控制与决策
Control and Decision

基于模糊强化学习和模型预测控制的追逃博弈

胡鹏林, 潘泉, 赵春晖

引用本文:
胡鹏林, 潘泉, 赵春晖. 基于模糊强化学习和模型预测控制的追逃博弈[J]. 控制与决策, 2025, 40(6): 1855-1865.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2024.1039

基于模糊强化学习和模型预测控制的追逃博弈

胡鹏林†，潘 泉，赵春晖

(西北工业大学 自动化学院，西安 710129)

摘 要: 针对三维空间中智能体追逃博弈策略制定与鲁棒控制问题, 提出一种基于模糊强化学习与模型预测控制 (MPC) 的分层追逃博弈框架. 所提出框架结合三维空间的阿氏圆和模糊行动者-评论家学习 (FACL) 算法获得智能体的运动信息, 并将其用作 MPC 算法的参考输入来设计四旋翼无人机的控制器. 通过对四旋翼欠驱动系统模型进行解耦, 设计考虑误差系统积分项的高度、平移和姿态控制器. 通过 FACL 算法提供的参考信息, 有效提高了 MPC 算法的控制效率. 仿真和实验结果表明, 所设计的分层框架可以很好地解决三维空间追逃博弈问题.

关键词: 三维追逃博弈；阿氏圆；模糊强化学习；模型预测控制

中图分类号: TP13 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2024.1039

引用格式: 胡鹏林, 潘泉, 赵春晖. 基于模糊强化学习和模型预测控制的追逃博弈 [J]. 控制与决策, 2025, 40(6): 1855-1865.

Pursuit-evasion game based on fuzzy reinforcement learning and model predictive control

HU Peng-lin†，PAN Quan，ZHAO Chun-hui

(School of Automation，Northwestern Polytechnical University，Xi'an 710129，China)

Abstract: Addressing the strategy formulation and robust control issues in the pursuit-evasion game of agents in 3D space, this paper proposes a hierarchical pursuit-evasion game framework based on fuzzy reinforcement learning and model predictive control(MPC). The proposed framework integrates the Apollonius circle in 3D space with the fuzzy actor-critic learning(FACL) algorithm to obtain the agents' motion information, which is then used as the reference input for the MPC algorithm to design the controller for quadrotor unmanned aerial vehicles. By decoupling the underactuated system model of the quadrotor, altitude, translation, and attitude controllers that consider the integral term of the error system are designed. The reference information provided by the FACL algorithm effectively enhances the control efficiency of the MPC algorithm. Simulation and experimental results demonstrate that the designed hierarchical framework can effectively solve the pursuit-evasion game problem in 3D space.

Keywords: 3D pursuit-evasion game；Apollonius circle；fuzzy reinforcement learning；model predictive control

0 引 言

追逃博弈 (pursuit-evasion game, PEG) 涉及两个对立的群体: 追击者和逃避者. PEG 主要关注追击者如何配合抓捕逃避者, 以及逃避者应采取何种策略避免被追击者抓捕或延长被抓捕的时间[1]. 近年来, PEG 广泛应用于军事和民用领域, 如导弹攻防[2]、空战[3]、搜救[4]、运输管理[5] 等场景. 学者们提出了各种形式的 PEG, 如单对单[6]、单对多[7]、多对单[8]、多对多[9] 等类型. 同时, 提出了各种算法, 如几何方法[10-11]、微分对策方法[12]、经典控制方法[13] 和强化学习 (reinforcement learning, RL) 等方法[14-15].

与其他传统控制算法相比, 模型预测控制 (model predictive control, MPC) 由于其灵活的优化能力、约束处理能力、适应性和鲁棒性, 在 PEG 问题中引起了极大关注. Eklund 等[16] 针对固定翼无人机的 PEG, 提出了一种非线性 MPC 算法, 实现了自主避障和追击机动. de Simone 等[17] 将 MPC 算法应用于解决障碍环境中无人车的 PEG, 并且证明了算法的鲁棒性和安全性. 考虑到避障需求, Sani 等[18] 提出了一种可以在博弈论和 MPC 算法之间交替转换策略的算法, 降低了计算复杂度. Sani 等[19] 针对有限信息约束, 提出了一种基于 MPC 的 PEG 算法, 使玩家

收稿日期: 2024-08-30；**录用日期:** 2024-12-25.
基金项目: 国家自然科学基金项目 (62073264).
责任编委: 高会军.
†通信作者. E-mail: penglinhu@mail.nwpu.edu.cn.

能够利用有限信息预测对手在障碍环境中的运动策略. Manoharan 等[20] 为了解决具有状态和控制约束的多对多 PEG 问题, 开发了一种基于非线性 MPC 和阿氏圆的追逃策略.

然而, 传统的控制算法, 如 MPC 算法通常依赖于精确的数学模型来设计控制律, 实际应用中经常涉及随机干扰因素, 如传感器误差和环境干扰, 难以获取准确的控制模型, 为智能体的控制带来了挑战. 在这种背景下, RL 算法成为解决 PEG 问题的重要方法. Yu 等[21] 提出了一种分布式深度 Q 网络算法, 通过分布式学习框架来增强智能体之间的协作追踪或逃避策略. Wang 等[22] 提出了一种基于 RL 的分布式合作 PEG 算法, 并设计了通信网络以节省通信和计算资源. Kartal 等[23] 应用积分强化学习, 实现了 PEG 场景中在线和实时的最优动作选择. Yang 等[24] 采用 DDPG 算法的两阶段追踪策略, 利用实时反馈信息解决 PEG 问题. Zhang 等[25] 将预测网络集成到 RL 框架中, 提高了算法在 PEG 场景中的性能和智能体的预测能力. Selvakumar 等[26] 利用 Q 学习和矩阵博弈论将非零和 PEG 转化为多个双玩家静态零和 PEG, 然后确定了复杂场景中智能体的最优动作.

传统的 RL 算法为复杂控制问题的建模和分析提供了框架, 但是可解释性和模糊信息处理能力等成为 RL 新的挑战. 模糊强化学习可以有效处理不精确和模糊的信息, 特别是对于不确定的环境感知和决策制定具有很好的效果. 模糊逻辑固有的灵活性使其能够熟练地处理复杂关系, 可以有效处理非线性和非凸性等复杂问题, 并且模糊强化学习产生的规则和推理过程是直观且易于理解的. 近年来, 人们提出了许多将模糊逻辑原理与传统学习方法相结合的学习算法. Camci 等[27] 提出了一种结合模糊逻辑控制和 RL 技术的控制算法, 该算法能够适应不同的初始位置和噪声条件. Awheda 等[28] 利用模糊行动者-评论家学习 (FACL) 算法为智能体获取控制策略, 并使用编队控制机制捕获更高级的逃避者, 有效解决了多对一 PEG 问题. Wang 等[29] 将模糊逻辑控制与 RL 相结合, 解决了连续空间中的 PEG 问题. Liu 等[30] 针对 Q 学习中连续状态和动作空间的挑战, 提出了一种将模糊逻辑控制和 Q 学习相结合的 PEG 算法.

通过文献分析可以发现, 现有方法在解决 PEG 问题时往往存在局限性, 如依赖于精确的数学模型、难以处理不确定性等. 而模糊 RL 和 MPC 的结合则能够克服这些局限性, 提供一种更加灵活和鲁棒的控制方案. 因此, 本文提出一种基于 FACL 和 MPC 的分层控制框架, 用以解决基于四旋翼的 PEG. 该框架由顶层感知与决策和底层鲁棒控制组成. 在顶层, 利用 FACL 算法来处理感知和决策任务, FACL 算法通过其内置的模糊逻辑处理能力, 将模糊信息转化为可用于决策的有用信息, 并且预测出逃避者最可能的未来轨迹; 然后结合当前环境、逃避者的位置和速度以及追击者的动力学特性等因素, 生成追击者的行动策略轨迹. 在底层, 采用 MPC 算法对四旋翼飞行器进行实时控制, FACL 生成的行动策略轨迹在 MPC 框架中作为参考状态输入, 指导控制目标的制定. 利用 MPC 强大的约束处理能力, 确保四旋翼在复杂的 PEG 场景中不违反物理约束和飞行安全要求.

本文主要贡献有: 1) 为 3D 空间的阿氏圆提供明确的解析形式, 并借助阿氏圆推导智能体的主导区域. 2) 与现有方法不同, 采用 FACL 算法输出的预测轨迹为 MPC 提供参考信号来设计四旋翼控制器, 实现四旋翼的鲁棒控制. 文章内容安排如下: 首先介绍基于 FACL 算法的感知和决策; 然后介绍基于 MPC 算法的四旋翼鲁棒控制; 最后通过仿真和飞行实验验证算法的有效性.

1 基于 FACL 的感知与决策

1.1 追逃博弈模型

假设在追逃博弈中, 追击者和逃避者具有相似的动力学特性和运动能力, 如图 1 所示, 图中 P_t、E_t、O分别为追击者、逃避者、障碍物. 在 3D 空间中建立智能体的运动学模型为

$$\begin{aligned} x_{t+1} &= x_t + v\sin\theta\cos\alpha,\\ y_{t+1} &= y_t + v\sin\theta\sin\alpha,\\ z_{t+1} &= z_t + v\cos\theta. \end{aligned} \tag{1}$$

其中: (x_t, y_t, z_t)为智能体的位置, v为智能体的速度, α为速度v在x-y平面的映射向量与x轴的夹角, θ为速度v与z轴的夹角. $U=[\alpha,\theta]^{\mathrm{T}}$为智能体的转向角. 为了使智能体的运动符合实际约束, 转向角变化量$\Delta\alpha$和$\Delta\theta$限制在区间$[-\pi/4,\pi/4]$.

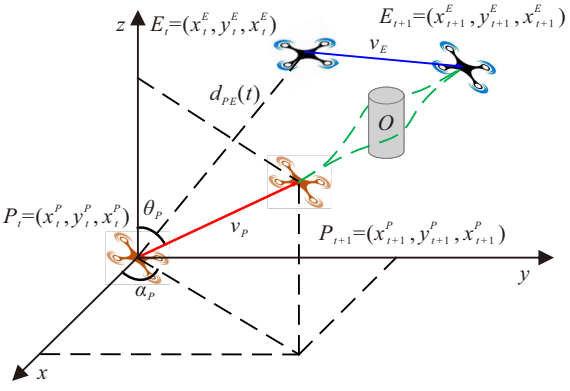


图1 PEG 模型

智能体参与追铺或躲避的能力取决于算法的输入信息, 即智能体的位置和速度. 因此, 用阿氏圆来描述双方的几何关系, 确定智能体的主导区域, 并制定运动策略. 假设追击者的速度v_P比逃避者的速度v_E快, 即$v_P > v_E$. 如图 2 所示, 与现有研究不同[31-32], 本文将阿氏圆从 2D 空间扩展到 3D 空间中的广义形式, 并给出其解析表达式. 其中A, B, C和D是广义阿氏圆O_{AC}上的点, 且直线PA和PB与广义阿氏圆O_{AC}相切, 满足$PA\perp AO_{AC}$, $PB\perp BO_{AC}$, 由两直线旋转形成的锥面是智能体的捕获区域和逃逸区域的边界. 在 3D 空间中给定追击者的位置(x_t^P, y_t^P, z_t^P)和逃避者的位置(x_t^E, y_t^E, z_t^E), 以及智能体的速度比a, 可得

$$a=\frac{v_E}{v_P}=\frac{|EA|}{|PA|}=\frac{|EB|}{|PB|}=\frac{|EC|}{|PC|}<1, \tag{2}$$

其中$|EA|$表示逃避者E与点A之间的欧氏距离. 广义阿氏圆的圆心O_{AC}和半径R_{AC}为

$$O_{AC}=\left(\frac{x_t^E-a^2x_t^P}{1-a^2}, \frac{y_t^E-a^2y_t^P}{1-a^2}, \frac{z_t^E-a^2z_t^P}{1-a^2}\right),$$

$$R_{AC}=\frac{a\sqrt{(x_t^P-x_t^E)^2+(y_t^P-y_t^E)^2+(z_t^P-z_t^E)^2}}{|1-a^2|}.$$

在图 2 黄色球面包围的空间内, 逃避者始终比追击者更早到达, 因此是逃避者的主导区域. 相反, 球面外部区域是追击者的主导区域.

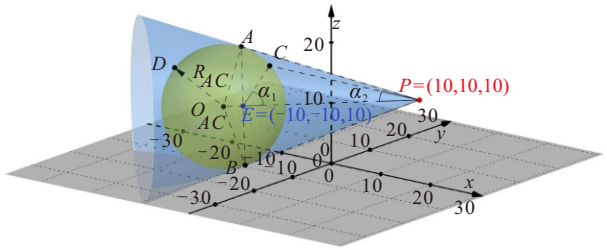


图2　3D 空间广义阿氏圆

为了确保追击者总是朝着追捕逃避者的方向移动, 并且逃避者尽可能地避开追击者的追捕, 定义基于位置关系的智能体运动策略. 对于广义阿氏圆上的任意点C, 有

$$\frac{\sin\alpha_1}{\sin\alpha_2}=\frac{v_P}{v_E}. \tag{3}$$

其中: α_1为向量$\overrightarrow{EP}$与向量$\overrightarrow{EC}$之间的夹角, α_2为向量$\overrightarrow{PE}$与向量$\overrightarrow{PC}$之间的夹角. 可得

$$\alpha_2=\sin^{-1}\left(\frac{v_E}{v_P}\sin\alpha_1\right). \tag{4}$$

当$\alpha_1=\dfrac{\pi}{2}$, 获得最优解α_2^*, 有

$$\alpha_2^*=\sin^{-1}\left(\frac{v_E}{v_P}\right). \tag{5}$$

因此, 双方智能体的最优运动空间都位于圆锥面包围的区域内. 当$a<1$时, 追击者在圆锥包络内运动, 且满足$\alpha_2\leqslant\alpha_2^*=\sin^{-1}(v_E/v_P)$. 逃避者在圆锥包络的剩余区域内运动, 且满足$\alpha_1\geqslant\pi/2$.

定义 PEG 的终端条件如下: 如果追击者P与逃避者E之间的距离满足$d_{PE}(t_f)\leqslant d_s$, 则 PEG 在时刻t_f结束, 其中d_s表示捕获距离. 此外, 如果持续时间超过指定阈值, 则认为追逃博弈过程结束.

1.2　FACL 算法

FACL 是一种将模糊逻辑和 actor-critic 强化学习框架相结合的算法, 用于处理不确定环境中的复杂决策任务. 本文采用经典的 actor-critic 框架, 在 actor-critic 框架的基础上引入模糊逻辑来处理感知和决策任务中的模糊信息. 其中 actor 基于模糊逻辑生成当前状态的行动策略, critic 用于评估 actor 生成动作的优劣并指导策略训练.

为每个智能体的 FACL 算法定义 4 个输入, 追击者的输入为

$$\bar{x}_P=[d_{PE}, \delta_P, d_{PO}, \delta_{PO}]. \tag{6}$$

其中: d_{PE}为追击者与逃避者之间的距离, δ_P为追击者的航向与从追击者到逃避者的向量$\overrightarrow{PE}(t)$之间的夹角, d_{PO}为追击者与障碍物之间的距离, δ_{PO}为追击者的航向与从追击者到障碍物的向量$\overrightarrow{PO}(t)$之间的夹角. 逃避者 FACL 系统输入为

$$\bar{x}_E=[d_{PE}, \delta_E, d_{EO}, \delta_{EO}]. \tag{7}$$

其中: δ_E为逃避者的航向与从逃避者到追击者的向量$\overrightarrow{EP}(t)$之间的夹角, d_{EO}为逃避者与障碍物之间的距离, δ_{EO}为逃避者的航向与从逃避者到障碍物的向量$\overrightarrow{EO}(t)$之间的夹角. 下面介绍 FACL 算法的更新过程.

本文中, actor 和 critic 都采用一阶 Takagi-Sugeno 规则实现模糊推理系统. 假设系统有n个输入$\bar{x}=[x_1, \ldots, x_n]$, 则 actor 的输出为

$$u_t=\sum_{l=1}^{L}\Phi_t^l w_t^l. \tag{8}$$

其中: u_t为t时刻的控制信号; L为规则总数; $w_t^l=\max\limits_{u_t, x_i\in\bar{x}}\mu^{A_i^l}(x_i)$为 actor 在$t$时刻关于规则$l$的输出参数, $\mu^{A_i^l}$为模糊集A_i^l的隶属度; Φ_t^l为规则l的触发强度, 定义为

$$\Phi_t^l=\prod_{i=1}^{n}\mu^{A_i^l}(x_i)\Big/\sum_{l=1}^{L}\left(\prod_{i=1}^{n}\mu^{A_i^l}(x_i)\right). \tag{9}$$

actor 执行动作后, critic 通过计算近似值$\hat{V}_t$来评估动作的质量, critic 的输出为

$$\hat{V}_t = \sum_{l=1}^{L} \Phi_t^l \zeta_t^l, \tag{10}$$

其中ζ_t^l为 critic 在t时刻关于规则l的输出参数. TD 误差δ_t定义为

$$\delta_t = r_{t+1} + \gamma \hat{V}_{t+1} - \hat{V}_t. \tag{11}$$

其中: r_{t+1}为奖励值, δ_t用来更新 actor 和 critic 的参数. 为控制信号u_t添加白噪声$N_0 \sim (0, \sigma_a^2)$来提高对动作空间的探索, actor 的输出参数更新方式为

$$w_{t+1}^l = w_t^l + \alpha_a \delta_t \left(\frac{u_t' - u_t}{\sigma_a}\right) \frac{\partial u_t}{\partial w_t^l}. \tag{12}$$

其中: α_a为 actor 的学习率, $u_t' = u_t + N_0$, $\dfrac{\partial u_t}{\partial w_t^l}$为

$$\frac{\partial u_t}{\partial w_t^l} = \frac{\prod_{i=1}^{n} \mu^{A_i^l}(x_i)}{\sum_{l=1}^{L}\left(\prod_{i=1}^{n} \mu^{A_i^l}(x_i)\right)} = \Phi_t^l. \tag{13}$$

critic 的输出参数ζ_{t+1}^l更新方式为

$$\zeta_{t+1}^l = \zeta_t^l + \alpha_c \delta_t \frac{\partial \hat{V}_t}{\partial \zeta_t^l}. \tag{14}$$

其中: α_c为 critic 的学习率, $\dfrac{\partial \hat{V}_t}{\partial \zeta_t^l}$为

$$\frac{\partial \hat{V}_t}{\partial \zeta_t^l} = \frac{\prod_{i=1}^{n} \mu^{A_i^l}(x_i)}{\sum_{l=1}^{L}\left(\prod_{i=1}^{n} \mu^{A_i^l}(x_i)\right)} = \Phi_t^l. \tag{15}$$

本文设置学习率$\alpha_a < \alpha_c$, 使得 actor 的收敛速度慢于 critic, 防止 actor 出现不稳定情况.

不同于已有的奖励函数设计方法[33], 本文基于人工势场的思想设计奖励函数, 障碍物对智能体产生斥力, 而逃避者对追击者产生引力. 首先定义障碍物对智能体的排斥力

$$r_{PO} = \exp(-\alpha_r \mathrm{d_{PO}}), \tag{16}$$

其中α_r为控制排斥力强度的系数. 逃避者对追击者表现出吸引力

$$r_{PE} = \frac{\beta_a}{d_{PE}}, \tag{17}$$

其中β_a为控制吸引力强度的系数, 对于逃避者则表述为$-r_{PE}$. 考虑追击者是否成功捕获了逃避者, 设计捕获成功触发奖励函数

$$r_s = \gamma_s g_s. \tag{18}$$

其中: γ_s为基于成功的奖励系数; g_s为指标函数, 当成功捕获逃避者时$g_s = 1$, 否则$g_s = 0$. 将上述奖励进行加权平衡, 制定综合奖励函数

$$r_{\mathrm{total}} = w_r r_{PO} + w_a r_{PE} + r_s, \tag{19}$$

其中权重w_r和w_a用于平衡排斥力和吸引力. 在式 (11) 中, 令$r_{t+1} = r_{\mathrm{total}}$, 通过调整权重, 可以控制奖励函数的整体行为. 将 FACL 算法概括为算法 1.

算法 1 模糊行动者-评论家学习算法

1: 初始化: $\bar{x}_0 = [x_1(0), \ldots, x_n(0)]$, $l \in \{1, 2, \ldots, L\}$, $\hat{V}_0 = 0$, $\zeta_0^l = 0$, $w_0^l = 0$, $\alpha_a < \alpha_c$, 奖励函数r_0

2: for 每个时间步长t do

3: 获得输入$\bar{x}_t = [x_1(t), \ldots, x_n(t)]$

4: 通过式 (8) 计算 actor 的输出u_t

5: 在t时刻执行动作u_t

6: 获得奖励值r_{t+1}和新的输入$\bar{x}_{t+1}$

7: 通过式 (10) 计算 critic 的输出$\hat{V}_t$和$\hat{V}_{t+1}$

8: 通过式 (11) 计算 TD 误差δ_t

9: 通过式 (12) 和 (14) 更新w_{t+1}^l和ζ_{t+1}^l

10: end for

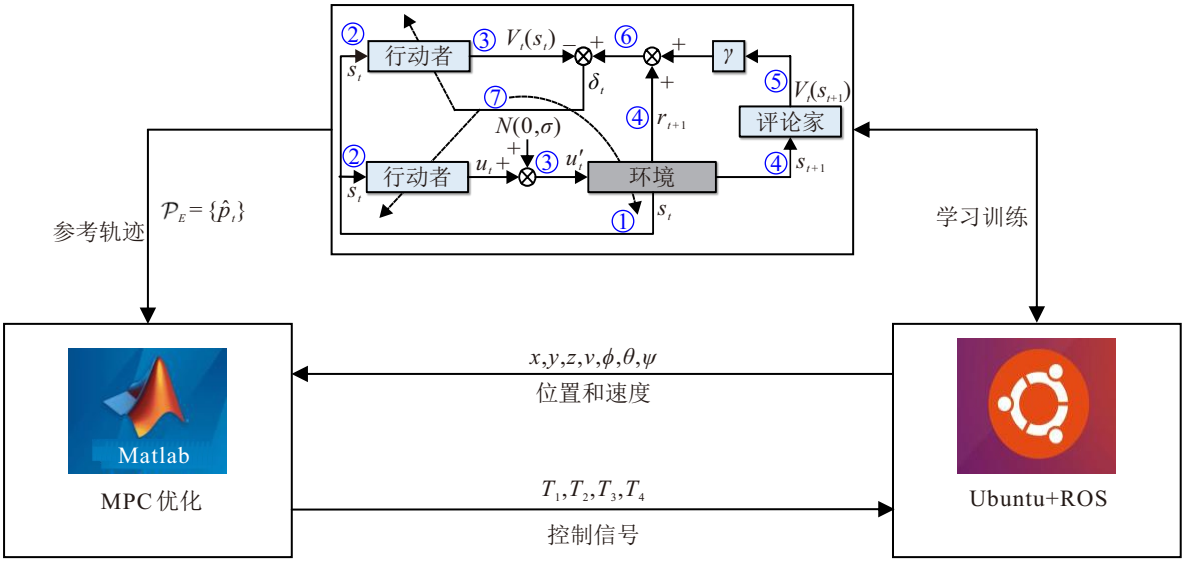


图3 四旋翼追逃博弈分层控制框架

综上, 通过同步训练 FACL 算法, 追击者在每个时间步可以预测逃避者的运动策略, 这些预测值代表了逃避者在未来一段时间内可能的状态信息, 从而形成一个对逃避者未来轨迹的预测值集合, 将其定义为参数化形式

$$\mathcal{P}_E=\{\hat{p}_t\in\mathbb{R}^3|\hat{p}_t=[s_x,s_y,s_z]^{\mathrm{T}}\},\tag{20}$$

其中$\hat{p}_t$为轨迹参数. 本文的目标是为具有控制约束的四旋翼设计鲁棒控制器, 实现稳定和精确的跟踪控制. 如图 3 所示, 设计一种分层控制框架, 首先使用 FACL 算法训练获得智能体的控制策略, 然后将获得的逃避者的运动轨迹预测值作为 MPC 算法的参考输入, 在每个控制循环中, MPC 算法结合四旋翼的当前位置和速度信息以及由 FACL 提供的预测轨迹, 生成追击者四旋翼的控制信号.

2　基于 MPC 的四旋翼鲁棒控制

2.1　四旋翼模型及控制问题描述

在惯性坐标系$\mathcal{I}=\{I_x,I_y,I_z\}$中, 四旋翼的质心、线速度和姿态角分别为$\ell=[x,y,z]^{\mathrm{T}}$, $v_I=[\dot{x},\dot{y},\dot{z}]^{\mathrm{T}}$, $\eta=[\phi,\theta,\psi]^{\mathrm{T}}$. 在四旋翼机体坐标系$\mathcal{B}=\{B_x,B_y,B_z\}$中, 速度和姿态角速度表示为$v_B=[u,v,w]^{\mathrm{T}}$, $\omega_B=[p,q,r]^{\mathrm{T}}$. 在初始状态下, 假设机体坐标系与惯性坐标系重合, 在运动过程中, 为了将机体坐标系与惯性坐标系对齐, 依次围绕坐标轴I_x、I_y、I_z, 以滚转角ϕ、俯仰角θ和偏航角ψ旋转惯性坐标系. 从机体坐标系到惯性坐标系的旋转矩阵为

$$\boldsymbol{R}(\eta)=\begin{bmatrix}\mathrm{C}\psi\mathrm{C}\theta & \mathrm{C}\psi\mathrm{S}\theta\mathrm{S}\phi-\mathrm{S}\psi\mathrm{C}\phi & \mathrm{C}\psi\mathrm{S}\theta\mathrm{C}\phi+\mathrm{S}\psi\mathrm{S}\phi\\ \mathrm{S}\psi\mathrm{C}\theta & \mathrm{S}\psi\mathrm{S}\theta\mathrm{S}\phi+\mathrm{C}\psi\mathrm{C}\phi & \mathrm{S}\psi\mathrm{S}\theta\mathrm{C}\phi-\mathrm{C}\psi\mathrm{S}\phi\\ -\mathrm{S}\theta & \mathrm{C}\theta\mathrm{S}\phi & \mathrm{C}\theta\mathrm{C}\phi\end{bmatrix}.$$

其中: $\mathrm{C}\cdot=\cos(\cdot)$, $\mathrm{S}\cdot=\sin(\cdot)$. 四旋翼的平移和旋转运动学模型描述为

$$\begin{aligned}&v_I=\boldsymbol{R}(\eta)v_B,\\&\dot{\eta}=\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix}\omega_B.\end{aligned}\tag{21}$$

对姿态角进行合理近似, 旋转模型可简化为

$$\dot{\eta}=\omega_B.\tag{22}$$

简化后的四旋翼运动学模型为

$$\begin{aligned}&m\ddot{\ell}=\boldsymbol{R}(\eta)T_tI_z-mgI_z,\\&\tau_B=\boldsymbol{I}\ddot{\eta}+\dot{\eta}\times\boldsymbol{I}\dot{\eta}.\end{aligned}\tag{23}$$

其中: m为四旋翼的质量; I_z为沿着z轴的单位向量; g为重力加速度; T_t为由转子T_1、T_2、T_3、T_4产生的总推力; $\tau_B=[\tau_\phi,\tau_\theta,\tau_\psi]^{\mathrm{T}}$为控制力矩, 滚转$\tau_\phi=d(T_4-T_2)$, 俯仰$\tau_\theta=d(T_1-T_3)$, 偏航$\tau_\psi=Q_c(T_2+T_4-T_1-T_3)$, d为四旋翼机翼长度, Q_c为阻力系数; $\boldsymbol{I}$为四旋翼的转动惯量. 实际控制输入与系统模型的控制输入之间存在以下转换关系:

$$\begin{bmatrix}T_t\\ \tau_B\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & 1\\ 0 & -d & 0 & d\\ d & 0 & -d & 0\\ -Q_c & Q_c & -Q_c & Q_c\end{bmatrix}\begin{bmatrix}T_1\\ T_2\\ T_3\\ T_4\end{bmatrix}.\tag{24}$$

根据以上分析, 可以用问题 1 来描述追击者四旋翼的控制问题.

问题 1　给定系统模型 (23)、预测轨迹 (20), 以及轨迹参数$\hat{p}_t$, 控制目标是计算控制输入T_t和控制力矩τ_B, 使得下式成立:

$$\lim_{t\to t_f}\|\ell(t)-\hat{p}_t\|=0,\tag{25}$$

其中t_f为终端时间, 且控制系统的总推力T_t满足约束$0\leqslant T_t\leqslant T_{\max}$.

下面对问题 1 进行求解, 四旋翼无人机的控制输入为 4 个旋翼产生的推力, 而输出状态包括位置和姿态 6 个控制变量, 并且存在复杂的耦合关系, 因此四旋翼飞行器是典型的欠驱动系统. 为了有效地控制四旋翼在惯性坐标系中运动, 对系统模型进行解耦. 对位置状态ℓ进行分析, 有

$$\begin{aligned}\ddot{x}&=\frac{1}{m}(\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi)T_t,\\ \ddot{y}&=\frac{1}{m}(\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi)T_t,\\ \ddot{z}&=\frac{1}{m}(\cos\theta\cos\phi)T_t-g.\end{aligned}\tag{26}$$

定义中间变量$u_x=\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi$和$u_y=\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi$, 则位置控制系统描述为

$$\dot{\boldsymbol{x}}=\begin{bmatrix}\dot{\ell}\\ \dot{v}_I\end{bmatrix}=\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ u_x\dfrac{T_t}{m}\\ u_y\dfrac{T_t}{m}\\ \cos\theta\cos\phi\dfrac{T_t}{m}-g\end{bmatrix}=f(\boldsymbol{x},\boldsymbol{u}_\ell).\tag{27}$$

其中: $\boldsymbol{u}_\ell=[u_x,u_y,T_t]^{\mathrm{T}}$为位置系统的等效控制输入, $[u_x,u_y]^{\mathrm{T}}$为x轴和y轴的控制输入, $u_x\dfrac{T_t}{m}$、$u_y\dfrac{T_t}{m}$和$\cos\theta\cos\phi\dfrac{T_t}{m}-g$分别为由控制输入产生的沿惯性坐标轴的线性加速度. 计算期望的滚转角ϕ_d和俯仰角θ_d为

$$\phi_d = \arcsin(u_x \sin\psi_d + u_y \cos\psi_d),$$
$$\theta_d = \arcsin\frac{u_x - \sin\phi_d \sin\psi_d}{\cos\phi_d \cos\psi_d}. \tag{28}$$

根据位置系统 (27), 高度控制器的设计与u_x、u_y无关, 是可以解耦出来的, 因此关于四旋翼的 MPC 控制器可以分解为高度控制器、平移控制器和姿态控制器.

定义追击者关于逃避者四旋翼的参考系统为

$$\dot{\boldsymbol{x}}_r = f(\boldsymbol{x}_r, \boldsymbol{u}_{\ell_r}). \tag{29}$$

其中: $\dot{\boldsymbol{x}}_r = [\dot{\ell}_r, \dot{v}_{I_r}]^{\mathrm{T}}$为系统参考状态, $\boldsymbol{u}_{\ell_r} = [u_{x_r}, u_{y_r}, T_{t_r}]^{\mathrm{T}}$为参考控制输入. 假设四旋翼的高度是固定的, 根据给定的期望位置轨迹, 参考控制输入可以定义为

$$\boldsymbol{u}_{\ell_r} = \Big[\frac{m\ddot{x}_r}{T_{t_r}}, \frac{m\ddot{y}_r}{T_{t_r}}, m(\ddot{z}_r + g)\Big]^{\mathrm{T}}. \tag{30}$$

通过比较参考系统 (29) 和位置系统 (27), 可以得到轨迹跟踪误差系统矩阵为$\boldsymbol{e} = [e_x, e_y, e_z]^{\mathrm{T}}$, 其中$e_x$、$e_y$、$e_z$定义如下:

$$e_x = \begin{bmatrix} x - x_r \\ \dot{x} - \dot{x}_r \\ \int (x - x_r)\mathrm{d}t \end{bmatrix},\ e_y = \begin{bmatrix} y - y_r \\ \dot{y} - \dot{y}_r \\ \int (y - y_r)\mathrm{d}t \end{bmatrix},$$
$$e_z = \begin{bmatrix} z - z_r \\ \dot{z} - \dot{z}_r \\ \int (z - z_r)\mathrm{d}t \end{bmatrix}. \tag{31}$$

与常见的误差设计方法不同, 在式 (31) 中考虑了误差积分项, 以增强控制器鲁棒性.

2.2 高度控制器设计

根据式 (29), 高度参考系统为

$$\dot{\boldsymbol{x}}_{r,z} = \begin{bmatrix} \dot{z}_r \\ \cos\theta\cos\phi\dfrac{T_{t_r}}{m} - g \end{bmatrix}, \tag{32}$$

其中$\cos\theta\cos\phi$可以看作是控制器的时变参数. 结合误差系统矩阵$\boldsymbol{e}$和式 (32), 得到z轴轨迹跟踪误差的状态方程为

$$\dot{e}_z = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} e_z + \begin{bmatrix} 0 \\ \dfrac{\cos\theta\cos\phi}{m} \\ 0 \end{bmatrix} \tilde{u}_z, \tag{33}$$

其中$\tilde{u}_z = T_t - T_{t_r}$, 目标是求解控制输入$\tilde{u}_z$, 使得$\lim\limits_{t\to\infty} e_z = \boldsymbol{0}$. 将在时刻$t_k$求解的 MPC 优化问题描述为

$$\min_{\hat{\tilde{u}}_{z,\cdot|k}} J_z = \sum_{i=0}^{N_z-1} (\|\hat{e}_{z,i|k}\|^2_{Q_z} + \|\hat{\tilde{u}}_{z,i|k}\|^2_{R_z}) + \|\hat{e}_{z,N_z|k}\|^2_{P_z}. \tag{34a}$$

$$\text{s.t. } \hat{e}_{z,i+1|k} = A_z\hat{e}_{z,i|k} + B_z\hat{\tilde{u}}_{z,i|k}, \tag{34b}$$
$$\hat{e}_{z,0|k} = e_{z,k}, \tag{34c}$$
$$\hat{\tilde{u}}_{z,i|k} \in \mathcal{U}. \tag{34d}$$

其中: N_z为预测时域, Q_z、R_z、P_z为权重矩阵. 对跟踪误差预测方程 (34b) 离散化处理, 得到A_z和B_z分别为

$$A_z = \begin{bmatrix} 1 & \delta & 0 \\ 0 & 1 & 0 \\ \delta & 0 & 1 \end{bmatrix},\ B_z = \begin{bmatrix} 0 \\ \dfrac{\delta}{m}\cos\theta_k\cos\phi_k \\ 0 \end{bmatrix}. \tag{35}$$

其中: δ为采样周期, $e_{z,k} = e_z(t_k)$为状态量测的当前时间反馈, $\mathcal{U} = \{\tilde{u}_z | \tilde{u}_{\min} \leqslant \tilde{u}_z \leqslant \tilde{u}_{\max}\}$为误差系统的控制输入约束.

根据线性状态空间预测模型 (34b) 和预测输入序列$U_{z,k} = [\hat{\tilde{u}}_{z,0|k}, \hat{\tilde{u}}_{z,1|k}, \ldots, \hat{\tilde{u}}_{z,N_z-1|k}]^{\mathrm{T}}$得到预测状态序列的紧凑形式如下:

$$E_{z,k} = M_z e_{z,k} + C_z U_{z,k}. \tag{36}$$

其中: $E_{z,k} = [\hat{e}_{z,0|k}, \ldots, \hat{e}_{z,N_z|k}]^{\mathrm{T}}$, $M_z = [A_z, A_z^2, \ldots, A_z^{N_z}]^{\mathrm{T}}$, $C_z = \begin{bmatrix} B_z & 0 & \cdots & 0 \\ A_zB_z & B_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_z^{N_z-1}B_z & A_z^{N_z-2}B_z & \cdots & B_z \end{bmatrix}$.

因此, 性能指标函数 (34a) 可以转化为

$$J_z = U_{z,k}^{\mathrm{T}} H_z U_{z,k} + 2e_{z,k}^{\mathrm{T}} F_z^{\mathrm{T}} U_{z,k} + e_{z,k}^{\mathrm{T}} G_z e_{z,k}, \tag{37}$$

其中H_z, F_z, G_z定义为

$$H_z = C_z^{\mathrm{T}}\bar{Q}_z C_z + \bar{R}_z,$$
$$F_z = C_z^{\mathrm{T}}\bar{Q}_z M_z,$$
$$G_z = M_z^{\mathrm{T}}\bar{Q}_z M_z + \bar{Q}_z.$$

权重矩阵定义为

$$\bar{Q}_z = \begin{bmatrix} Q_z & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & 0 & Q_z & 0 \\ 0 & 0 & 0 & P_z \end{bmatrix},$$
$$\bar{R}_z = \begin{bmatrix} R_z & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 \\ \vdots & 0 & R_z & 0 \\ 0 & 0 & 0 & R_z \end{bmatrix}.$$

首先分析无约束问题$U_{z,k}^* = \arg\min\limits_{U_{z,k}} J_z$的解, 计算梯度$\nabla_{U_{z,k}} J_z$有

$$\nabla_{U_{z,k}} J_z = 2(H_z U_{z,k} + F_z e_{z,k}). \tag{38}$$

如果H_z是正定矩阵, 令$\nabla_{U_{z,k}} J_z = 0$, 则最优控制序列为$U_{z,k}^* = -H_z^{-1}F_z e_{z,k}$. 如果$H_z$是半正定矩阵, 则

可以得到一个广义最优解$U_{z,k}^{*}=-H_z^{\sim}F_z e_{z,k}$, 其中$H_z^{\sim}$是$H_z$的左逆矩阵, 满足$H_z^{\sim}H_z=I$. 对于有约束优化问题 (34), 可利用二次规划进行求解, 得到最优控制输入序列数值解$U_{z,k}^{*}$. 因此, t_k时刻高度控制器信号为

$$T_{t,k}=\hat{\tilde{u}}_{z,0|k}^{*}+m(\ddot{z}_{r,k}+g). \tag{39}$$

2.3　平移控制器设计

根据式 (29), 沿x轴和y轴的平移参考系统为

$$\dot{x}_{r,xy}=\left[\dot{x}_r,u_{x_r}\frac{T_{t_r}}{m},\dot{y}_r,u_{y_r}\frac{T_{t_r}}{m}\right]^{\mathrm{T}}, \tag{40}$$

其中T_{t_r}可以看作平移控制器的时变参数. 轨迹跟踪误差的状态方程为

$$\dot{e}_{xy}=\begin{bmatrix}0&1&0&&&\\0&0&0&&\boldsymbol{0}_{3\times3}&\\1&0&0&&&\\&&&0&1&0\\&\boldsymbol{0}_{3\times3}&&0&0&0\\&&&1&0&0\end{bmatrix}e_{xy}+\begin{bmatrix}0&0\\\frac{T_t}{m}&0\\0&0\\0&0\\0&\frac{T_t}{m}\\0&0\end{bmatrix}\tilde{u}_{xy}, \tag{41}$$

其中$\tilde{u}_{xy}=[u_x-u_{x_r},u_y-u_{y_r}]^{\mathrm{T}}$. 目标是求解控制输入$\tilde{u}_{xy}$使得$\lim\limits_{t\to\infty}e_{xy}=\boldsymbol{0}$, 将在时刻$t_k$求解的 MPC 优化问题描述为

$$\min_{\hat{\tilde{u}}_{xy,\cdot|k}} J_{xy}=\sum_{i=0}^{N_{xy}-1}(\|\hat{x}_{xy,i|k}\|_{Q_{xy}}^2+\|\hat{\tilde{u}}_{xy,i|k}\|_{R_{xy}}^2)+\|\hat{x}_{xy,N_{xy}|k}\|_{P_{xy}}^2; \tag{42a}$$

$$\text{s.t. }\hat{x}_{xy,i+1|k}=A_{xy}\hat{x}_{xy,i|k}+B_{xy}\hat{\tilde{u}}_{xy,i|k}, \tag{42b}$$

$$\hat{x}_{xy,0|k}=x_{xy|k}. \tag{42c}$$

其中: N_{xy}为预测时域, $x_{xy,k}=x_{xy}(t_k)$为t_k时刻的状态量测. A_{xy}和B_{xy}定义如下:

$$A_{xy}=\begin{bmatrix}A_x&\boldsymbol{0}_{3\times3}\\\boldsymbol{0}_{3\times3}&A_y\end{bmatrix},\ A_x=A_y=\begin{bmatrix}1&\delta&0\\0&1&0\\\delta&0&1\end{bmatrix},$$

$$B_{xy}=\begin{bmatrix}B_x&\boldsymbol{0}_{3\times1}\\\boldsymbol{0}_{3\times1}&B_y\end{bmatrix},\ B_x=B_y=\begin{bmatrix}0\\\frac{\delta}{m}T_{t,k}\\0\end{bmatrix}.$$

与高度控制器类似, 使用数值方法获得控制序列为$U_{xy,k}^{*}=[\hat{\tilde{u}}_{xy,0|k}^{*},\hat{\tilde{u}}_{xy,1|k}^{*},\dots,\hat{\tilde{u}}_{xy,N_{xy}-1|k}^{*}]^{\mathrm{T}}$. 因此, t_k时刻的平移控制器信号为

$$\begin{bmatrix}u_{x,k}\\u_{y,k}\end{bmatrix}=\hat{\tilde{u}}_{xy,0|k}^{*}+\begin{bmatrix}u_{x_r,k}\\u_{y_r,k}\end{bmatrix}. \tag{43}$$

在获得控制信号$u_{x,k}$和$u_{y,k}$后, 可以利用式 (28) 反向计算ϕ_d和θ_d以设计姿态控制器.

2.4　姿态控制器设计

为四旋翼设计姿态控制器, 定义姿态角控制误差为$e_\eta=\eta-\eta_d$, 其中η_d为期望的姿态角. 令$\xi_1=e_\eta,\xi_2=\dot{e}_\eta$, 则姿态角的跟踪误差状态向量为$\xi=[\xi_1,\xi_2]^{\mathrm{T}}$, 误差状态方程为

$$\begin{aligned}&\dot{\xi}_1=\xi_2,\\&\dot{\xi}_2=\boldsymbol{I}^{-1}(\tau_B-\dot{\eta}\times\boldsymbol{I}\dot{\eta})-\ddot{\eta}_d.\end{aligned} \tag{44}$$

需要求解控制输入τ_B, 使得$\lim\limits_{t\to\infty}\|\xi(t)\|=\boldsymbol{0}$, 实现姿态角$\eta$对期望信号$\eta_d$的精确跟踪. 令$\xi_2=\boldsymbol{s}-\rho\xi_1$, 其中$\boldsymbol{s}$是滑模面, 且$\rho>0$, 有

$$\boldsymbol{s}=c\xi_1+m\int\xi_1\mathrm{d}t+d\xi_2, \tag{45}$$

c, m, d为滑模面参数. 设计李雅普诺夫函数为

$$V_\eta=\frac{1}{2}\xi_1^{\mathrm{T}}\xi_1+\frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s}. \tag{46}$$

计算V_η的一阶偏导数

$$\begin{aligned}\dot{V}_\eta=&\xi_1^{\mathrm{T}}\dot{\xi}_1+\boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{s}}=\\&\xi_1^{\mathrm{T}}(\boldsymbol{s}-\rho\xi_1)+\boldsymbol{s}^{\mathrm{T}}(c\xi_2+m\xi_1+d\dot{\xi}_2)=\\&-\rho\xi_1^{\mathrm{T}}\xi_1+\xi_1^{\mathrm{T}}\boldsymbol{s}+\boldsymbol{s}^{\mathrm{T}}\{c(\boldsymbol{s}-\rho\xi_1)+m\xi_1+\\&d[\boldsymbol{I}^{-1}(\tau_B-\dot{\eta}\times\boldsymbol{I}\dot{\eta})-\ddot{\eta}_d]\}.\end{aligned} \tag{47}$$

为了使系统在平衡点处指数稳定, 设计如下控制律:

$$\begin{aligned}&\tau_B=\tau_1+\tau_2,\\&\tau_1=\boldsymbol{I}\{d^{-1}[(c\rho-1-m)\xi_1-c\boldsymbol{s}]+\ddot{\eta}_d\}+\dot{\eta}\times\boldsymbol{I}\dot{\eta},\\&\tau_2=\boldsymbol{I}(-\varepsilon\mathrm{sgn}(\boldsymbol{s})-\omega(\boldsymbol{s})).\end{aligned} \tag{48}$$

其中: ε,ω为指数趋近律参数, $\mathrm{sgn}(\cdot)$为符号函数. 将控制律τ_B代入式 (47) 有

$$\begin{aligned}\dot{V}_\eta=&-\rho\xi_1^{\mathrm{T}}\xi_1+\xi_1^{\mathrm{T}}\boldsymbol{s}+\mathbf{s}^{\mathrm{T}}\{-\varepsilon\mathrm{sgn}(\boldsymbol{s})-\omega\boldsymbol{s}-\xi_1\}=\\&-\rho\xi_1^{\mathrm{T}}\xi_1-\omega\mathrm{s}^{\mathrm{T}}\boldsymbol{s}-\varepsilon|\boldsymbol{s}|.\end{aligned}$$

取$\varepsilon>0$, 且令$\gamma=\min\{2\rho,2\omega\}$, 则有

$$\dot{V}_\eta\leqslant-\rho\xi_1^{\mathrm{T}}\xi_1-\omega\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s}\leqslant-\gamma V_\eta. \tag{49}$$

从而有$V_\eta(t)\leqslant V_\eta(0)\exp(-\gamma t)$, 因此可得误差系统在平衡点$\xi=\boldsymbol{0}$处指数稳定.

本文的算法流程如图 4 所示, 包括 3 个步骤.

step 1: 在t_1周期中, FACL 算法以逃避者的当前状态信息x, y, z, v, ϕ, θ, ψ为输入, 通过基于模糊逻辑的 actor-critic 学习算法生成追击者对逃避者的行动预测轨迹$\mathcal{P}_E=\{\hat{p}_t\}$.

step 2: 在t_1周期中, 以 FACL 生成的追击者的预测轨迹$\mathcal{P}_E=\{\hat{p}_t\}$为参考状态输入, 设计基于 MPC 算法的高度、平移和姿态控制器, 并生成对四

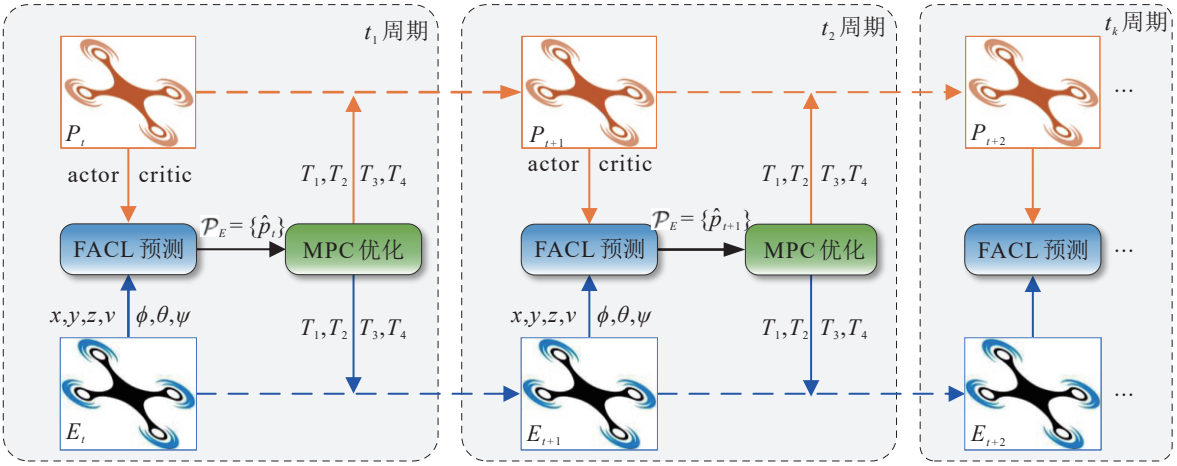
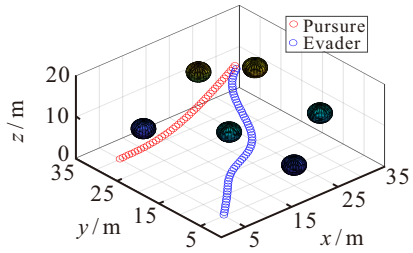


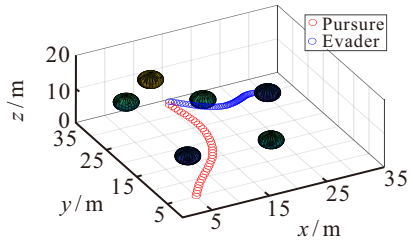
图4 基于 FACL 和 MPC 的四旋翼追逃博弈流程

旋翼无人机的控制指令.

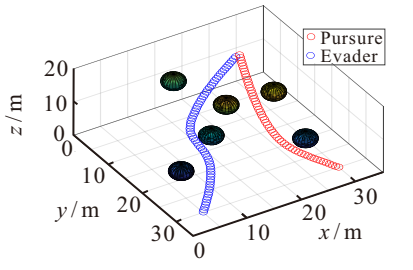
step 3: 四旋翼无人机根据 MPC 生成的控制指令进行运动, 执行追逃任务, 获得逃避者新的状态信息, 然后进入t_2周期的循环.



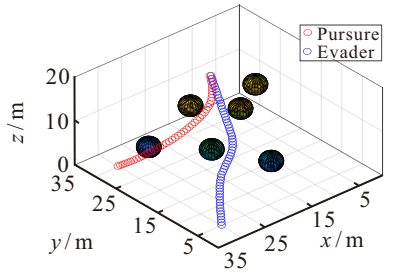
(a) 场景1



(b) 场景2



(c) 场景3



(d) 场景4

图5 3D 环境中智能体追逃博弈轨迹

3 仿真实验与分析

3.1 基于 FACL 的 PEG

智能体在大小为35 m × 35 m × 20 m的 3D 空间进行追逃博弈, 最大移动速度为$v_P = 1.1$ m/s和$v_E = 1$ m/s, 根据式 (19), 将奖励函数的参数设置为$\alpha_r = 10$, $\beta_a = 5$, $\gamma_s = 20$, $w_r = 5$, $w_a = 10$. 采样时间为0.1 s, 折扣因子为$\gamma = 0.95$, 随机白噪声为$N_0 \sim (0, 0.01)$, 学习率为$\alpha_a = 0.001$, $\alpha_c = 0.05$. 当$d_{PE} \leqslant d_s = 1$ m或当时间超过100 s时, 追逃过程终止. 为了减少模糊规则的计算量, 选择三角隶属度函数, FACL 的每项输入有 5 个三角隶属函数, 距离输入区间为$[0, 35]$, 角度输入区间为$[-\pi, \pi]$, 规则总数为$5 \times 5 \times 5 \times 5 = 625$. 每条规则的输出为转向角组成的元组$\{(\alpha, \theta)|\Delta\alpha, \Delta\theta \in [-\pi/4, \pi/4]\}$.

设置了 4 种不同场景的 PEG, 追击者的位置为$[5, 30, 0]$, $[5, 5, 0]$, $[30, 30, 0]$, $[30, 30, 0]$, 对应逃避者的位置为$[5, 5, 0]$, $[30, 30, 0]$, $[30, 5, 0]$, $[5, 30, 0]$, 障碍物是随机放置的半径为1 m的球体. 图 5 展示了追击者和逃避者的轨迹. 可以看出在 3D 环境中追击者成功地捕获了逃避者, 并且与障碍物没有发生任何碰撞. 图 6 展示了训练过程中 critic 网络的损失曲线与算法的平均奖励曲线, 可以看出在训练过程中

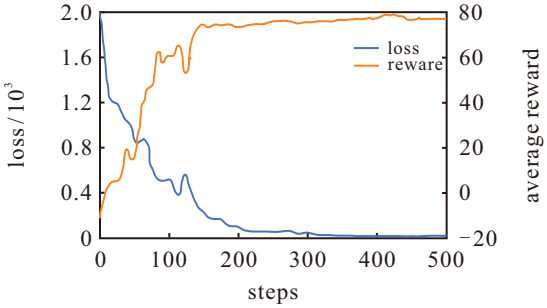


图6 训练过程中的损失曲线与奖励曲线

损失曲线整体呈下降趋势, 奖励曲线整体呈上升趋势, 虽然在某些步骤出现了峰值或者波动, 但是二者变化趋势反映出的算法逐渐收敛的特性, 仿真结果验证了 FACL 算法有效性.

3.2　基于 MPC 的四旋翼鲁棒控制

定义四旋翼参数: $m=1$ kg, $g=10$ m/s^2, $I_x=0.004$ kg·m^2, $I_y=0.004$ kg·m^2, $I_z=0.0084$ kg·m^2. 采样时间为$\delta=0.1$ s, 推力约束为$0\ \text{N}\leqslant T_t\leqslant 10\text{N}$. 设置四旋翼从$(0,0,0)$起飞, 并沿着下述参考轨迹飞行:

$$x_t(t)=\cos\left(\frac{\pi t}{10}\right),$$
$$y_t(t)=\sin\left(\frac{\pi t}{10}\right),$$
$$z_t(t)=\frac{\pi t}{10}.$$

整个仿真过程持续了20 s. 如图 7 所示, 将本文设计的考虑积分项的 MPC 算法 (图中用 IMPC 表示) 和不考虑积分项的 MPC 算法的性能进行比较. 两种算法都展示了出色的轨迹跟踪性能, 但本文算法展现出更小的超调和更好的性能. 图 8 从 3 个维度分析了轨迹跟踪误差, 可以看出, 本文所提出算法具有更小的轨迹跟踪误差和更好的鲁棒性能.

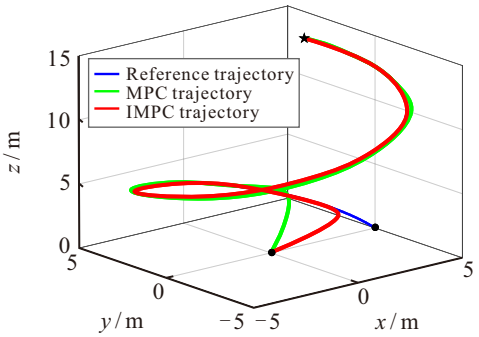


图7　四旋翼轨迹跟踪曲线

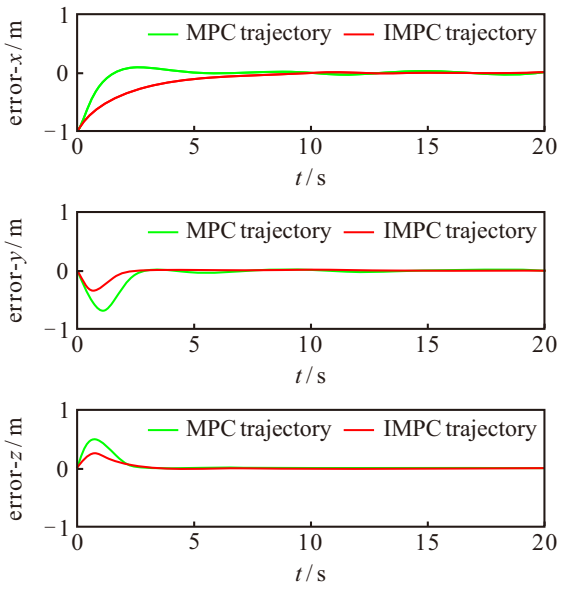


图8　四旋翼轨迹跟踪误差曲线

3.3　基于 FACL 和 MPC 的四旋翼 PEG

在 Gazebo 平台验证算法的有效性, 场景大小和速度与 3.1 节相同, 用1.5 m × 1.5 m × 15 m的立方体表示障碍物. FACL 算法获得的预测轨迹通过中心计算机发送给 Matlab 中运行的 MPC 算法, 四旋翼的控制算法在 Ubuntu 18.04 系统中运行. 整个仿真过程持续31 s, 四旋翼飞行轨迹如图 9 所示. 图 10 展示了仿真过程截图. 四旋翼在有效避开障碍物的

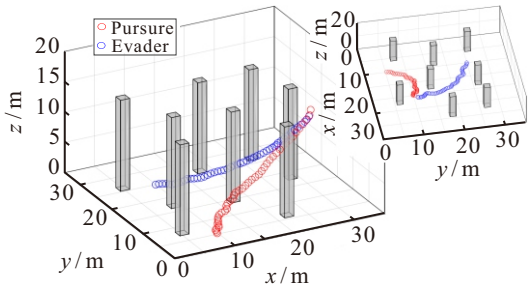
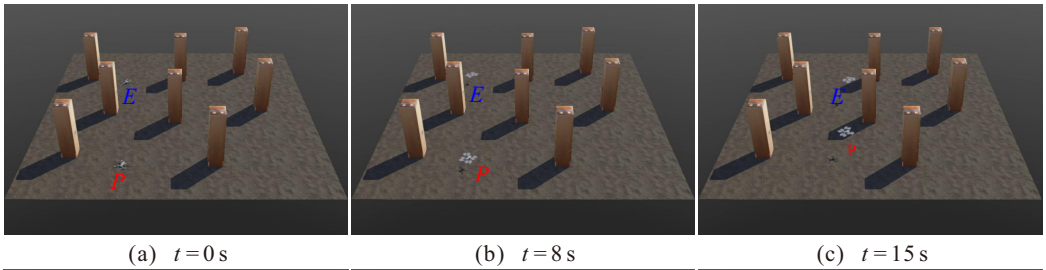
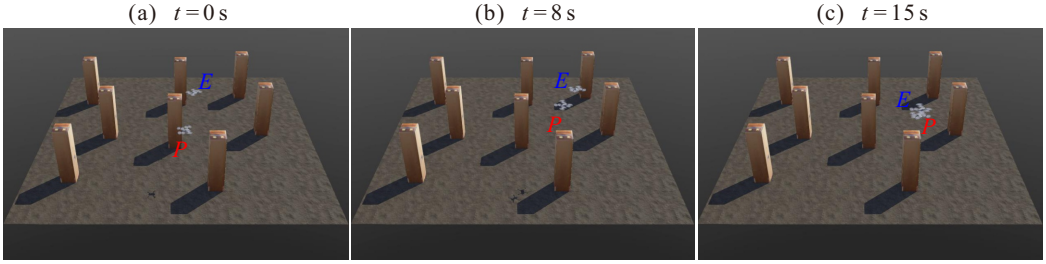


图9　四旋翼飞行轨迹, 立方体为障碍物



(a)　$t=0$ s　　(b)　$t=8$ s　　(c)　$t=15$ s



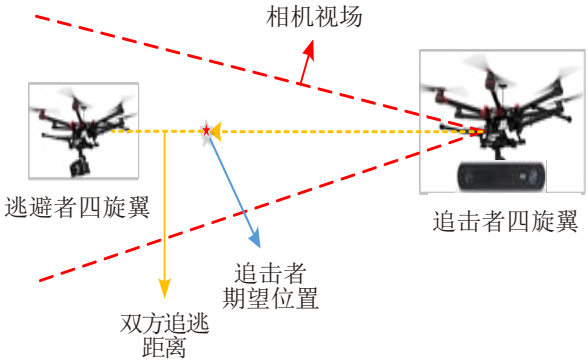
(d)　$t=20$ s　　(e)　$t=26$ s　　(f)　$t=31$ s

图10　基于 Gazebo 的四旋翼追逃博弈

同时, 成功完成了追逃任务, 验证了 FACL 和 MPC 算法联合控制四旋翼执行 PEG 的有效性和可行性.

为了验证所提出算法的实时性能和可移植性, 进行四旋翼无人机追逃博弈飞行测试. 如图 11 所示, 平台包括 2 架大疆 M100 无人机, 2 台 ZED 双目相机, 采用立体视觉对目标进行识别, 并计算出期望位置. FACL 算法在中央计算机上运行, 随后将其学习到的策略传输给四旋翼的机载计算机, 然后使用 MPC 控制器对四旋翼进行飞行控制. 出于安全原因, 四旋翼之间的安全距离设置为1 m, 捕获距离设置为间隔[1 m, 2 m], 整个实验过程持续30 s. 四旋翼之间的距离如图 12 所示, 可以看到 25 s 后, 追击者与逃避者之间的距离满足捕获条件, 同时保持大于 1 m 的安全线. 通过飞行实验验证了所提出框架的有效性, 也为实际工程应用提供了参考示例.

(a) 四旋翼平台的追逃博弈场景



(b) 四旋翼追逃感知过程

图11 四旋翼追逃博弈场景设置与感知示意图

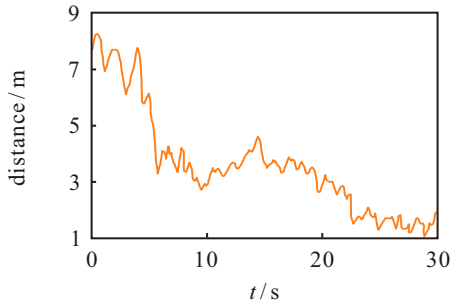


图12 四旋翼追逃博弈过程中无人机之间的距离

4 结 论

针对四旋翼无人机 PEG 问题, 本文提出了一种结合 FACL 和 MPC 的分层控制算法. 使用 3D 空间中的广义阿氏圆来确定智能体的运动主导区域. 使用 FACL 算法进行智能体训练, 获得了逃避者的预测轨迹. 以此作为参考, 为 PEG 场景中欠驱动四旋翼设计考虑积分项的 MPC 控制算法, 设计高度、平移和姿态控制器. 通过仿真验证了 FACL 和 MPC 算法的性能, 在 Gazebo 平台验证了 FACL 和 MPC 联合控制四旋翼完成 PEG 的有效性, 并在四旋翼平台对算法的实时性能和可移植性进行了验证. 本研究为 RL 算法和传统控制算法的结合提供了参考, 在未来的研究中, 将设计不同的学习算法, 以及不同的控制算法, 在相同 PEG 场景下进行联合控制的性能测试, 并通过仿真或实际飞行实验来验证所提出方案的有效性. 同时研究具有多目标优化的 RL 算法与非线性 MPC 算法相结合来解决 PEG 问题的控制算法.

参考文献 (References)

[1] WeintraubI E, Pachter M, Garcia E. An introduction to pursuit-evasion differential games[C]. American Control Conference. Piscataway: IEEE, 2020: 1049-1066.

[2] Lee E S, Shishika D, Loianno G, et al. Defending a perimeter from a ground intruder using an aerial defender: Theory and practice[C]. IEEE International Symposium on Safety, Security, and Rescue Robotics. New York, 2021: 184-189.

[3] Biediger D, Popov L, Becker A T. The pursuit and evasion of drones attacking an automated turret[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. Prague, 2021: 9677-9682.

[4] Zhang F, Zha W. Evasion strategies of a three-player lifeline game[J]. Science China Information Sciences, 2018, 61: 1-11.

[5] Yan R, Deng R L, Lai H W, et al. Homicidal chauffeur reach-avoid games via guaranteed winning strategies[J]. IEEE Transactions on Automatic Control, DOI: 10.1109/TAC.2023.3329693.

[6] Liang L, Deng F, Peng Z H, et al. A differential game for cooperative target defense[J]. Automatica, 2019, 102: 58-71.

[7] Ibragimov G, Ferrara M, Ruziboev M, et al. Linear evasion differential game of one evader and several pursuers with integral constraints[J]. International Journal of Game Theory, 2021, 50(3): 729-750.

[8] Deng Z Q, Kong Z D. Multi-agent cooperative pursuit-defense strategy against one single attacker[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 5772-5778.

[9] Garcia E, Casbeer D W, Von Moll A, et al. Multiple pursuer multiple evader differential games[J]. IEEE Transactions on Automatic Control, 2021, 66(5): 2345-2350.

[10] Yan R, Shi Z Y, Zhong Y S. Cooperative strategies for

two-evader-one-pursuer reach-avoid differential games[J]. International Journal of Solids and Structures, 2021, 52(9): 1894-1912.

[11] Yan R, Shi Z Y, Zhong Y S. Task assignment for multiplayer reach-avoid games in convex domains via analytical barriers[J]. IEEE Transactions on Robotics, 2020, 36(1): 107-124.

[12] 刘坤, 郑晓帅, 林业茗, 等. 基于微分博弈的追逃问题最优策略设计[J]. 自动化学报, 2021, 47(8): 1840-1854. (Liu K, Zheng X S, Lin Y M, et al. Design of optimal strategies for the pursuit-evasion problem based on differential game[J]. Acta Automatica Sinica, 2021, 47(8): 1840-1854.)

[13] Sani M, Hably A, Robu B, et al. Real-time game-theoretic model predictive control for differential game of target defense[J]. Asian Journal of Control, 2023, 25(5): 3343-3355.

[14] Schwartz H M. Multi-agent machine learning[M]. Hoboken: JohnWiley & Sons, 2014.

[15] 符小卫, 王辉, 徐哲. 基于 DE-MADDPG 的多无人机协同追捕策略[J]. 航空学报, 2022, 43(5): 522-535. (Fu X W, Wang H, Xu Z. Cooperative pursuit strategy for multi-UAVs based on DE-MADDPG algorithm[J]. Acta Aeronautica et Astronautica Sinica, 2022, 43(5): 522-535.)

[16] Eklund J M, Sprinkle J, Sastry S S. Switched and symmetric pursuit/evasion games using online model predictive control with application to autonomous aircraft[J]. IEEE Transactions on Control Systems Technology, 2012, 20(3): 604-620.

[17] de Simone D, Scianca N, Ferrari P, et al. MPC-based humanoid pursuit-evasion in the presence of obstacles[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver, 2017: 5245-5250.

[18] Sani M, Robu B, Hably A. Pursuit-evasion games based on game-theoretic and model predictive control algorithms[C]. International Conference on Control, Automation and Diagnosis. Grenoble, 2021: 1-6.

[19] Sani M, Robu B, Hably A. Limited information model predictive control for pursuit-evasion games[C]. The 60th IEEE Conference on Decision and Control. Piscataway: IEEE, 2021: 265-270.

[20] Manoharan A, Sujit P B. NMPC-based cooperative strategy to lure two attackers into collision by two targets[J]. IEEE Control Systems Letters, 2023, 7: 496-501.

[21] Yu C, Dong Y Z, Li Y N, et al. Distributed multi-agent deep reinforcement learning for cooperative multi-robot pursuit[J]. The Journal of Engineering, 2020, 2020(13): 499-504.

[22] Wang Y D, Dong L, Sun C Y. Cooperative control for multi-player pursuit-evasion games with reinforcement learning[J]. Neurocomputing, 2020, 412: 101-114.

[23] Kartal Y, Subbarao K, Dogan A, et al. Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning[J]. International Journal of Robust and Nonlinear Control, 2021, 31(16): 7886-7903.

[24] Yang B, Liu P X, Feng J L, et al. Two-stage pursuit strategy for incomplete-information impulsive space pursuit-evasion mission using reinforcement learning[J]. Aerospace, 2021, 8(10): 299.

[25] Zhang R, Zong Q, Zhang X, et al. Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(10): 7900-7909.

[26] Selvakumar J, Bakolas E. Min-Max Q-learning for multi-player pursuit-evasion games[J]. Neurocomputing, 2022, 475: 1-14.

[27] Camci E, Kayacan E. Game of drones: UAV pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning[C]. IEEE International Conference on Fuzzy Systems. Vancouver, 2016: 618-625.

[28] Awheda M D, Schwartz H M. A decentralized fuzzy learning algorithm for pursuit-evasion differential games with superior evaders[J]. Journal of Intelligent & Robotic Systems, 2016, 83: 35-53.

[29] Wang L X, Wang M L, Yue T. A fuzzy deterministic policy gradient algorithm for pursuit-evasion differential games[J]. Neurocomputing, 2019, 362: 106-117.

[30] Liu S Z, Hu X X, Dong K J. Adaptive double fuzzy systems based Q-learning for pursuit-evasion game[J]. IFAC-PapersOnLine, 2022, 55(3): 251-256.

[31] Huang Y, Luo Y, Nie Y, et al. Escape strategy based on apollonius circles in the pursuit-evasion game[C]. International Conference on Autonomous Unmanned Systems. Piscataway: IEEE, 2022: 2143-2153.

[32] Dorothy M, Maity D, Shishika D, et al. One Apollonius Circle is enough for many pursuit-evasion games[J]. Automatica, 2024, 163: 111587.

[33] Qu X, Gan W, Song D, et al. Pursuit-evasion game strategy of USV based on deep reinforcement learning in complex multi-obstacle environment[J]. Ocean Engineering, 2023, 273: 114016.

作者简介

胡鹏林 (1996−), 男, 博士生, 主要研究方向为强化学习、智能体追逃博弈控制, E-mail: penglinhu@mail.nwpu.edu.cn;

潘泉 (1961−), 男, 教授, 博士, 主要研究方向为无人机信息安全、多源信息融合, E-mail: quanpan@nwpu.edu.cn;

赵春晖 (1973−), 男, 教授, 博士, 主要研究方向为无人机导航、无人机鲁棒控制, E-mail: zhaochunhui@nwpu.edu.cn.