 网络参数; ϕ' 是目标 Actor 网络参数; ϕ 是 Actor 网络参数; τ 是目标网络更新率。

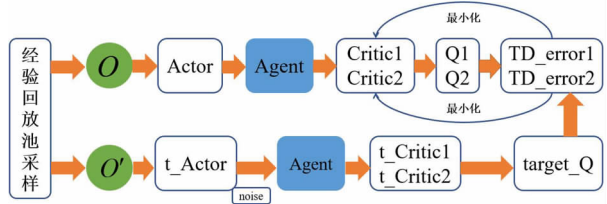


图 2 TD3 算法更新框架

TD3 算法伪代码如下:

算法 1 TD3 算法

```

随机初始化 Critic 网络和 Actor 网络参数  $\theta_1, \theta_2, \varphi$ 
初始化目标网络参数  $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2, \varphi' \leftarrow \varphi$ 
初始化经验回放池 B
for episode = 1 to MaxEpisode do
    初始化仿真环境
    for timestep = 1 to MaxStep do
        智能体获得初始观测空间  $O_0$ 
        智能体根据观测空间采取动作  $a_t$ 
        获取奖励  $r_t$ , 生成新的环境
        获取新的观测空间  $O_{t+1}$ 
        将经验  $(O_t, a_t, r_t, O_{t+1})$  储存在 B 中
    end for
    当 B 达到容量上限后采样, 大小为 Batchsize
    根据式(7)计算目标动作  $a'$ 
    根据式(8)计算目标 Q 值
    根据式(9)更新 Critic 1 和 Critic 2 网络
    d 步后
    根据式(10)更新 Actor 网络
    根据式(11)更新目标网络
end for

```

2.4.2 多智能体强化学习算法框架

“集中式训练,分布式执行”^[16]是当下流行的一种多智能体强化学习算法结构。如图 3 所示,算法中有 N 个相同的 Critic 网络和 N 个不同的 Actor 网络——对应。在训练阶段,算法采用集中式结构进行训练,即所有智能体的动作和观测被当作一个整体,构造更加完整的联合动作空间和联合观测空间,每个 Critic 网络更新时可以获取联合动作空间和联合观测空间的所有信息。在 Critic 网络更新多次后,Critic 网络再指导对应的 Actor 网络进行更新,即每个 Critic 网络输出智能体的 Q 值,智能体在接收到 Q 值后,结合本身的 Actor 网络往 Q 值最大的方向更新参数。而在测试阶段,智能体则分布式执行策略,即不需要 Critic 网络,智能体根据自身观测和有限信道传来的其他智能体的信息进行动作选择,进而与环境交互。

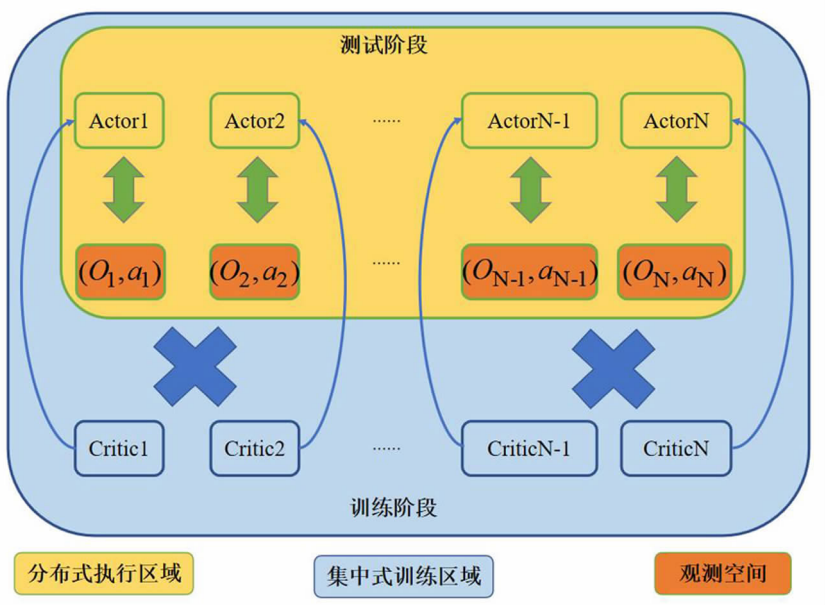


图 3 多智能体强化学习算法框架

3 仿真及对比实验

为验证算法的有效性,设计了三组对照试验,分别是:(1)基于稠密奖励函数和稀疏奖励函数的对比实验;(2)基于智能体数量增加的泛化实验;(3)与其他主流算法的对比实验。

3.1 仿真环境

实验以多智能体避障和到达目标点的行为为背景,使用 Python 进行数值仿真来验证算法的有效性。仿真画面为正方形地图,地图上有一个固定位置的目标点;有若干障碍物,其在一个完整的训练周期中,每回合障碍物的位置随机摆放;有若干固定初始位置的智能体。仿真环境的具体参数如表 1 所示:

表 1 仿真环境参数

参数	值
智能体个数 N	3
障碍物个数 M	8
正方形地图边长	4
智能体半径	0.02
障碍物半径	0.03
智能体间安全距离 d_{a_s}	0.07
智能体与障碍物间安全距离 d_{ob_s}	0.08

实验是在 Ubuntu 系统上运行,服务器为专门用于深度学习的戴尔 Poweredge r750xa 双路机架服务器主机。MATD3 算法的训练超参数如表 2 所示:

表 2 MATD3 算法超参数

超参数	值
网络学习率 lr	0.001
折扣因子 γ	0.95
经验回放池 B	1000000
采样批次 Batchsize	1024
最大回合数 MaxEpisode	25000
最大步数 MaxStep	45
目标网络更新率 τ	0.01
随机噪声 c	0.1
网络神经元个数 units	256/512
网络更新步数 d	45

3.2 奖励函数对比实验

为验证基于人工势场法思想设计的稠密奖励函数的有效性,其与稀疏奖励函数进行对比实验。实验中,MATD3 算法的网络层数为 3 层,稀疏奖励设置为:当智能体间发生碰撞和智能体碰撞障碍物后 $r = -20$ 。实验结果如图 4 所示,可以看出在相同的参数和环境下,两种奖励函数都可以让智能体完成实验任务。分析表 3 的基本性能指标可知,

稠密奖励函数与稀疏奖励函数相比,算法稳定需要的回合数减少 36.67%, 稳定所需时间减少 34.19%, 拥有稠密奖励函数的算法稳定后的奖励略高于稀疏奖励函数。MATD3 算法在稠密奖励函数的加持下不仅获得了更多的奖励, 而且其收敛速度比稀疏奖励函数更快, 奖励收敛后波动更小。图 5 为仿真环境示意图。

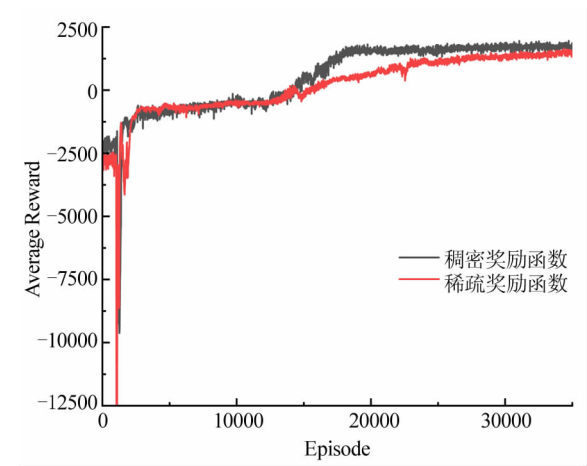


图 4 每 10 回合平均奖励对比

增加的泛化实验来验证算法的泛化能力。在实验中不仅增加了智能体的数量, 而且在完整训练周期中每回合的障碍物位置都是随机摆放, 以此来验证算法的泛化能力, 实验结果如图 6 所示(截取奖励大于-20000 以上部分画图)。从实验结果可以看出, 设计的算法不仅能够完成实验任务, 而且奖励收敛依然稳定。

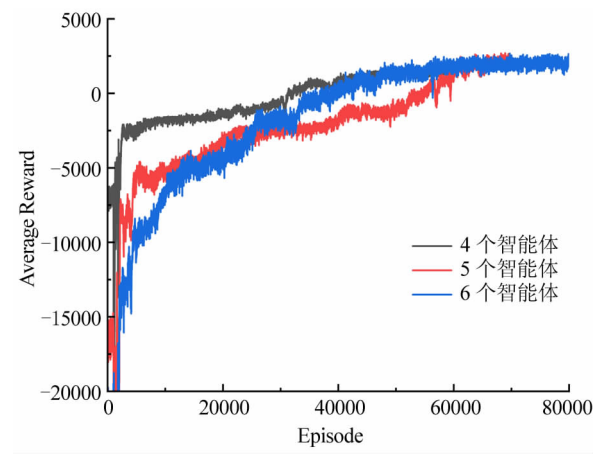


图 6 每 10 回合平均奖励对比

3.4 不同算法对比实验

为了加强实验对比, 在每回合障碍物位置随机的基础上, 在相同超参下将 MATD3 算法与 MADDPG 算法、MASAC 算法和 MAD4PG 算法这些结构相似的 A-C 算法进行对比, 实验结果如图 7 所示(截取奖励大于-10000 以上部分画图)。可以看出 MATD3 算法与其它算法相比不仅收敛速度快, 奖励收敛后更稳定, 而且收敛后能获得最高的奖励, 体现出相较 MADDPG 算法、MASAC 算法和 MAD4PG 算法的优越性。

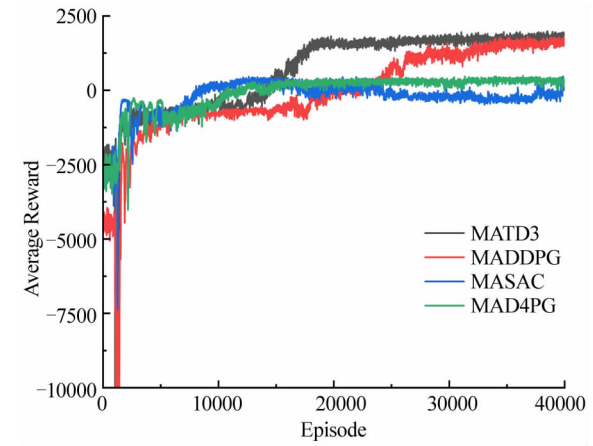


图 7 每 10 回合平均奖励对比

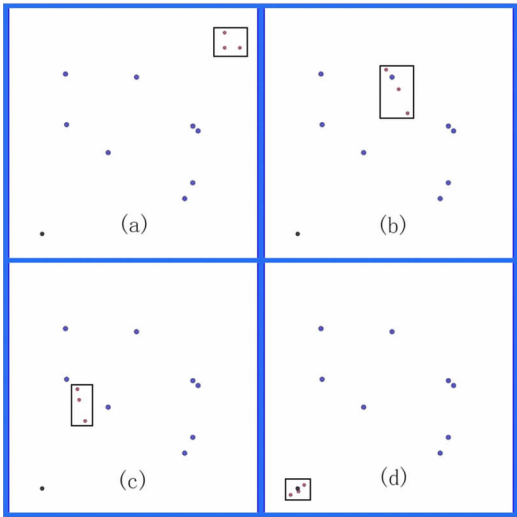


图 5 仿真环境示意图

表 3 基本性能指标

	稳定所需 回合数	稳定所需 时间	收敛后 平均奖励
稠密奖励	约 19000	约 3926.56s	1684.88
稀疏奖励	约 30000	约 5966.17s	1514.94

3.3 泛化实验

为验证算法的泛化性, 设计了基于智能体数量

4 结 论

在 MATD3 算法的框架下研究了考虑智能体间避碰的多智能体避障控制问题。依据人工势场法的思想设计了稠密奖励函数,为智能体提供积极的反馈,加快训练速度,奖励收敛后更加平滑。通过三组对比实验和泛化实验验证了算法的有效性。

参考文献

[1] 攀登. 纳卡冲突及背后的博弈[J]. 军事文摘, 2021, 481(1): 7—11.

[2] 李鹏举, 毛鹏军, 耿乾, 等. 无人机集群技术研究现状与趋势[J]. 航空兵器, 2020, 27(4): 25—32.

[3] REZA O, RICHARD M M. Distributed cooperative control of multiple vehicle formations using structural potential functions [J]. IFAC Proceedings Volumes, 2002, 35(1): 495—500.

[4] SEN M K, DATTA -GUPTA A, STOFFA P L, et al. Stochastic reservoir modeling using simulated annealing and genetic algorithm[J]. SPE Formation Evaluation, 1995, 10(1): 49—56.

[5] TOMAS L P, WESLEY M. An algorithm for planning collision-free paths among polyhedral obstacles [J]. Commun. ACM, 1979, 22(10): 560—570.

[6] HART P, NILSSON N, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100—107.

[7] BELLMAN R. A markovian decision process[J]. Indiana University Mathematics Journal, 1957, 6(4): 679—684.

[8] WATKINS C. Learning from delayed rewards[D]. London: University of Cambridge, 1989.

[9] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529—533.

[10] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C] //International Conference on Learning Representations (ICLR), 2016.

[11] FUJIMOTO S, HOOFF H, MEGER D. Addressing function approximation error in actor-critic methods[C]// Proceedings of the 35th International Conference on Machine Learning (PMLR), 2018: 1587—1596.

[12] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [C]// Advances in Neural Information Processing Systems (NeurIPS), 2017: 6379—6390.

[13] YAN P, BAI C, ZHENG H, et al. Flocking control of uav swarms with deep reinforcement learning approach[C] //2020 3rd International Conference on Unmanned Systems (ICUS), 2020: 592—599.

[14] ZHANG Y, WU Z, MA Y, et al. Research on autonomous formation of Multi-UAV based on MADDPG algorithm [C]//2022 IEEE 17th International Conference on Control & Automation (ICCA), 2022: 249—254.

[15] GUTPA A, NALLANTHIGHAL R. Decentralized multi-agent formation control via deep reinforcement learning [C]//International Conference on Agents and Artificial Intelligence (ICAART), 2021: 289—295.

[16] SUN C Y, MU C X. Important scientific problems of multi-agent deep reinforcement learning[J]. Acta Automatica Sinica, 2020, 46(7): 1301—1312.