文章编号:1674-2974(2022)06-0101-09 DOI:10.16339/j.cnki.hdxbzkb.2022290

基于元强化学习的无人机自主避障与目标追踪

江未来[1,2],吴俊[1,2†],王耀南[1,2]
(1. 湖南大学 电气与信息工程学院,湖南 长沙 410082;
2. 湖南大学 机器人视觉感知与控制技术国家工程研究中心,湖南 长沙 410082)

摘 要:针对传统深度强化学习在求解无人机自主避障与目标追踪任务时所存在的训练效率低、环境适应性差的问题,在深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)算法中融入与模型无关的元学习(Model-Agnostic Meta-Learning, MAML),设计一种内外部元参数更新规则,提出了元深度确定性策略梯度(Meta-Deep Deterministic Policy Gradient, Meta-DDPG)算法,以提升模型的收敛速度和泛化能力.此外,在模型预训练部分构造基本元任务集以提升实际工程中的预训练效率.最后,在多种测试环境下对所提算法进行了仿真验证,结果表明基本元任务集的引入可使模型预训练效果更优,Meta-DDPG 算法相比 DDPG 算法在收敛特性和环境适应性方面更有优势,并且元学习方法和基本元任务集对确定性策略强化学习具有通用性.

关键词:元强化学习;无人机;自主避障;目标追踪;路径规划
中图分类号:V249.1 **文献标志码**:A

Autonomous Obstacle Avoidance and Target Tracking of UAV Based on Meta-Reinforcement Learning

JIANG Weilai[1,2], WU Jun[1,2†], WANG Yaonan[1,2]
(1. College of Electrical and Information Engineering, Hunan Unviersity, Changsha 410082, China;
2. National Engineering Research Center of Robot Visual Perception & Control Technology, Hunan University, Changsha 410082, China)

Abstract: There are some problems with traditional deep reinforcement learning in solving autonomous obstacle avoidance and target tracking tasks for unmanned aerial vehicles (UAV), such as low training efficiency and weak adaptability to variable environments. To overcome these problems, this paper designs an internal and external meta-parameter update rule by incorporating Model-Agnostic Meta-Learning (MAML) into Deep Deterministic Policy Gradient (DDPG) algorithm and proposes a Meta-Deep Deterministic Policy Gradient (Meta-DDPG) algorithm inovder to improve the convergence speed and generalization ability of the model. Furthermore, the basic meta-task sets are constructed in the model's pre-training stage to improve the efficiency of pre-training in practical engineering. Finally, the proposed algorithm is simulated and verified in Various testing environments. The results show that

* **收稿日期**:2021-07-16
基金项目:国家自然科学基金资助项目(61903133,61733004), National Natural Science Foundation of China(61903133,61733004); 国家重点研发计划重点专项项目(2021YFC1910400), Project of the National Key Research and Development Program of China(2021YFC1910400); 江苏省重点研发计划项目(BE2020082-1), Key Research and Development Program of Jiangsu Province(BE2020082-1)
作者简介:江未来(1989—),男,江西上饶人,湖南大学副教授,博士生导师
† 通信联系人,E-mail: znova@hnu.edu.cn

the introduction of the basic meta-task sets can make the model's pre-training more efficient, Meta-DDPG algorithm has better convergence characteristics and environmental adaptability when compared with the DDPG algorithm. Furthermore, the meta-learning and the basic meta-task sets are universal to deterministic policy reinforcement learning.

Key words: meta-reinforcement learning; Unmanned Aerial Vehicle (UAV); autonomous obstacle avoidance; target tracking; path planning

随着卫星导航、信号传输、电气储能等相关技术的进步,无人机的应用领域在不断扩大,如森林防火、电力巡检、物流运输等.这些任务的基本前提均为无人机目标追踪,只有追上目标或到达指定地点才可以继续执行任务.无人机执行目标追踪任务时不可避免地会遇到障碍物,例如房屋、树木、电线等.如何让无人机安全自主地避开障碍物并实现目标追踪是无人机领域一大研究热点.

传统避障算法有蚁群算法[1]、最短路径制导向量场[2]和贝叶斯推理等.这些算法都是将避障问题转换为优化问题,通过求解优化模型而得到最终的无人机飞行轨迹.但是这些方法由于存在迭代时间长、泛化能力弱、智能化水平低等缺点,无法适用于环境多变或环境未知下的避障问题.随着人工智能技术发展,深度强化学习逐渐被运用于求解无人机自主避障与目标追踪问题.文献[3-5]基于深度Q网络(Deep Q Net, DQN)[6]算法完成无人机离散动作空间下路径规划.文献[7-8]采用深度确定性策略梯度(Deep Deterministic Policy Gradient, DDPG)[9]算法实现无人机连续动作空间下目标追踪的自主决策.

虽然上述算法均取得了一定的成果,但是传统深度强化学习算法训练速度慢,且只能应对单一环境下的任务,而当障碍物环境或目标运动轨迹改变时,智能体往往需要重新进行大量探索和训练.因此如何提升深度强化学习算法面对复杂动态任务时的收敛速度和适应性成为强化学习领域的一大热点和难点.

近年来,有学者将元学习与深度强化学习相结合,提出了元强化学习概念.元学习主张让机器学习如何学习,人类之所以比机器更智能是因为当遇到一个新任务时,人类知道怎么在短时间内得出执行任务的要领.Finn等[10]提出与模型无关的元学习(Model-Agnostic Meta-Learning, MAML),可适用于任何采用梯度下降更新方法的机器学习算法.Wang等[11]首次将长短期记忆网络与强化学习结合,使得神经网络具有能够独立训练任务的能力.Xu等[12]提出在深度强化学习神经网络中添加嵌入层对上下文潜在变量进行元训练以提高分布式数据挖掘的效率.然而,发挥元强化学习可根据新任务自主适应的优势,用以解决复杂动态环境下的无人机自主避障与目标追踪问题鲜有报道.

综上,为解决传统深度强化学习在求解无人机自主避障与目标追踪任务时收敛特性差、环境适应性弱的问题,本文提出了一种元深度确定性策略梯度(Meta-Deep Deterministic Policy Gradient, Meta-DDPG)算法.将元学习算法MAML与深度强化学习算法DDPG相结合,在预训练过程中设计内外部元参数更新规则,获取可以适应多种任务的元初始参数.此外,构造基本元任务集运用于Meta-DDPG算法预训练阶段.最后仿真结果表明,采用基本元任务集使得工程应用更加高效,Meta-DDPG算法与DDPG算法相比具有更优的收敛特性与环境适应性,并且元学习方法和基本元任务集对确定性策略强化学习算法具有较高的通用性.

1 问题描述

1.1 无人机运动模型

本文重点讨论无人机在执行自主避障与目标追踪任务时的决策问题,故将无人机视为二维空间下的运动模型,使用水平与垂直方向的加速度来控制无人机的运动,如图1所示[13].图中,(x_t, y_t)为无人机t时刻的位置坐标;$(\hat{x}_t, \hat{y}_t)$为无人机通过GPS等设备获取目标t时刻的位置坐标;$\hat{d}_{\max}$为无人机利用避障传感器感知环境的最大欧氏距离;v_t为无人机t时刻的飞行速度;$\hat{v}_t$为目标t时刻的运动速度;d_t为无人机t时刻与目标之间的欧氏距离;$\hat{d}_t$为无人机t时刻与障碍物之间的直线距离;v_{xlt}、v_{ylt}分别表示t时刻无人机水平与垂直方向上的飞行速度.

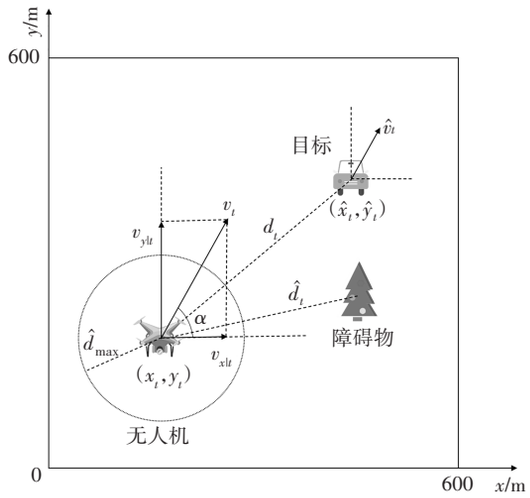


图 1　无人机的运动学模型

Fig.1　Kinematic model of UAV

考虑到实际情况中无人机速度不能瞬间变化,故无人机运动方程可表示为

$$\begin{cases} v_{xlt+1} = v_{xlt} + n_t t \cdot \cos\alpha \\ v_{ylt+1} = v_{ylt} + n_t t \cdot \sin\alpha \end{cases} \tag{1}$$

$$\begin{cases} x_{t+1} = x_t + v_{xlt} \\ y_{t+1} = y_t + v_{ylt} \end{cases} \tag{2}$$

式中:n_t为无人机t时刻的加速度大小;α为加速度方向与水平线的夹角.

1.2　无人机自主避障与目标追踪任务建模

为了更好地描述无人机自主避障与目标追踪任务,将其定义为马尔可夫决策过程(Markov decision process, MDP). MDP由状态空间S、动作空间A、状态转移概率P、奖励函数R和折扣因子γ组成,并以元组表示为(S, A, P, R, γ). 在该任务中状态空间S为无人机的本体状态与传感器采集的环境信息;动作空间A为无人机采取的追踪动作;状态转移概率$P[s_{t+1}|s_t, a_t]$为状态s_t下执行动作a_t转移到s_{t+1}的概率;奖励函数R为在状态s_t下采取动作a_t,无人机可以获得的即时奖励,即$R(s_t, a_t)$;折扣因子γ为未来奖励对当前状态的影响因素. 在此定义动作值函数的贝尔曼方程为

$$Q_\pi(s,a) = \mathrm{E}_\pi[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|S_t = s, A_t = a] \tag{3}$$

式中:π表示智能体所采取的动作序列,称为策略;

$Q_\pi(s,a)$表示在状态s处,采取动作a后,所得到的折扣累计奖励的期望. 根据$Q_\pi(s,a)$值大小可评估策略π的优劣.

1.2.1　状态空间S

状态空间S为智能体自身状态和环境信息的集合. 在该任务中设状态空间S由无人机所在位置坐标(x_t, y_t)、目标所在位置坐标$(\hat{x}_t, \hat{y}_t)$、无人机与目标之间的欧氏距离d_t、传感器范围内无人机与障碍物的欧氏距离$\hat{d}_t$和无人机的速度v_t共7个数据组成,并对数据进行归一化.

$$x_t' = \frac{x_t}{d_{\max}}, y_t' = \frac{y_t}{d_{\max}}, \hat{x}_t' = \frac{\hat{x}_t}{d_{\max}}, \hat{y}_t' = \frac{\hat{y}_t}{d_{\max}},$$

$$d_t' = \frac{d_t}{d_{\max}}, \hat{d}_t' = \frac{\hat{d}_t}{\hat{d}_{\max}}, v_t' = \frac{v_t}{v_{\max}} \tag{4}$$

式中,$x_t', y_t', \hat{x}_t', \hat{y}_t', d_t', \hat{d}_t', v'$为最终归一化结果;$d_{\max}$为在追踪范围内无人机距离目标的最远欧氏距离;$v_{\max}$为无人机的最大飞行速度.

最终状态空间S记作

$$\boldsymbol{S} = [x_t', y_t', \hat{x}_t', \hat{y}_t', d_t', \hat{d}_t', v']^{\mathrm{T}} \tag{5}$$

1.2.2　动作空间A

动作空间A为智能体可执行的动作. 由于无人机速度不能瞬间变化,所以动作空间由加速度大小n和加速度方向与水平线的夹角α组成,同样进行归一化为

$$n' = \frac{n}{n_{\max}}, n \in [-n_{\max}, n_{\max}],\ \alpha' = \frac{\alpha}{\pi}, \alpha \in [-\pi, \pi] \tag{6}$$

式中,$n_{\max}$为无人机最大加速度.

所以动作空间A记作

$$\boldsymbol{A} = [n', \alpha']^{\mathrm{T}} \tag{7}$$

深度强化学习算法最终目标是获得最优策略π,即在任意状态s下所执行的动作a.

1.2.3　奖励函数R

奖励函数的设定对深度强化学习的训练结果至关重要,不同的奖励函数对模型收敛特性影响都不同. 此任务中,若采用稀疏奖励,也即只在无人机追踪成功或失败后才反馈奖励,会造成收敛速度缓慢的问题. 因此本文设置连续奖励函数为

$$r_1 = \begin{cases} 6000, \text{当无人机追上目标} \\ -1000, \text{当无人机或目标离开追捕范围} \\ -1500, \text{当无人机撞上障碍物} \end{cases} \tag{8}$$

$$r_2 = \begin{cases} \hat{d}*0.1, \text{当无人机感知范围内存在障碍物} \\ 70, \text{当无人机感知范围内无障碍物} \end{cases} \tag{9}$$

$$r_3 = -d*0.1 \tag{10}$$

$$r = r_1 + r_2 + r_3 \tag{11}$$

式中:r_1为追踪奖励;r_2为避障奖励;r_3为距离奖励;r为总奖励;$\hat{d}$为无人机感知范围内障碍物的欧氏距离;d为无人机与目标之间的欧氏距离.

2 DDPG算法

DDPG是一种处理连续状态空间和动作空间问题的确定性策略强化学习算法.传统演员-评论家(Actor-Critic, AC)算法中演员网络与评论家网络在训练时往往不稳定.DDPG算法针对此问题,分别构建了一对结构完全相同的评估(Eval)神经网络和目标(Target)神经网络.其中Eval神经网络用于训练更新网络参数,Target神经网络采用软更新的方式来跟随Eval神经网络参数,保证训练过程的稳定性.

对于演员Eval网络,可训练参数为θ,输入为状态s,输出为动作a.演员Eval神经网络损失函数为

$$loss_{\text{actor}} = -Q_{\pi_\theta}(s,a) \tag{12}$$

式中:$Q_{\pi_\theta}(s,a)$为在状态s处根据策略π_θ得到的动作值函数.采用梯度下降法,使$Q_{\pi_\theta}(s,a)$尽可能最大化.

对于评论家Eval网络,可训练的参数为w,输入为状态s和动作a,输出为动作值$Q_{\pi_\theta}(s,a)$.利用时间差分误差(TD-error)对神经网络进行训练,评论家Eval神经网络损失函数为

$$loss_{\text{critic}} = (r(s,a) + \gamma Q_{\pi_\theta}(s', \pi_{\theta'}(s');\ w') - Q_{\pi_\theta}(s, \pi_\theta(s);\ w))^2 \tag{13}$$

式中:$r(s,a)$为在状态s处采取动作a可获得的即时奖励;$Q_{\pi_\theta}(s', \pi_{\theta'}(s');\ w')$为评论家Target神经网络给出的下一个时刻的行为值;w'为评论家Target神经网络参数;θ'为演员Target神经网络参数;s'为下一时刻的状态;$\pi_{\theta'}(s')$为演员Target神经网络输出的动作;$Q(s,\pi_\theta(s);\ w)$为评论家Eval神经网络给出的当前时刻行为值;w为评论家Eval神经网络参数;s和$\pi_\theta(s)$为当前时刻状态与动作;γ为折扣因子;$r(s,a) + \gamma Q_{\pi_\theta}(s', \pi_{\theta'}(s');\ w') - Q_{\pi_\theta}(s, \pi_\theta(s);\ w)$为TD-error.

Target神经网络采用式(14)周期性地进行软更新,其中τ是常数.

$$\begin{cases} \theta' = \tau\theta + (1-\tau)\theta' \\ w' = \tau\theta + (1-\tau)w' \end{cases} \tag{14}$$

3 MAML

元学习使智能体具有学会学习的能力[14].元学习的重点在于如何在模型中引入先验知识,并在训练过程中优化外部记忆,从而在训练新任务时更快更准确地学习.MAML与其他深度学习算法不同之处在于其不是寻找完成某个任务的最优参数,而是通过训练一系列与任务相关的元任务来寻找使模型在面对新任务时快速达到最优的初始参数η.η具有对新任务学习域分布的敏感特性,在面临新任务时可使训练模型内部的某些特征更容易地在多种任务之间相互转换,经过几步更新后即可获得最优的模型网络参数.MAML梯度下降过程如图2所示.图中,η表示经过MAML预训练后得到的初始化参数;L_1, L_2, L_3分别表示新任务的损失函数;∇表示梯度算子;$\eta_1^*, \eta_2^*, \eta_3^*$表示在新任务下的最优更新方向.

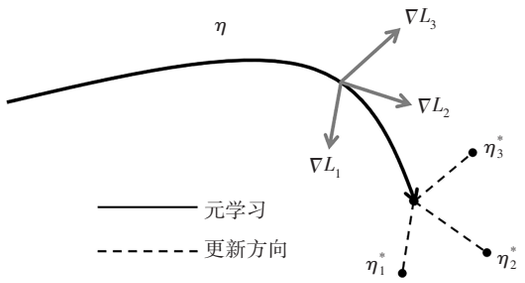


图2 MAML梯度下降过程

Fig.2 MAML gradient descent process

4 元强化学习

在深度强化学习的训练过程中,神经网络的不确定性往往会导致算法收敛特性较差,且训练的结果是一个仅适应当前任务和环境的策略.针对深度强化学习实施过程中存在的上述问题,本文在DDPG算法中引入MAML,提出一种元强化学习算法——Meta-DDPG算法.其基本思想是设计一种内外部元参数更新规则以获得一组元初始参数,提高模型面对不同任务的收敛速度和环境适应性.

4.1 基本元任务集

环境适应性是指模型面对一个新任务环境时经过少量训练便可获取正确策略的能力.元强化学习需要利用元任务集获得先验知识而提升模型的环境适应性,大部分元强化学习中的元任务集与实际任务场景相似[15-16].若把多种不同的复杂测试环境作为求解无人机自主避障与目标追踪问题的元任务集,使用Meta-DDPG算法进行预训练将花费大量时间,降低工程效率.为此,根据MAML中元任务的定义,将一个复杂多变的整体任务分解为多个只完成单一子任务目标的基本元任务,并将它们构成基本元任务集$T = \{T_1, T_2, \cdots, T_j\}$,其中$T_j$为第$j$个基本元

任务,同时为T中的每个基本元任务创建经验回放池$B_{T_j}=(s_t^{T_j},a_t^{T_j},r_t^{T_j},s_{t+1}^{T_j})$. Meta-DDPG 算法预训练过程中,智能体依次对T内每个基本元任务进行训练得到能够适应每个子任务的策略,最终获得学习整体任务的元初始参数.

在无人机自主避障与目标追踪任务中,基本元任务集中包含无人机追踪与无人机避障两个基本元任务,如图3所示. 在Meta-DDPG算法预训练中,首先学习无人机在无障碍物环境下静态目标追踪策略,然后学习无人机在简单障碍物环境下的自主避障策略,最终获得一个可以适应自主避障和目标追踪新任务的元初始参数. 由于基本元任务都较为简单,只需要较少幕数便可获取其中的先验知识,提高了预训练的效率.

4.2 Meta-DDPG 算法

Meta-DDPG算法分为预训练和整体任务训练两部分. 在预训练中,设计一种内外部元参数更新规则,内部网络训练和外部元参数更新以一定的频率交替进行. 内部网络依次学习各个基本元任务获得不同的内部参数,外部元参数更新通过优化不同的参数获得一个环境适应性较强的元初始参数. 在整体任务训练中,对于不同测试环境下无人机自主避障与目标追踪,Meta-DDPG算法仅通过少量训练幕数就能快速收敛,获取正确动作策略.

Meta-DDPG预训练中内部参数更新可描述为依次对每个基本元任务的训练过程,利用Meta-DDPG中Eval神经网络与Target神经网络配合不断更新获得适用于基本元任务的神经网络内部参数. 对于外部元参数更新,可描述为对基本元任务集的神经网络参数二次梯度优化过程. 外部周期性地对元参数进行更新,更新规则为

$$\begin{cases}\theta_{\text{meta}}=(1-n\tau)\theta_{\text{meta}}+\tau(\theta_{T_1}'+\theta_{T_2}'+\ldots+\theta_{T_n}')\\ w_{\text{meta}}=(1-n\tau)w_{\text{meta}}+\tau(w_{T_1}'+w_{T_2}'+\ldots+w_{T_n}')\end{cases}\tag{15}$$

式中:θ_{meta}为演员Target神经网络的外部元参数;w_{meta}为评论家Target神经网络的外部元参数;n为完成训练的基本元任务数量;τ为常数,控制元参数更新的速度;θ_{T_j}'为元任务T_j训练过程中的演员Target神经网络参数;w_{T_j}'为元任务T_j训练过程中的评论家Target神经网络参数. 预训练结束后,θ_{meta}和w_{meta}即为元初始参数. Meta-DDPG网络结构如图4所示.

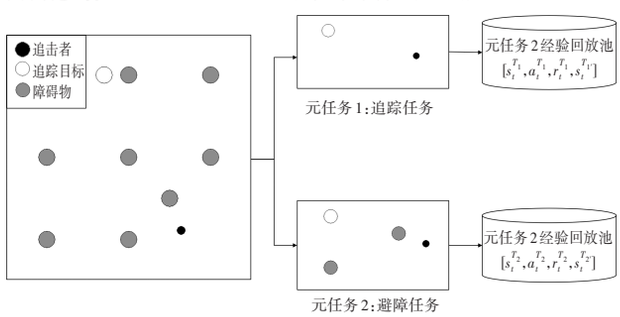


图3　基本元任务集构造

Fig.3　The construction of the basic meta-task sets

以基本元任务T_j为例,当Meta-DDPG网络内部

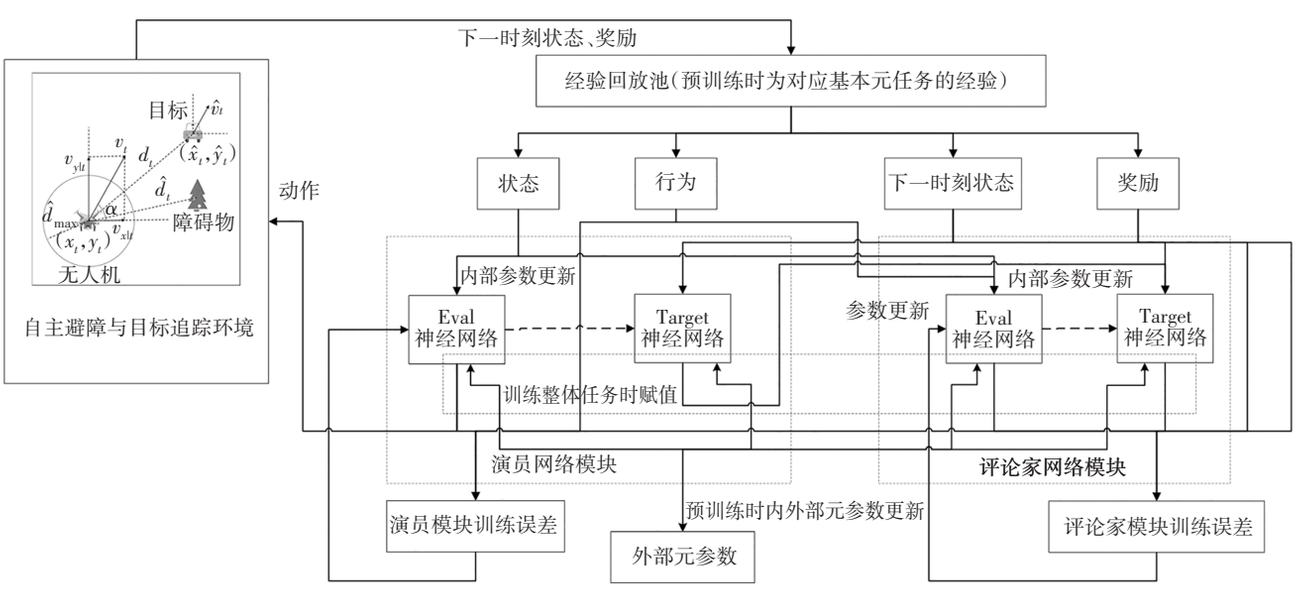


图4　Meta-DDPG 算法结构图

Fig.4　Meta-DDPG algorithm structure diagram

更新一定步数后外部元参数也进行更新.在每个基本元任务交替过程中,将外部元参数赋值给内部参数作为下一个基本元任务 T_{j+1} 的初始参数,直至遍历完基本元任务集后获得整体任务的元初始参数.预训练流程如算法1所示.

算法1 Meta-DDPG中预训练算法

Algorithm.1 Pre-training algorithm of META-DDPG

Meta-DDPG算法预训练部分具体步骤

1:需要:元任务集 $T=\{T_1, T_2, ..., T_n\}$

2:需要:训练超参数 τ,网络学习率 $lr_{\text{actor}}, lr_{\text{critic}}$

3:随机初始化演员网络和评论家网络参数 θ, w

4:初始化Target网络参数 $\theta' \leftarrow \theta, w' \leftarrow w$

5:初始化元参数 $\theta_{\text{meta}} \leftarrow \theta', w_{\text{meta}} \leftarrow w'$

6: for every T_j, j=1,2,..n do:

7: for episode=1, M do:

8:初始化经验回放池 B_{T_j}

9:初始化随机Ornstein-Uhlenbeck噪声 $\mathbb{N}$

10:接收初始观测状态 s_1

11: for t=1, T do:

12:根据当前策略和随机噪声选择动作 $a_t=\pi_\theta(s_t)+\mathbb{N}_t$

13:执行动作 a_t,观测奖励 r_t 和新状态 s_{t+1}

14:存储 (s_t, a_t, r_t, s_{t+1}) 至经验池 B_{T_j}

15:从 B_{T_j} 中随机采样 N 个经验数据 (s_i, a_i, r_i, s_{i+1})

16:利用评论家Target网络计算当前状态目标值:

$y_i=r_i+\gamma Q(s_{i+1}, \pi_{\theta'}(s_{i+1});\ w')$

17:通过最小化Loss函数

$$loss_{\text{critic}}=\frac{1}{N}(y_i-Q(s,\pi_\theta(s);\ w))^2$$

更新网络参数 w

18:利用采样梯度

$$\nabla_\theta loss_{\text{actor}}=\frac{1}{N}\sum\nabla Q(s_i, \pi_\theta(s_i))$$

更新演员网络参数 θ

19:软更新Target网络参数:

$\theta'=\tau\theta+(1-\tau)\theta'$

$w'=\tau\theta+(1-\tau)w'$

20:基本元任务内部参数:

$\theta_{T_j}'=\theta', w_{T_j}'=w'$

21: if t == meta_update_freq, do:

$\theta_{\text{meta}}=(1-n\tau)\theta_{\text{meta}}+\tau(\theta_{T_1}'+\theta_{T_2}'+...+\theta_{T_n}')$

$w_{\text{meta}}=(1-n\tau)w_{\text{meta}}+\tau(w_{T_1}'+w_{T_2}'+...+w_{T_n}')$

22: end for

23: end for

24:$\theta \leftarrow \theta_{\text{meta}}, w \leftarrow w_{\text{meta}}$

25:$\theta' \leftarrow \theta, w' \leftarrow w$

26: end for

通过以上算法可获得无人机自主避障与目标追踪任务的元初始参数 θ_{meta}、w_{meta}.训练整体任务时利用此参数初始化,模型可充分利用先验知识,仅需少量迭代便可收敛并获得完成任务的策略.后续对整体任务的训练过程与DDPG算法相同,在此不再赘述.另需指出的是,本文所提出的Meta-DDPG算法面对新任务时不必重复预训练,只需使用元初始参数进行网络初始化.

5 仿真结果与分析

使用Meta-DDPG算法求解无人机自主避障与目标追踪任务.设定追踪场景为600 m × 600 m的二维正方形领域,场景中存在多个障碍物,并且当目标感知无人机靠近时会产生逃逸动作.无人机可利用GPS等设备获取目标位置且通过传感器获取与障碍物的距离.当无人机自主避开障碍物并追踪到目标视为任务成功;当无人机撞上障碍物、无人机或目标离开正方形领域两种情况视为任务失败.

5.1 实验参数

设无人机与障碍物之间最大直线距离 $d_{\max}$ = 850m;无人机加速度大小 $n\in[-3\ \text{m/s}^2, 3\ \text{m/s}^2]$;无人机最大飞行速度 $v_{\max}$ = 20 m/s,最小飞行速度 $v_{\min}$ = 5 m/s;无人机避障传感器最大测量范围 $\hat{d}_{\max}$ = 50 m;预训练网络超参数 τ = 0.1,折扣因子 γ = 0.9;各元任务预训练幕数episode_meta=100;整体任务训练幕数episode=500;经验回放池容量为1 000;探索步数为1 000步;演员深度神经网络学习率 lr_{actor} = 0.000 1;评论家深度神经网络学习率 lr_{critic} = 0.000 5;外部元参数更新频率meta_update_freq = 10;采样经验数量 N = 32.

5.2 深度神经网络结构

根据式(5)知状态空间 S 包含7个参数,故演员深度神经网络为7维输入;由式(7)动作空间 A 包含2个参数,故为2维输出.评论家深度神经网络输入为当前状态与演员深度神经网络输出的动作,故为9维输入;输出为行为值 $Q(s, a)$,故为1维输出.由上可设演员和评论家深度神经网络结构分别为7×256×256×256×2和9×256×256×256×1.

演员深度神经网络中,输出动作均归一化至[-1, 1],输出层使用Tanh激活函数,其余层均使用Relu激活函数.评论家深度神经网络中,输出层为线性激活函数以确保行为值 $Q(s, a)$ 正常输出,其余层也均使用Relu激活函数.

5.3 实验结果

5.3.1 基本元任务集预训练效果验证

构造基本元任务集,将无人机自主避障与目标

追踪任务分解为无人机追踪与无人机避障两个基本元任务并分别构建经验回放池,如图 3 所示. 作为对比,将图 5 中两个复杂测试环境下动态目标追踪任务作为复杂元任务集. 使用 Meta-DDPG 算法,对两种元任务集各进行共 200 幕预训练. 整体任务为图 6(a)中测试环境(1)下的无人机自主避障与目标追踪.

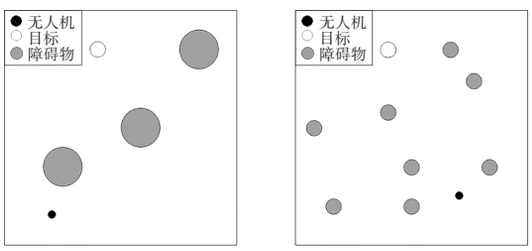


图 5　复杂元任务集

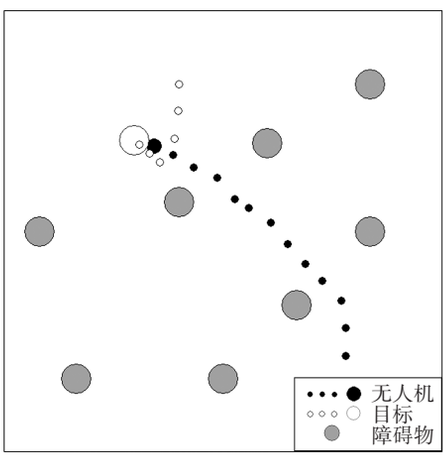
Fig.5　The complex meta-task sets

利用平均奖励值的收敛特性来衡量网络的预训练效果. 如图 6(b)所示,可知在每个元任务训练 100 幕的情况下,当采用基本元任务集时可以更充分地利用先验知识获得适应整体任务的元初始参数. 平均奖励值的上升速度与收敛特性都优于复杂元任务集. Meta-DDPG 算法整体任务测试结果如图 6(a)所示.

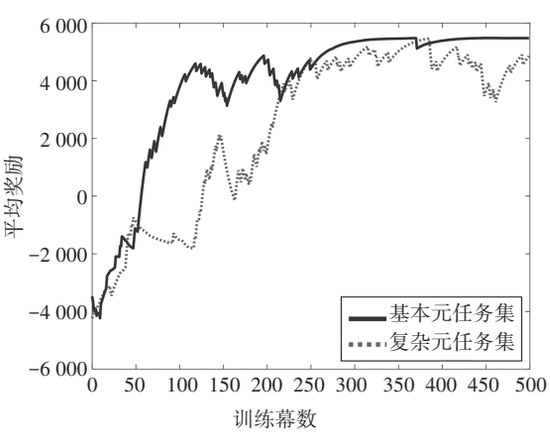
5.3.2　Meta-DDPG 收敛特性验证

使用 5.3.1 节中预训练获得的元初始参数,在图 7(a)测试环境(2)中进行 500 幕训练后测试. 为了更好地体现 Meta-DDPG 在收敛速度上的优势,使用不经预训练的 DDPG 算法与之比较. 利用平均奖励值的收敛特性和 Tensorboard 中演员 Eval 神经网络的 Loss 值来衡量算法的性能,仿真曲线分别如图 7(b)、图 7(c)所示.

由图 7(b)可知,使用 Meta-DDPG 算法时,平均奖励值在训练伊始就迅速上升,且经过 150 幕训练后逐渐达到收敛. 由图 7(c)知元初始参数可使演员 Eval 网络 Loss 值迅速下降,并在训练 120 幕后在一个较低的范围内波动. 使用 Meta-DDPG 训练 500 幕所得模型进行测试,测试结果如图 7(a)所示,由图知无人机可自主绕过障碍物并准确地追上逃逸的目标. 而 DDPG 算法由于先验知识缺失、探索效率低、经验样本质量差等原因,在较短的训练幕数与较少的经验池容量下陷入错误的局部最优,无法得到完成此任务的策略. 图 7(b)可知平均奖励曲线无法正确地收敛,平均奖励始终小于 0. 图 7(c)可知 DDPG 无法通过训练使演员 Eval 网络 Loss 函数梯度下降,loss 值始终大于 0.



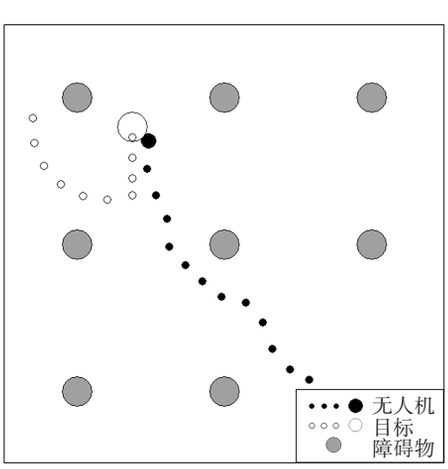
(a)Meta-DDPG 在测试环境(1)中测试结果



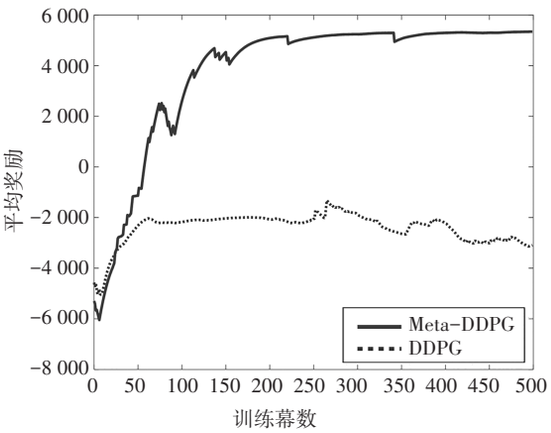
(b)测试环境(1)中平均奖励对比曲线

图 6　Meta-DDPG 在测试环境(1)中实验结果

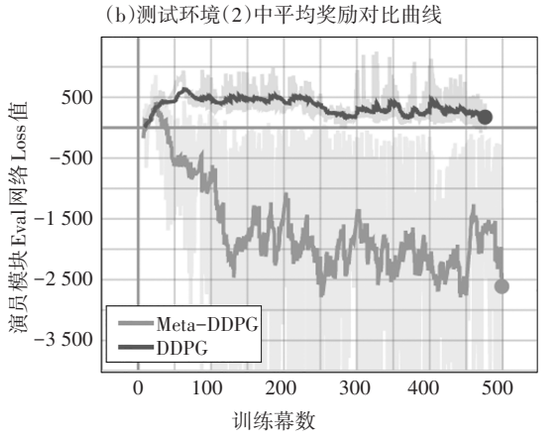
Fig.6　Experimental results of Meta-DDPG in the testing environment (1)



(a)Meta-DDPG 在测试环境(2)中实验结果



(b)测试环境(2)中平均奖励对比曲线



(c)演员 Eval 网络的 Loss 值对比曲线

图 7 Meta-DDPG 在测试环境(2)中实验结果

Fig.7 Experimental results of Meta-DDPG in the testing environment (2)

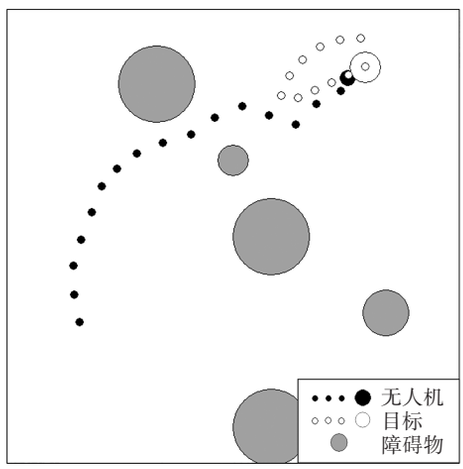
5.3.3 Meta-DDPG 环境适应性验证

为了突出 Meta-DDPG 算法的环境适应性,在如图 8(a)所示障碍物大小不同、目标运动轨迹不同的测试环境(3)中,使用与 5.3.2 节相同的元初始参数进行训练与测试. 作为对比,使用能够在测试环境(1)中完成任务的 DDPG 算法模型进行训练. 由图 8(b)可知,Meta-DDPG 算法的平均奖励在训练伊始就快速上升,150 幕后相对稳定,环境适应性较强. 而 DDPG 算法的平均奖励值在 350 幕才开始上升,且在 500 幕内尚未收敛. Meta-DDPG 算法整体任务测试结果如图 8(a)所示.

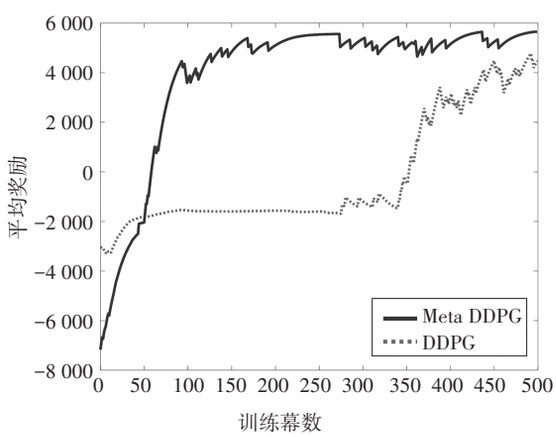
5.3.4 元学习方法与基本元任务集通用性验证

为了体现元学习方法和基本元任务集对确定性策略强化学习算法的通用性,将其运用于与 DDPG 算法同为确定性策略的双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient, TD3)[17]算法,构造 Meta-TD3 算法. 使用图 3 的基本元任务集预训练,并在测试环境(2)-(3)中对其收敛特性和环境适应性进行仿真验证,仿真结果见图 9.

由图 9(a)可知,Meta-TD3 算法与 Meta-DDPG 算法结果类似,均可在较短训练幕数与较小经验池容量下充分利用元初始参数内的先验知识,平均奖励曲线在 250 幕后逐渐收敛. 而 TD3 算法在此情况下同样陷入错误的局部最优,无法正确收敛且平均奖



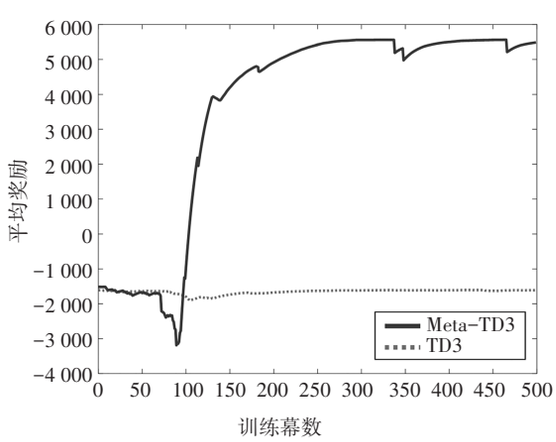
(a) Meta-DDPG 在测试环境(3)中测试结果



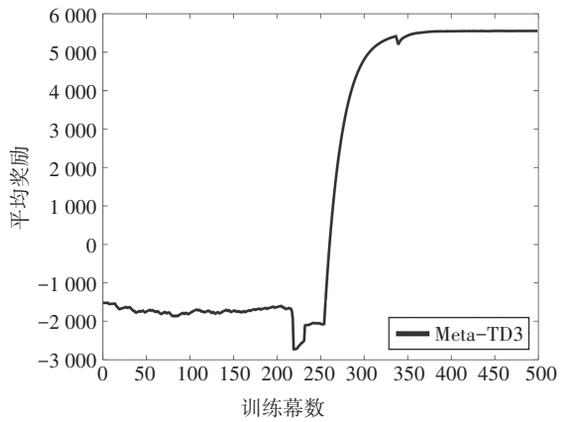
(b) 测试环境(3)中平均奖励对比曲线

图 8 Meta-DDPG 在测试环境(3)中实验结果

Fig.8 Experimental results of Meta-DDPG in the testing environment(3)



(a)测试环境(2)中平均奖励对比曲线



(b) Meta-TD3在测试环境(3)中平均奖励曲线

图9　元学习方法与基本元任务集通用性实验结果

Fig.9　Experimental results of generality of the meta learning and the basic meta-task sets

励始终小于0. 由图9(b)可知Meta-TD3算法面对新测试环境时可在300幕后逐渐达到收敛,具有较高的环境适应性. 以上表明元学习方法和基本元任务集对确定性策略强化学习算法具有较好的通用性,且元强化学习方法能够有效地解决传统深度强化学习算法中存在的收敛特性差、面对新任务泛化能力弱的问题.

6　结　论

本文对无人机自主避障与目标追踪任务进行建模,将深度强化学习算法DDPG与元学习算法MAML结合,并设计一种内外部元参数更新规则,提出元强化学习算法Meta-DDPG. 该算法能够有效地解决传统深度强化学习存在的收敛特性差、面对新任务泛化能力弱的问题. 此外,构建基本元任务集以提升工程应用时预训练的效率. 仿真结果表明,在求解无人机自主避障与目标追踪任务时,不论是对于该无人机任务训练的收敛特性,还是面对不同任务的环境适应性,Meta-DDPG算法与DDPG算法对比都有着显著的提高. 同时,使用基本元任务集进行预训练时,比传统元任务集更为高效. 且元学习方法和基本元任务集对于确定性策略强化学习算法具有较好的通用性.

参考文献

[1] 马小铭,靳伍银. 基于改进蚁群算法的多目标路径规划研究[J]. 计算技术与自动化,2020,39(4):100-105.
MA X M,JIN W Y. Mulit-objcctive path planning based on improved and colony algorithm[J]. Computing Technology and Automation,2020,39(4):100-105.(In Chinese).

[2] XU H T,HINOSTROZA M A,GUEDES SOARES C G. Modified vector field path-following control system for an underactuated autonomous surface ship modelin the presence of static obstacles[J]. Journal of Marine Science and Engineering,2021,9(6):652.

[3] ZHANG T K,LEI J Y,LIU Y W,*et al*. Trajectory optimization for UAV emergency communication with limited user equipment energy: a safe-DQN approach[J]. IEEE Transactions on Green Communications and Networking,2021,5(3):1236-1247.

[4] HUANG H J,YANG Y C,WANG H,*et al*. Deep reinforcement learning for UAV navigation through massive MIMO technique[J]. IEEE Transactions on Vehicular Technology,2020,69(1):1117-1121.

[5] WU X,CHEN H L,CHEN C G,*et al*. The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method[J]. Knowledge-Based Systems,2020,196:105201.

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

[7] YOU S X,DIAO M,GAO L P,et al. Target tracking strategy using deep deterministic policy gradient[J]. Applied Soft Computing,2020,95:106490.

[8] HU Z J,WAN K F,GAO X G,*et al*. Deep reinforcement learning approach with multiple experience pools for UAV's autonomous motion planning in complex unknown environments[J]. Sensors(Basel,Switzerland),2020,20(7):1890.

[9] LILLICRAP T P,HUNT J J,PRITZEL A,*et al*. Continuous control with deep reinforcement learning[EB/OL]. 2015: arXiv:1509.02971[cs. LG]. https://arxiv. org/abs/1509.02971.

[10] FINN C,ABBEEL P,LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[EB/OL]. 2017: arXiv:1703.03400[cs. LG]. https://arxiv. org/abs/1703.03400.

[11] WANG J X,KURTH-NELSON Z,TIRUMALA D,et al. Learning to reinforcement learn[EB/OL]. 2016: arXiv: 1611.05763[cs. LG]. https://arxiv. org/abs/1611.05763.

[12] XU J Y,YAO L,LI L,*et al*. Argumentation based reinforcement learning for meta-knowledge extraction[J]. Information Sciences,2020,506:258-272.

[13] 张耀中,许佳林,姚康佳,等. 基于DDPG算法的无人机集群追击任务[J]. 航空学报,2020,41(10):324000.
ZHANG Y Z,XU J L,YAO K J,*et al*. Pursuit missions for UAV swarms based on DDPG algorithm[J]. Acta Aeronautica et Astronautica Sinica,2020,41(10):324000.(In Chinese).

[14] 陆嘉猷,凌兴宏,刘全,等. 基于自适应调节策略熵的元强化学习算法[J]. 计算机科学,2021,48(6):168-174.
LU J Y,LING X H,LIU Q,*et al*. Meta-reinforcement learning algorithm based on automating policy entropy[J]. Computer Science,2021,48(6):168-174.(In Chinese).

[15] HU Y,CHEN M Z,SAAD W,*et al*. Distributed multi-agent meta learning for trajectory design in wireless drone networks[J]. IEEE Journal on Selected Areas in Communications, 2021, 39(10):3177-3192.

[16] BELKHALE S, LI R, KAHN G, *et al*. Model-based meta-reinforcement learning for flight with suspended payloads[J]. IEEE Robotics and Automation Letters,2021,6(2):1471-1478.

[17] FUJIMOTO S, VAN HOOF H, MEGER D. Addressing function approximation error in actor-critic methods[EB/OL]. 2018: arXiv:1802.09477[cs. AI]. https://arxiv. org/abs/1802.09477.