面向飞行器智能协同控制的分层双时延策略梯度强化学习方法

马宇[1]，安豆[2]，林熙祥[2]，赵建福[2]，张光华[2]，牛鸿敏[1]
（1. 长安大学电子与控制工程学院，710064，西安；2. 西安交通大学自动化科学与工程学院，710049，西安）

摘要：针对多飞行器智能协同控制中因规模大、环境复杂及资源受限导致的建模与协同难题，以提高决策算法效率为目标，构建了多智能体分层决策架构，提出了智能协同控制方法。首先，将飞行器作为智能体构建协同控制模型；其次，采用部分可观测马尔可夫决策过程模型解决观测信息不全问题；然后，针对博弈环境多变和学习成本问题，提出基于集中训练分布执行的分层双时延策略梯度强化学习方法，融合有模型（model-based）与无模型（model-free）机制高效利用现有博弈环境的演化模型；最后，在分层智能决策框架下，进行典型多飞行器博弈及千次多场景的仿真验证。结果表明，新方法有效解决多飞行器协同控制问题，相较于多智能体强化学习算法MAPPO和QMIX，训练时间分别减少了51.03%和79.03%，算法效率（累积回报）分别提升了37.51%和58.73%，规避机动成功率分别提高了17.63%和39.79%。

关键词：智能决策；多飞行器智能协同控制；分层决策；强化学习
中图分类号：TJ765 **文献标志码**：A
DOI：10.7652/xjtuxb202509009 **文章编号**：0253-987X(2025)09-0088-11

Hierarchical Twin-Delayed Policy Gradient Reinforcement Learning for Intelligent Cooperative Control of Aircraft

MA Yu[1], AN Dou[2], LIN Xixiang[2], ZHAO Jianfu[2], ZHANG Guanghua[2], NIU Hongmin[1]
(1. School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China;
2. School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: To address the modeling and coordination challenges in intelligent cooperative control of aircraft caused by large-scale systems, complex environments, and resource constraints, this study proposes an intelligent cooperative control method by establishing a hierarchical multi-agent decision-making architecture with the goal of improving decision-making algorithm efficiency. First, aircraft is modeled as an intelligent agent to establish a cooperative control framework. Second, a partially observable Markov decision process (POMDP) model is employed to handle incomplete observation information. Then, to tackle the issues of dynamic game environments and high learning costs, a hierarchical twin-delayed policy gradient reinforcement learning method based on centralized training with decentralized execution is proposed, which effectively combines model-based and model-free mechanisms to leverage existing game environment evolution models. Finally, under the hierarchical decision-making framework, simulations of typical multi-aircraft

收稿日期：2025-02-24。 作者简介：马宇（1988—），男，讲师。 基金项目：国家自然科学基金资助项目（62173268，62103318）；陕西省自然科学基金资助项目（2021JQ-288）。
网络出版时间：2025-04-27 网络出版地址：https://link.cnki.net/urlid/61.1069.T.20250425.1553.004

game scenarios and thousands of multi-scenario tests are conducted. The results demonstrate that the proposed method successfully resolves multi-aircraft cooperative control problem. Compared to the multi-agent reinforcement learning algorithms MAPPO and QMIX, the training time is reduced by 51.03% and 79.03%, algorithm efficiency (cumulative reward) is improved by 37.51% and 58.73%, and evasion maneuver success rate is increased by 17.63% and 39.79%, respectively.

Keywords: intelligent decision-making; multi-aircraft intelligent cooperative control; hierarchical decision; reinforcement learning

当前多飞行器协同控制体系正加速从单平台独立控制向跨域智能博弈演进，其技术突破聚焦于信息物理系统的深度耦合、智能决策架构重构以及集群协同机制创新[1]。尽管经典控制理论在特定场景中取得进展，诸如最优协同控制理论成功解决三体对抗问题[2-3]，微分对策方法实现对拦截器的主动规避[4-5]，但其应用边界受限于低维假设，存在策略稳定性不足、环境适应性弱等固有缺陷，难以应对多飞行器集群的高维非线性博弈。这驱动了研究向群体智能方向迁移，利用群体机动优势，通过多飞行器间的信息共享与任务规划构建基于多飞行器协同控制的智能规避多拦截器体系[6]。

近年来，深度强化学习(DRL)通过融合深度神经网络(DNN)的感知优势与强化学习(RL)的序列决策能力，在复杂多飞行器协同控制场景下的自主决策领域展现出显著潜力[7-9]。特别是在飞行器追逃博弈场景中，早期研究聚焦于单智能体场景，基于深度 Q 网络(DQN)[10]和异步优势行动者-评论家(A3C)[11]的深度强化学习已实现飞行器在单对单博弈中的最优机动决策。随着任务复杂度的提升，研究逐步向多智能体协同决策延伸，文献[12]提出了一种基于多智能体强化学习的多飞行器协同控制策略。研究进一步向精细化方向发展，文献[13]结合近端策略优化(PPO)算法，提出预采样机制优化拦截器轨迹预测，分析拦截器空间分布与数量对策略的影响，验证算法在飞行器动态环境中的泛化能力，但其未考虑高机动拦截器或突变路径。针对协同决策中的维度灾难问题，文献[14]提出的分层决策网络与经验分解机制有效降低了动作空间的决策复杂度，但该策略集中于个体避障，缺乏全局协同战术。在对抗控制策略研究领域，现有成果主要集中于简化场景的算法验证：基于 DQN[15]、双延迟深度确定性策略梯度(TD3)[16]以及 PPO[17]的控制策略虽在二维空间单拦截器场景取得突破，但难以适应三维空间多拦截器的复杂博弈场景。尽管文献[18]的三体对抗模型拓展了研究边界，但其复杂的奖励函数导致策略训练收敛难，制约了算法的实用性。

此外，针对高超声速类特殊飞行器，规避机动策略研究已从传统程序化控制向智能协调控制加速演进，逐步形成与 DRL 反拦截方法[19]并行的技术路线。文献[20]将传统的程序化机动策略提升为智能机动策略，提高了飞行器的自主性和适应性，但限于 TD3 算法的局限性，在复杂多变博弈场景下的泛化能力不足。文献[21]通过结合 TD3 算法和 DNN，在保证收敛速度的同时增强策略鲁棒性，然而模型复杂度和数据依赖性增加，仅适用于结构化对抗场景。文献[22]的双拦截器协同控制模型则推动 DRL 向异构对抗场景延伸。值得注意的是，超高速机动场景的特殊性催生了新型控制器的开发。文献[23]利用 D3Q 控制器的高智能性和适应性，能够在多种随机场景下智能地规避拦截器。文献[24]在此基础上改进记忆池生成方法，提高模型泛化能力。此外，Q-learning 算法[25]和 TD3 算法[26]也在机动控制策略优化领域取得显著进展，通过动态环境适应机制展现出良好的鲁棒性与泛化性能。然而，现有研究普遍受限于二维空间建模，导致三维运动学耦合效应缺失，严重制约了智能策略向三维场景的迁移能力。

针对多智能体协同控制的高维复杂性难题，分层强化学习框架[27]有效利用其在多智能体复杂环境中高效学习的优势，突显了在多层决策过程中合理分配任务和职责的重要性。鉴于多智能体近端策略优化算法(MAPPO)[28]和 Q 混合算法(QMIX)[29]在多智能体复杂环境中已展现出的良好效果，MADDPG 算法在连续动作控制与多引擎协同方面表现突出，实现飞行器自主规避多拦截器，但存在训练稳定性不足、奖励函数依赖性强等问题[30]。然而 TD3 算法的双重学习和策略延迟更新机制则有效降低了价值估计的过估计和策略的方差，适合低稳定性场景的任务[31]。因此，本文创新性地融合上述方法优势，改进了 TD3 算法，提出了分层双时延策略梯度强化学习，在更加复杂的多飞行器三维博弈

环境中进行研究，设计了集中训练与分布执行相结合的强化学习机制，增强了算法的适应性和效率，在保持多智能体协同优势的同时，突破传统算法在复杂博弈场景下的泛化瓶颈。得到的主要的贡献如下。

(1)设计集成重要性评价与信息融合的集中训练分布执行策略机制，通过局部观测选择动作，在集中训练中在线学习其他单元决策模型，结合信息融合迁移至自身策略以降低学习成本。

(2)提出结合有模型与无模型的学习机制，通过先验知识或飞行控制经验，分析多飞行器协同机动的动态变化规律，利用环境模型生成训练数据，实现小样本数据的高效利用与决策策略优化。

(3) 多飞行器协同控制中，基于分层双时延策略梯度强化学习架构，通过机动模式与动作的动态调整，在持续向目标导引的同时提升对动态环境的适应能力。

1 多飞行器协同控制系统模型

针对多飞行器协同控制问题，将飞行器抽象为强化学习智能体，在构建协同控制系统模型时，将飞行器 i 与拦截器 j 分别设定为博弈双方，其中飞行器需通过机动规避拦截器。博弈双方统一的质心运动学模型为

$$\dot{x}_l = V_l \cos\theta_l \cos\psi_l \tag{1}$$

$$\dot{y}_l = V_l \sin\theta_l \tag{2}$$

$$\dot{z}_l = -V_l \cos\theta_l \sin\psi_l \tag{3}$$

式中：l 为 i 和 j；(x_l, y_l, z_l) 为飞行器坐标；V_l、θ_l、ψ_l 分别为飞行器速度、倾角、偏角；g 为重力加速度。

动力学模型为

$$\dot{V}_l = -g\sin\theta_l \tag{4}$$

$$\dot{\theta}_l = P_y/(m_l V_l) - g\cos\theta_l/V_l \tag{5}$$

$$\dot{\psi}_l = -P_z/(m_l V_l \cos\theta_l) \tag{6}$$

式中：P_y、P_z 为法向推力、侧向推力。质量模型为

$$\dot{m}_l = -(|P_y| + |P_z|)/I_{sp} \tag{7}$$

式中：m_l 为质量；I_{sp} 为比冲。

2 分层双时延策略梯度强化学习算法

针对多飞行器协同控制环境中可以共享信息的情况下，本文设计集中训练分布执行的策略以降低学习成本，将在统一的分层决策框架中完成，同时使用有模型与无模型相结合的学习机制，高效利用现有信息学习博弈环境的演化模型，更新智能体的决策。基于飞行器大规模协同飞行的发展趋势，博弈场景中飞行器 $i(i = 1, 2, \cdots, n)$ 的数量增多，分为领机和从机两种协同实体。通过构建多智能体马尔科夫决策过程，并使用强化学习机制对博弈场景中的多个飞行器进行训练，成功机动规避多个拦截器 j，$j = 1, 2, \cdots, m$。

2.1 三维相对运动环境模型建立

通过建飞行器 i 和拦截器 j 的三维相对运动模型，描述博弈环境里双方在地面坐标系(o-xyz)上的运动关系方程，其中相对距离和相对速度的表达式如下

$$[X_{ij}, Y_{ij}, Z_{ij}]^{\mathrm{T}} = [x_i - x_j, y_i - y_j, z_i - z_j]^{\mathrm{T}} \tag{8}$$

$$[V_{ij}^x, V_{ij}^y, V_{ij}^z]^{\mathrm{T}} = [V_i^x - V_j^x, V_i^y - V_j^y, V_i^z - V_j^z]^{\mathrm{T}} \tag{9}$$

相对距离 R_{ij} 和相对距离变化率 $\dot{R}_{ij}$ 为

$$R_{ij} = \sqrt{X_{ij}^2 + Y_{ij}^2 + Z_{ij}^2} \tag{10}$$

$$R_{ij} = (X_{ij}V_{ij}^x + Y_{ij}V_{ij}^y + Z_{ij}V_{ij}^z)/R_{ij} \tag{11}$$

视线高低角 q_{ij}^1 和视线方位角 q_{ij}^2 为

$$q_{ij}^1 = \arctan(Y_{ij}/\sqrt{X_{ij}^2 + Z_{ij}^2}) \tag{12}$$

$$q_{ij}^2 = \arctan(-Z_{ij}/X_{ij}) \tag{13}$$

对应的视线高低角变化率和视线方位角变化率为

$$\dot{q}_{ij}^1 = \frac{(\sqrt{X_{ij}^2 + Z_{ij}^2})V_{ij}^y - Y_{ij}(X_{ij}V_{ij}^x + Z_{ij}V_{ij}^z)}{R_{ij}^2\sqrt{X_{ij}^2 + Z_{ij}^2}} \tag{14}$$

$$\dot{q}_{ij}^2 = \frac{V_{ij}^x Z_{ij} - X_{ij}V_{ij}^z}{X_{ij}^2 + Y_{ij}^2} \tag{15}$$

2.2 部分可观测的马尔可夫决策过程模型构建

针对多飞行器协同博弈中部分可观测性问题，本文采用部分可观测马尔可夫决策过程(POMDP)建模，通过协同控制与不完全信息推理实现提高规避机动成功率。POMDP 模型中，飞行器依据历史推断的状态概率分布选择动作，通过观测更新进行概率推理决策，优化长期收益，确保不确定环境下生成自适应策略。POMDP 模型的建立如下所示。

(1)状态空间：主要包括飞行器的局部状态观测空间 o_i 以及全局状态空间 s。

(2)局部状态观测空间：由于博弈场景中有多个需要规避的拦截器，且环境复杂多变和通信资源有限，每个飞行器的局部状态观测只需关注对自身威胁程度最高的一个拦截器，Critic 网络的全局状态输入保证了对状态-动作评价的全局性。于是 Actor 网络的输入维度将是固定值，从而提高飞行器的学习效率。

将零控规避距离作为威胁评估量化指标，为各

飞行器智能体实时匹配威胁度最高的拦截器。目标分配之后，根据与该拦截器的相对位置关系构造局部状态观测空间，即每个飞行器均有自身的局部状态观测空间 o_i，其数量与飞行器的个数呈线性关系

$$o_i = \{X_{ij_{\max}}, Y_{ij_{\max}}, Z_{ij_{\max}}, V^x_{ij_{\max}}, V^y_{ij_{\max}}, V^z_{ij_{\max}}, \dot{q}^1_{ij_{\max}}, \dot{q}^2_{ij_{\max}}, \overline{m}_i\} \tag{16}$$

式中：$j_{\max}$ 代表对飞行器 i 威胁程度最大的一个拦截器；$\overline{m}_i$ 为飞行器 i 的剩余燃料。

(3)全局状态空间：全局状态空间包含所有飞行器以及所有拦截器的信息。全局状态空间 s 的设计为

$$s = \{x_i, y_i, z_i, V^x_i, V^y_i, V^z_i, \cdots, x_j, y_j, z_j, V^x_j, V^y_j, V^z_j, \cdots, \overline{m}_i, \cdots\} \tag{17}$$

(4)动作空间：动作空间是飞行器所有可以执行的动作集合，则飞行器 i 的控制量为法向和侧向推力，动作空间设计为 $A_i \in [P_y, P_z]$。

(5)奖励函数：奖励函数包括末端奖励函数和瞬时奖励函数。末端奖励函数需准确映射规避机动效能，其数学表征与博弈终态的有效规避的飞行器数量呈正相关，表达式如下

$$R_t = n_s R_c \tag{18}$$

式中：n_s 表示规避成功的飞行器数量；R_c 为大于 0 的常数。成功规避的飞行器数量越多，飞行器获得的奖励值也就越高，引导着各飞行器在规避的过程中加强协作，使得所有的飞行器都能够规避成功。

为了提高各飞行器之间的协同性，每个飞行器的瞬时奖励函数之中也包括了与其他飞行器和拦截器的零控规避距离相关的奖励值，表达式为

$$R^i_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} \omega_i \mathrm{e}^{-k_1 t^{ij}_{go}} \lg(k_2 d^{ij} + k_3) \tag{19}$$

式中：ω_i 表示每个飞行器的奖励权重，不同类型飞行器的奖励权重不同，在奖励合成时综合地考虑交互重要性；t^{ij}_{go} 和 d^{ij} 分别表示拦截器 j 相对于飞行器 i 的剩余飞行时间和规避距离；k_1、k_2、k_3 为给定系数。另外，瞬时奖励函数也应包括自身的燃料消耗情况，表达式为

$$R^i_2 = \overline{m}_i / m_{i0} - 1 \tag{20}$$

式中：$\overline{m}_i$ 和 m_{i0} 分别为飞行器 i 的燃料剩余量和初始量。飞行器 i 的综合瞬时奖励函数为

$$R^i_s = c_1 R^i_1 + c_2 R^i_2 \tag{21}$$

式中：c_1 和 c_2 为奖励函数权重。

2.3　集中训练分布执行的分层决策架构设计

多飞行器协同控制中，观测局限与策略调整引发的环境不稳定性导致局部信息难以优化策略及收敛训练，故强化学习需解决观测局部性与博弈环境不稳定性难题。针对多飞行器复杂三维交互场景，本文引入分层决策架构，通过任务分层分配降低环境维度，在维持多智能体协同优势的同时，突破传统 TD3 在非稳态博弈环境中的泛化瓶颈。此外，设计了基于集中式训练分布式执行的训练框架，飞行器在执行动作时只考虑自身观测的局部信息，而在对动作进行评价时使用了全局状态信息且考虑了其他飞行器的动作。使用全局信息的状态-动作评价，解决单个飞行器学习中环境不稳定的问题，而局部的动作输出会提高网络学习效率，可以针对每个飞行器设计不同奖励函数以达到协同的目的。本文设计的分层双时延策略梯度强化学习框架如图 1 所示，其中 s、s' 为飞行器当前和下一个状态，a 为执行动作，r 为奖励。

(1)机动模式选择设计。机动规避的最终目标为飞行器到达目标点，主要目标为完成过程，次要目标为节约燃料。规避过程会遇到 3 种情况：一是规避前距离目标位置较远且未观测到威胁度较高的目标；二是规避中观测到威胁度较高目标时进行机动；三是观测到威胁消失时进行最终导引向目标。因此，在多飞行器协同规避拦截器的过程中，分层双时延策略梯度强化学习的两个设计要素为机动模式切换和机动动作。

上层的决策机动模式。决策基于飞行器 i 的当前状态、观测和威胁度，利用强化学习环节的结束状态奖励作为正负样本，通过神经网络进行类监督学习来优化规避的距离。优化具体过程为，网络首先输入当前状态、观测和威胁度，每个实例都会根据规避成功与否获得一个标签。初始化神经网络的权重 θ_{top}，网络将输出一个决策信号 A_{top}，表示是否切换机动模式。对于每个输入，网络进行前向传播来预测 A_{top}。使用网络的预测输出和实际的标签来计算交叉熵损失作为损失函数，定义为

$$L(\theta_{top}) = -\frac{1}{N}\sum_{k=1}^{N}[y_k \log(\hat{y}_k) + (1-y_k)\log(1-\hat{y}_k)] \tag{22}$$

式中：N 为是样本的数量；y_k 是样本的真实标签；$\hat{y}_k$ 是神经网络预测的概率输出。之后网络使用损失函数来进行反向传播，计算每个权重参数的梯度。更新网络权重以最小化损失函数，使用梯度下降来进行更新，如下

$$\theta_{top} = \theta_{top} - \alpha_{top} \nabla_{\theta_{top}} L(\theta_{top}) \tag{23}$$

式中：α_{top} 为上层学习率；$\nabla_{\theta_{top}} L(\theta_{top})$ 是损失函数相对于权重的梯度。

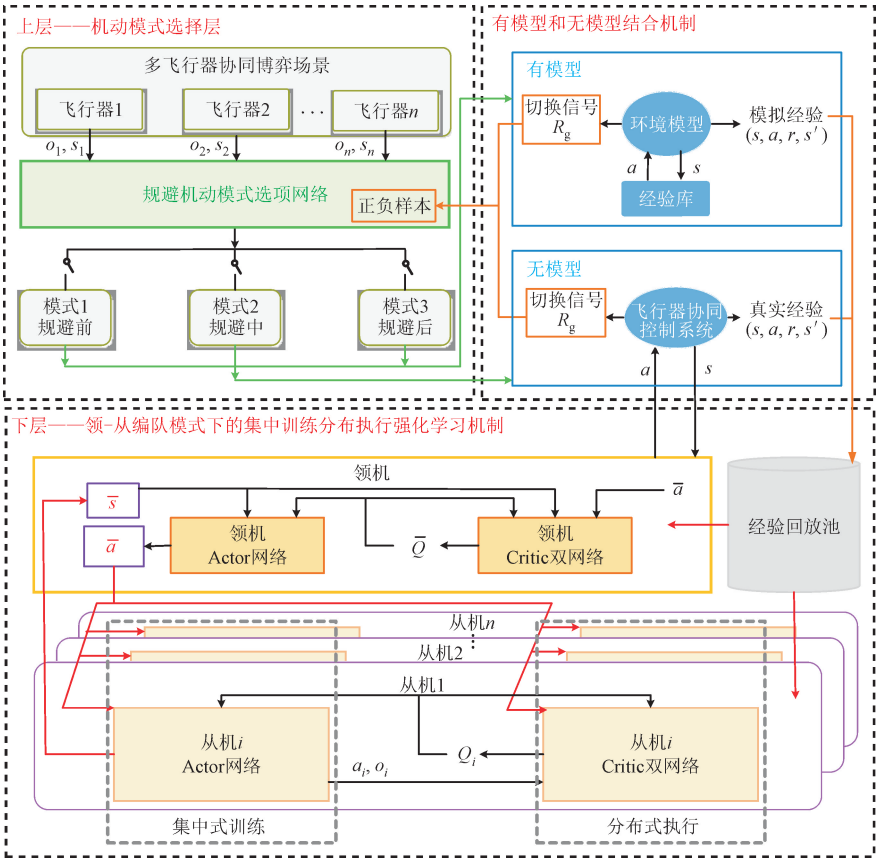


图 1 分层双时延策略梯度强化学习框架

Fig. 1 Schematic of hierarchical twin delayed policy gradient reinforcement learning

迭代以上过程，每一次迭代都旨在减少预测输出和实际标签之间的差异，从而使得神经网络可以更准确地预测机动模式的切换信号。决策结果决定了飞行器是采用经验库中的策略进行稳定飞行，还是切换到强化学习策略以应对更高的威胁度。这一切换本质上是一个单步动作 $A_{\text{top}}=[R_{\text{p}},R_{\text{g}}]$，$R_{\text{p}}$ 表示规避距离，$R_{\text{g}}=\begin{cases}0, & R_{\text{p}}>R_{\text{h}}\\1, & R_{\text{p}}\leqslant R_{\text{h}}\end{cases}$ 表示模式的切换信号，如果当前位置到达了所给出的切换距离 R_{h}，则 R_{g} 为 1，否则为 0，仍然保持当前模式。

上层的决策结果一旦执行，下层的训练方式将会确定。如果飞行器 i 当前处于规避前或规避后，则使用有模型训练，利用已有的专家经验信息对博弈态势的演变规律进行建模，利用专家经验知识中包含的状态 - 动作对预训练网络，使其能够选择在给定状态下获得最大预期回报的动作。使用环境模型 D，预测下一个状态和奖励，从而生成模拟经验〈s，a，r，s'〉用于对环境模型 D 和专家策略网络的更新。基于专家经验与生成的模拟数据训练个体的决策模型，能够做到对少量数据的高效利用。如果飞行器 i 当前状态处于规避中，则开启无模型强化学习训练方法增强策略多样性，以应对动态变化大、威胁度高的场景。一旦飞行器进入到具体的规避动作决策阶段，飞行器将会开启下层的强化学习机制。本文设计了集中训练分布执行机制，在这一层次，飞行器需要根据实时的观测信息来执行具体的机动动作，以有效地规避拦截并保持向目标飞行。动作的选择和执行是通过强化学习训练的，利用 Actor-Critic 网络对规避过程中每对状态-动作-奖励信息进行优化，从而提高规避成功率。

(2)有模型和无模型相结合的强化学习训练机制。基于上述分层双时延策略梯度强化学习框架得到的飞行器控制系统有较强适应性。有模型方法更多在规避前后阶段采用，以最大程度利用专家经验来提高飞行器的稳定性。而无模型方法则在规避过程中增强对突发威胁的应变能力。结合两者的训练机制，不仅能够从经验中学习，而且能够利用环境模型来计划和预测未来状态。

在规避前后两个阶段，模型 D 利用专家数据进行监督学习，以预测从当前状态到下一个状态的转换及其可能的奖励，在有模型部分中用来预测飞行状态和环境反馈。在此阶段，使用训练好的模型 D 来预测未来的飞行路径和可能遭遇的威胁，辅助飞行控制策略，同时所产生的模拟经验 $\langle s,a,r,s'\rangle$ 会一并存入经验池中辅助无模型强化学习的决策。类似地，模型 D 的更新公式如下

$$\theta_{\mathrm{D}}=\theta_{\mathrm{D}}-\alpha_{\mathrm{D}}\nabla_{\theta_{\mathrm{D}}}L_{\mathrm{D}}(\theta_{\mathrm{D}}) \tag{24}$$

式中：α_{D} 为模型 D 学习率；$\nabla_{\theta_{\mathrm{D}}}L_{\mathrm{D}}(\theta_{\mathrm{D}})$ 是损失函数相对于权重的梯度。

在机动规避阶段，基于无模型的强化学习方法，智能体通过与环境的交互获得真实经验 $\langle s,a,r,s'\rangle$。通过从包含真实经验和模拟经验的经验池中取样，更新 Actor-Critic 网络，不过于依赖模拟环境模型 D 的准确性。结合有模型方法提供的未来飞行和威胁信息与无模型方法学习到的直接经验，更新飞行控制策略。最后，在实际或高度仿真的环境中执行更新后的飞行器控制策略，观察其在完成规避和抵达目标中的性能。

(3)领-从编队模式下的集中训练分布执行强化学习机制。在多飞行器协同博弈场景中，领导者(领机)负责制定高层策略，旨在最大化整个系统的性能，而跟随者(从机)则在领导者发布的动作指导下行动，学习如何最好地响应上层的策略以优化自己的策略。在领导者-跟随者模型中，信息的流动是从领导者到跟随者的，而跟随者则基于领导者的决策和局部信息进行行动。如图 1 所示，领机通过从机获得全局信息 $\bar{s}$，根据全局信息学习一个最优策略 $\bar{\pi}^{*}(\bar{a}\mid\bar{s})$，以最大化累积的合作回报。领机值函数 $\bar{Q}$ 表示在当前状态下执行不同操作可以获得的预期累积回报。对于探索，利用 ε 贪婪法来选择领机的行动。也就是说，以 $1-\varepsilon$ 的概率随机选择一个动作 $\bar{a}$，或以 ε 的概率采取与当前状态下的最大值函数相对应的动作。完成操作选择后，领机将广播一条消息，将动作策略发布给下属的从机。从机根据所选择的交互对象的局部观测信息 o_i 来扩展自身观测，最后构建出由关键特征向量组成的全局信息 $\bar{s}$ 传递给领机。在从机集中训练和分布执行阶段，从机将从领机获取的高级动作信息 $\bar{a}$ 广播给所有从机进行策略指导，指示下属的所有从机。从机随后根据收到的消息选择自身的行动 a_i，获得从机 i 值函数 Q_i。

强化学习中的智能体使用 Actor-Critic 架构，特别是采用了双 Critic 网络来稳定训练过程和减少方差。在训练时，每个智能体都需要和其它智能体进行频繁的交互。在执行阶段，智能体间独立做出决策可以极大地减少实时操作中的通信需求，尤其是在态势紧张的博弈环境中，快速响应是非常重要的。因此，本文采用分布式执行机制，每个智能体只基于观测，通过 Actor 网络来输出动作。此外，双 Critic 结构能避免过拟合等问题，在训练阶段，每个智能体都带有两组 Actor-Critic 网络，以及它们相应的目标网络。经验回放缓存用于存放一系列形如 $\langle s,a,r,s'\rangle$ 的 4 元组，4 元组部分来自真实经验，部分来自交互产生的模拟经验。在神经网络学习的时候从中随机抽取一定数量的 4 元组用于学习，以打断它们之间的关联，从而提高神经网络的训练稳定性。基于集中训练分布执行框架的描述，智能体 u 包含一个 Actor 网络，一个 Critic 网络，Critic 网络实际是双网络，3 个网络权重简记为 μ_u、$Q_{u,1}$ 和 $Q_{u,2}$，对应的目标网络权重简记为 μ'_u、$Q'_{u,1}$ 和 $Q'_{u,2}$。

Critic 网络权重 $Q_{u,k}(k=1,2)$ 通过最小化损失函数来更新，损失函数的表达式如下

$$L_{u,k}=E\left[\left(y-Q_{u,k}(\boldsymbol{s},\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_u,\cdots,\boldsymbol{a}_U)\right)^2\right] \tag{25}$$

式中：$y=r_u+\gamma\min_{k=1,2}Q'_{u,k}(\boldsymbol{s}',\tilde{\boldsymbol{a}}_1,\tilde{\boldsymbol{a}}_2,\cdots,\tilde{\boldsymbol{a}}_u,\cdots,\tilde{\boldsymbol{a}}_U)$，$\tilde{\boldsymbol{a}}_u$ 表达式为

$$\tilde{\boldsymbol{a}}_u=\mathrm{clip}(\mu'_u(s'_u)+\mathrm{clip}(\tilde{\varepsilon},-h,h),\boldsymbol{a}_{\mathrm{low}},\boldsymbol{a}_{\mathrm{high}}) \tag{26}$$

其中，clip 函数指将输入值限制在指定的上、下界之间，$\boldsymbol{a}_{\mathrm{low}}$ 和 $\boldsymbol{a}_{\mathrm{high}}$ 分别表示动作的下界和上界，$-h$ 和 h 分别表示噪声的下界和上界，$\tilde{\varepsilon}\sim N(0,\tilde{\sigma}^2)$，$\gamma$ 为折扣因子。

累积奖励的期望值为 $J(\theta_u)=E\left[\sum_{t=0}^{\infty}\gamma^t r_u^t\right]$，其中 r_u^t 是 t 步收到的奖励。为了最大化 $J(\theta_u)$，Actor 网络权重 μ_u 通过使用下式的策略梯度来更新

$$\nabla_{\theta_u}J(\theta_u)=E\left[\nabla_{\theta_u}(\boldsymbol{s}_u)\nabla_{a_u}Q_{u,1}(\boldsymbol{s},\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_u,\cdots,\boldsymbol{a}_U\mid_{a_u=\mu_u(s_u)})\right] \tag{27}$$

定义 G 为从经验池 B 中随机采样到的一个小批量样本。在训练之前，B 需要被预热，即算法先产生一系列 $\langle s,a,r,s'\rangle$ 的 4 元组来填充 B，直到 B 里面的 4 元组的数量达到预先设定的值，使得在开始训练时从 B 中随机抽取到的 4 元组有更低的关联性，从而使训练更稳定。因此，本文算法的详细流程见算法 1，其中 d 是 Actor 网络的更新频率，动作噪声 $\varepsilon\sim N(0,\bar{\sigma}^2)$。算法流程如图 2 所示。

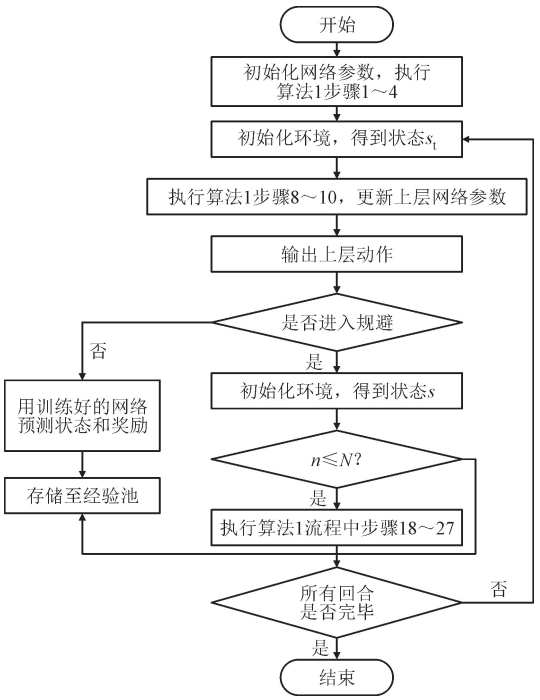


图 2 算法 1 的流程图

Fig. 2 Flowchart of algorithm 1

算法 1

1. 初始化上层网络权重 θ_{top}
2. 初始化下层模型 D 权重 θ_{D}，领机网络权重 $\bar{\mu}_{\theta}$、$\bar{Q}_{\varphi,1}$、$\bar{Q}_{\varphi,2}$，从机 Actor 网络权重 μ_u，两个 Critic 网络权重 $Q_{u,1}$ 和 $Q_{u,2}$（$u=1,2,\cdots,U$）
3. 初始化所有目标网络 μ'_u，$Q'_{u,1}$，$Q'_{u,2}$
4. 初始化经验池 B
5. 对于每个回合：
6. 　初始化环境，得到当前飞行器状态 s_t
7. 　进入上层网络，循环：
8. 　　对于每个输入的状态 s_t、观测 o_t 和威胁度，进行前向传播来预测机动模式切换信号 A_{top}
9. 　　计算损失函数 $L(\theta_{\text{top}})$
10. 　　使用损失函数反向传播，更新网络权重 θ_{top}
11. 　跳出循环，输出上层动作 A_{top}
12. 　If 进入规避前 or 规避后：
13. 　　用训练好的模型 D 预测未来状态 s_{t+1} 和奖励 r_t
14. 　　将 (s_t,a_t,r_t,s_{t+1}) 存储到经验池 B
15. 　If 进入规避：
16. 　　初始化环境，并获取初始状态 s
17. 　　for $n=1:N$ 执行：
18. 　　　每个智能体 u 选择动作 $\boldsymbol{a}_u=\mu_u(\boldsymbol{s}_u)+\varepsilon$
19. 　　　执行动作 $\boldsymbol{a}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_U)$ 然后获取到奖励 $\boldsymbol{r}$ 和下一个状态 $\boldsymbol{s}'$
20. 　　　把 $(\boldsymbol{s},\boldsymbol{a},\boldsymbol{r},\boldsymbol{s}')$ 储存到经验池 B 中
21. 　　　$\boldsymbol{s}\leftarrow\boldsymbol{s}'$
22. 　　　for 智能体 $u=1:U$ 执行：
23. 　　　　随机从 B 中采样包含 G 个样本的小批量
24. 　　　　最小化损失函数更新 Critic 网络 $Q_{u,1}$，$Q_{u,2}$
25. 　　　　If $n \bmod d=0$ 执行：
26. 　　　　　通过策略梯度来更新 Actor 网络 μ_u
27. 　　　　　更新目标网络：

$$\mu'_u\leftarrow\tau\mu_u+(1-\tau)\mu'_u$$

$$Q'_{u,k}\leftarrow\tau Q_{u,k}+(1-\tau)Q'_{u,k},k=1,2$$

28. 直到所有回合完毕，结束训练。

3 实验与结果分析

3.1 初始参数设置

实验软件环境为 Windows 11，训练框架基于 Python 及 Pytorch；硬件环境为 Intel core i7 + GeForce GTX 3060+32GB。实验场景包括 5 个高速飞行器与 10 个拦截器的博弈场景，使用本文算法来训练飞行器。场景的参数均在地面坐标系下给出，飞行器和拦截器的总质量分别为 350 和 75 kg，燃料质量分别为 120 和 25 kg，比冲分别为 2 500 和 3 200 N·s/kg。初始态势设置如表 1 所示，拦截器采用比例导引进行拦截，成功拦截的半径设为 10 m。

表 1 博弈场景的初始态势设置

Table 1 Initial situation setting of the game scenario

方案	位置/km	速度/(m·s^{-1})	速度倾角/(°)	速度偏角/(°)
领机 1	[100,0,100]	3 100	[−35, −45]	[270,280]
从机 1	[105,0,100]	3 100	[−35, −45]	[270,280]
从机 2	[95,0,100]	3 100	[−35, −45]	[270,280]
从机 3	[100,5,100]	3 100	[−35, −45]	[270,280]
从机 4	[100, −5,100]	3 100	[−35, −45]	[270,280]
拦截器 1	[32,50,0]	3 100	[40,50]	[160,170]
拦截器 2	[32,50,0]	3 100	[40,50]	[160,170]
拦截器 3	[32,50,0]	3 100	[40,50]	[160,170]
拦截器 4	[32,50,0]	3 100	[40,50]	[160,170]
拦截器 5	[32,50,0]	3 100	[40,50]	[160,170]
拦截器 6	[32, −50,0]	2 900	[40,50]	[20,30]
拦截器 7	[32, −50,0]	2 900	[40,50]	[20,30]
拦截器 8	[32, −50,0]	2 900	[40,50]	[20,30]
拦截器 9	[32, −50,0]	2 900	[40,50]	[20,30]
拦截器 10	[32, −50,0]	2 900	[40,50]	[20,30]

算法中各飞行器的 Actor 网络和 Critic 网络均采用双隐层前馈神经网络。神经网络以及算法的参数如表 2 所示。

表 2　神经网络及算法参数

Table 2　Neural network and algorithm parameters

参数	数值和函数
Actor 网络学习率 α_μ	1×10^{-5}
Critic 网络学习率 α_Q	2×10^{-5}
折扣因子 γ	0.98
软更新率 τ	0.005
经验池容量 N	2×10^{6}
批处理样本量 N_b	128
决策步长 h	0.1
Actor 网络的输入	o_i
Critic 网络第 1 层输入	s
Actor 和 Critic 第 1 隐层节点数	128
Actor 和 Critic 第 2 隐层节点数	128
Actor 激活函数	tanh
Critic 激活函数	Leaky-ReLU

用本文算法对飞行器进行 5 000 回合的训练，得到飞行器训练集每回合的奖励值和测试集，每 100 回合的奖励均值如图 3 和图 4 所示。

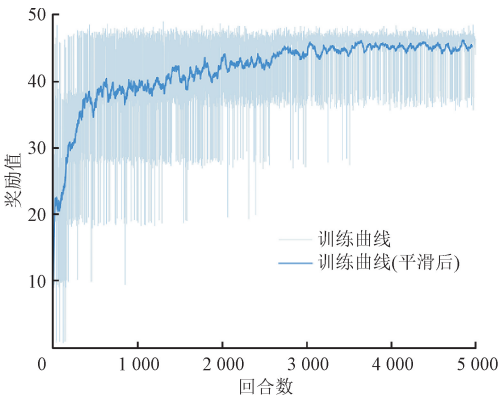


图 3　训练集每回合奖励值

Fig. 3　Reward values per turn for the training set

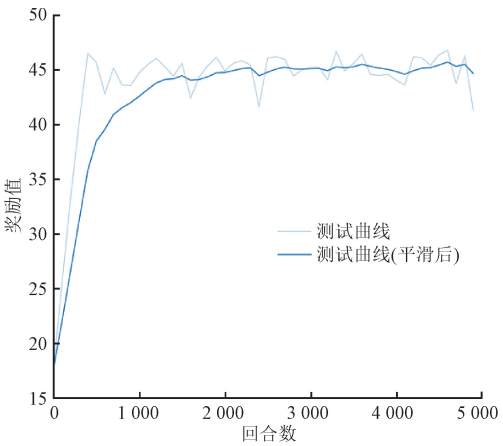


图 4　测试集每 100 回合奖励均值

Fig. 4　Mean reward per 100 rounds for the test set

从图 3、图 4 可见，0～200 回合左右，飞行器获得的奖励值迅速增加，逐渐学到成功规避的策略。200～3 200 回合，奖励值缓慢增加，且首次达到稳定值附近。约 3 600 回合时，整体趋于稳定，奖励值稳定在 45 左右，算法达到收敛。将训练好的 Actor 网络作为 5 个飞行器的策略进行测试，结果如图 5～图 7 所示。

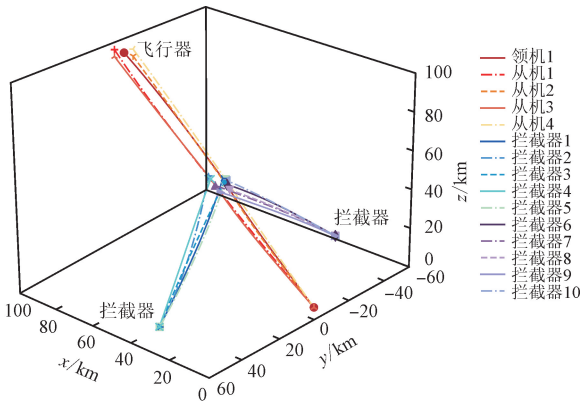
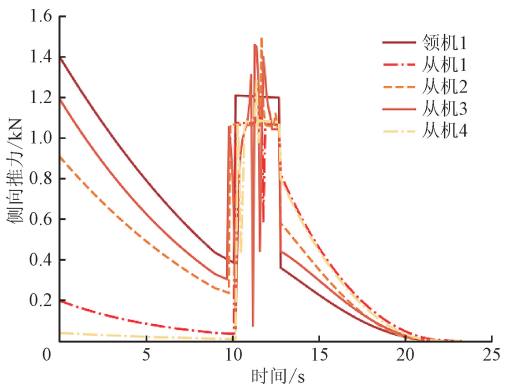
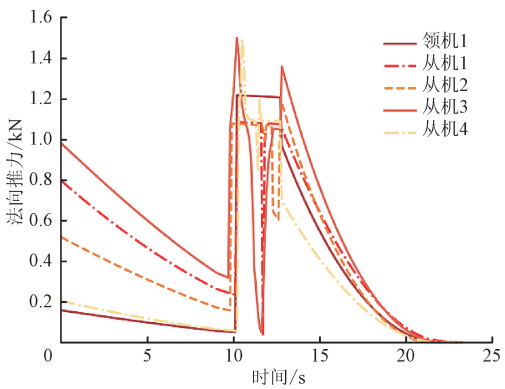


图 5　多飞行器机动曲线

Fig. 5　Multi-aircraft maneuvering curves



(a)飞行器侧向推力曲线



(b)飞行器法向推力曲线

图 6　飞行器推力曲线

Fig. 6　Aircraft thrust curve

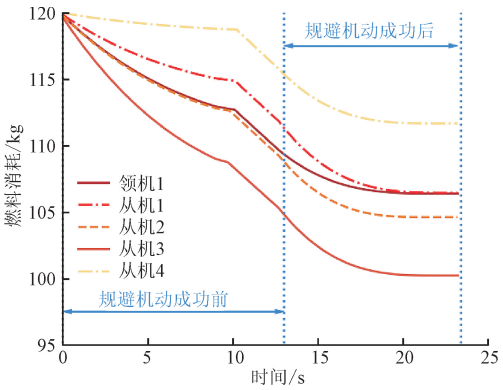


图 7 飞行器燃料消耗曲线

Fig. 7 Aircraft fuel consumption curve

领机与所有拦截器的最小规避距离为 48.64 m，从机的最小规避距离分别为 15.33、25.88、17.62、36.97 m，均大于成功拦截半径，规避成功。

由图 5 的机动曲线可以看出，飞行器基本保持初始队形，在检测到拦截器后，飞行器编队进入规避机动状态，且均完成规避任务。图 6 是飞行器的推力曲线，可以看到在 9.6 s 之前，所有飞行器的推力都在减小，在相距较远时节省燃料。在9.6～10.3 s 的过程中，领机观测到了拦截器。在 11 s 之后，所有的飞行器和拦截器的距离均减小到一定范围，此时飞行器根据自身观测到的威胁程度最大的拦截器信息，做出相应的规避动作。图 7 是飞行器在规避过程中的燃料消耗曲线，可以看到从机 4 的燃料消耗最少，结合推力曲线，其在所有飞行器后才开始有较大的推力输出，再观察规避机动曲线，发现从机 4 一直位于边缘的位置。出现上述实验结果的原因是，因为其他拦截器对从机 4 的威胁度较低，从而无需做出复杂的机动即可完成规避，规避成功时消耗 4.5 kg 左右的燃料，总燃料消耗约 8 kg。从机 3 的燃料消耗最多，说明其面临的规避任务重，需要较大的机动动作，规避成功时消耗 15.3 kg 左右燃料，总燃料消耗约 19.7 kg。观察其他的飞行器，燃料消耗总量均在 13～16 kg 之间，仍有足够燃料完成后续任务。以上的结果表明，本文算法对多飞行器进行训练，能够使所有飞行器完成规避任务，并保有后续飞行能力，验证了基于多智能体强化学习解决多飞行器协同控制问题的有效性。

3.2 规避机动成功率

规避机动成功率是评估算法性能的一个重要指标。针对 5v10(5 个飞行器，10 个拦截器)协同仿真环境下训练获得的强化学习智能体模型，在 1 000 次仿真中对规避的成功率进行评估，本文算法表现优异，成功率高达 91.2%，具备一定的泛化能力和适应性，在应对不同复杂度和规模的多飞行器协同问题上具有鲁棒性和稳定性。

3.3 算法对比

基于本文在多飞行器协同仿真环境下，对强化学习算法效率(累积回报)和训练时间的要求，选取了目前应用较为广泛的 MAPPO[28] 和 QMIX[29] 算法进行对比仿真实验。在保持超参数一致的情况下，选取 5v10 的经典场景进行仿真实验。仿真结果如图 8 和表 3 所示。

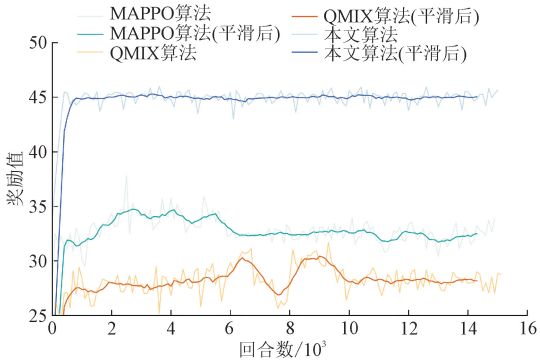


图 8 本文算法与其他多智能体强化学习算法评估过程累计奖励的比较

Fig. 8 Comparison of the cumulative reward of the evaluation process between the proposed algorithm and other multi-agent reinforcement learning algorithms

表 3 本文算法与其他多智能体强化学习算法效率的比较

Table 3 Efficiency comparison between the proposed algorithm and other multi-agent reinforcement learning algorithms

算法	指标				
	训练回报	评估回报	训练时间/s	收敛轮数	规避成功率/%
QMIX	42.281 1	28.384 3	117 043	10 000	65.26
MAPPO	34.602 1	32.762 1	50 129	6 000	77.56
本文算法	45.611 8	45.053 7	24 549	4 000	91.23
优化比例/%	↑7.88	↑37.51	↓51.03	↓33.33	↑17.63

注：表中↑、↓表示提高和下降。

在不同算法测试过程的对照中，各算法均不添加策略、动作噪声，此时本文算法的测试累计回报最高，在 4 000 步后达到收敛，且累计奖励波动较小。MAPPO 算法测试过程累计回报较 QMIX 算法更优，但仍然存在较大波动和难收敛现象。综合分析

表 3，本文算法在不同指标下与 QMIX、MAPPO 算法相比均展现出显著的性能优势。

在算法效率（累积回报）方面，训练回报为训练过程中添加了策略噪声的平均累计回报，而评估回报为不添加策略噪声的平均累计回报，更能反应出算法真实水平。本文算法在训练回报上相较于 QMIX 和 MAPPO 算法分别提升了 7.88%和31.82%，而在评估回报上分别达到了 58.73% 和 37.51% 的显著提升。

在训练时长方面，本文算法的训练速度有明显优势。与 QMIX 和 MAPPO 算法相比，本文算法在训练时间上分别缩短了 79.03%和 51.03%，在收敛轮数上也分别有 60.00%和 33.33%的减少。需要说明的是，在训练过程中，强化学习环境的迭代计算消耗了大部分算力，因此训练时长和算法复杂度弱相关，和算法收敛速度强相关，因此采用收敛速度较快的算法是减少训练时长的关键因素。

在规避机动成功率方面，本文算法相较于 QMIX 和 MAPPO 算法分别提高了 39.79%和 17.63%。

4　结　论

本文提出的智能协同控制方法解决了多飞行器智能协同控制中的博弈动态适应难题。通过构建集中式训练与分布式执行的协同策略学习架构，解决了传统方法在复杂场景下因局部观测与全局协同矛盾导致的策略稳定性不足的问题。同时，设计基于有模型与无模型相结合的学习机制，利用先验信息或专家经验，对博弈态势的演变规律进行建模，进而依靠环境模型生成训练数据。在统一分层决策模型框架中的仿真实验表明，本文算法在动态博弈场景中展现出卓越的收敛性能与环境适应性，在处理复杂的多飞行器智能协同控制领域中具有广泛应用前景。

参考文献：

[1] 郑卓，路坤锋，王昭磊，等. 飞行器集群协同控制技术分析与展望 [J]. 宇航学报，2023，44(4)：538-545.
ZHENG Zhuo，LU Kunfeng，WANG Zhaolei，et al. Analysis and prospect of vehicle swarm cooperative control technology [J]. Journal of Astronautics，2023，44(4)：538-545.

[2] 方峰，蔡远利. 三体对抗中的自适应协同突防策略 [J]. 西安交通大学学报，2017，51(4)：72-78.
FANG Feng，CAI Yuanli. An adaptive collaborative guidance strategy in three-body engagement [J]. Journal of Xi'an Jiaotong University，2017，51(4)：72-78.

[3] FANG Feng，CAI Yuanli，JABBARI F. 3D optimal defensive guidance strategy with safe distance [J]. Transactions of the Institute of Measurement and Control，2019，41(15)：4285-4300.

[4] 鲜勇，田海鹏，王剑，等. 基于微分对策的导弹智能机动突防研究 [J]. 飞行力学，2014，32(1)：70-73.
XIAN Yong，TIAN Haipeng，WANG Jian，et al. Research on intelligent maneuver penetration of missile based on differential game theory [J]. Flight Dynamics，2014，32(1)：70-73.

[5] BARDHAN R，GHOSE D. Nonlinear differential games-based impact-angle-constrained guidance law [J]. Journal of Guidance，Control，and Dynamics，2015，38(3)：384-402.

[6] 任章，郭栋，董希旺，等. 飞行器集群协同制导控制方法及应用研究 [J]. 导航定位与授时，2019，6(5)：1-9.
REN Zhang，GUO Dong，DONG Xiwang，et al. Research on the cooperative guidance and control method and application for aerial vehicle swarm systems [J]. Navigation Positioning and Timing，2019，6(5)：1-9.

[7] ZOU Xiaofei，YANG Ruopeng，YIN Changsheng，et al. Deploying tactical communication node vehicles with AlphaZero algorithm [J]. IET Communications，2020，14(9)：1392-1396.

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

[9] 谭浪，巩庆海，王会霞. 基于深度强化学习的追逃博弈算法 [J]. 航天控制，2018，36(6)：3-8.
TAN Lang，GONG Qinghai，WANG Huixia. Pursuit-evasion game algorithm based on deep reinforcement learning [J]. Aerospace Control，2018，36(6)：3-8.

[10] YANG Qiming，ZHANG Jiandong，SHI Guoqing，et al. Maneuver decision of UAV in short-range air combat based on deep reinforcement learning [J]. IEEE Access，2020，8：363-378.

[11] FAN Zihao，XU Yang，KANG Yuhang，et al. Air combat maneuver decision method based on A3C deep reinforcement learning [J]. Machines，2022，10(11)：1033.

[12] ZHANG Jiandong，YANG Qiming，SHI Guoqing，et al. UAV cooperative air combat maneuver decision based on multi-agent reinforcement learning [J]. Journal of Systems Engineering and Electronics，2021，32(6)：1421-1438.

[13] ZHUANG Xing，LI Dongguang，WANG Yue，et al.

Optimization of high-speed fixed-wing UAV penetration strategy based on deep reinforcement learning [J]. Aerospace Science and Technology, 2024, 148: 109089.

[14] WANG Huan, WANG Jintao. Enhancing multi-UAV air combat decision making via hierarchical reinforcement learning [J]. Scientific Reports, 2024, 14(1): 4458.

[15] WU Mingyu, HE Xianjun, QIU Zhiming, et al. Guidance law of interceptors against a high-speed maneuvering target based on deep Q-Network [J]. Transactions of the Institute of Measurement and Control, 2022, 44(7): 1373-1387.

[16] 裴培, 何绍溟, 王江, 等. 一种深度强化学习制导控制一体化算法 [J]. 宇航学报, 2021, 42(10): 1293-1304.
PEI Pei, HE Shaoming, WANG Jiang, et al. Integrated guidance and control for missile using deep reinforcement learning [J]. Journal of Astronautics, 2021, 42(10): 1293-1304.

[17] CHEN Wenxue, GAO Changsheng, JING Wuxing. Proximal policy optimization guidance algorithm for intercepting near-space maneuvering targets [J]. Aerospace Science and Technology, 2023, 132: 108031.

[18] GONG Xiaopeng, CHEN Wanchun, CHEN Zhongyuan. Intelligent game strategies in target-missile-defender engagement using curriculum-based deep reinforcement learning [J]. Aerospace, 2023, 10(2): 133.

[19] JIANG Liang, NAN Ying, ZHANG Yu, et al. Anti-interception guidance for hypersonic glide vehicle: a deep reinforcement learning approach [J]. Aerospace, 2022, 9(8):424.

[20] GAO Mengjing, YAN Tian, LI Quancheng, et al. Intelligent pursuit-evasion game based on deep reinforcement learning for hypersonic vehicles [J]. Aerospace, 2023, 10(1): 86.

[21] GUO Yunhe, JIANG Zijian, HUANG Hanqiao, et al. Intelligent maneuver strategy for a hypersonic pursuit-evasion game based on deep reinforcement learning [J]. Aerospace, 2023, 10(9): 783.

[22] YAN Tian, JIANG Zijian, LI Tong, et al. Intelligent maneuver strategy for hypersonic vehicles in three-player pursuit-evasion games via deep reinforcement learning [J]. Frontiers in Neuroscience, 2024, 18: 1362303.

[23] JIANG Liang, NAN Ying, LI Zhihan. Realizing midcourse penetration with deep reinforcement learning [J]. IEEE Access, 2021, 9: 89812-89822.

[24] 南英, 蒋亮. 基于深度强化学习的弹道导弹中段突防控制 [J]. 指挥信息系统与技术, 2020, 11(4): 1-9.
NAN Ying, JIANG Liang. Midcourse penetration and control of ballistic missile based on deep reinforcement learning [J]. Command Information System and Technology, 2020, 11(4): 1-9.

[25] WANG Yaokun, ZHAO Kun, GUIRAO J L G, et al. Online intelligent maneuvering penetration methods of missile with respect to unknown intercepting strategies based on reinforcement learning [J]. Electronic Research Archive, 2022, 30(12): 4366-4381.

[26] QIU Xiaoqi, GAO Changsheng, JING Wuxing. Maneuvering penetration strategies of ballistic missiles based on deep reinforcement learning [J]. Proceedings of the Institution of Mechanical Engineers: Part G Journal of Aerospace Engineering, 2022, 236(16): 3494-3504.

[27] YAN Mengda, YANG Rennong, ZHANG Ying, et al. A hierarchical reinforcement learning method for missile evasion and guidance [J]. Scientific Reports, 2022, 12(1): 18888.

[28] YU Chao, VELU A, VINITSKY E, et al. The surprising effectiveness of PPO in cooperative multi-agent games [C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2022: 24611-24624.

[29] RASHID T, SAMVELYAN M, DE WITT C S, et al. Monotonic value function factorisation for deep multi-agent reinforcement learning [J]. The Journal of Machine Learning Research, 2020, 21(1): 7234-7284.

[30] ZHAO Yu, ZHOU Ding, BAI Chengchao, et al. Reinforcement learning based spacecraft autonomous evasive maneuvers method against multi-interceptors [C]// 2020 3rd International Conference on Unmanned Systems (ICUS). Piscataway, NJ, USA: IEEE, 2020: 1108-1113.

[31] 王建波, 孙冉, 刘忠凯, 等. 面向储能辅助火电机组一次调频的深度强化学习控制策略 [J]. 西安交通大学学报, 2024, 58(6): 186-192.
WANG Jianbo, SUN Ran, LIU Zhongkai, et al. Deep reinforcement learning control strategy for primary frequency regulation of energy storage assisted thermal power units [J]. Journal of Xi'an Jiaotong University, 2024, 58(6): 186-192.

(编辑 杜秀杰)