计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications [doi: 10.15888/j.cnki.csa.009675]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# 基于深度强化学习的四旋翼无人机抗扰控制①

徐博洋, 时宏伟

(四川大学 计算机学院, 成都 610065)
通信作者: 时宏伟, E-mail: shihw001@126.com

**摘　要**: 随着无人机应用需求不断拓展, 为了保证无人机能够按要求完成预定任务, 抗干扰控制器的设计受到了诸多关注. 目前广泛使用的传统控制算法稳定性较好但抗干扰能力较差. 针对上述问题, 提出了一种基于改进双延迟深度确定性策略梯度 (TD3) 算法的混合抗干扰控制器, 该方法使用非线性模型预测控制 (NMPC) 作为基础控制器, 并引入了一个基于改进 TD3 的干扰补偿器进行混合控制. 该方法结合了 NMPC 控制器的优点的同时解决了传统控制算法在抗干扰方面的不足. 本文将多头注意力机制 (MA) 以及长短期记忆网络 (LSTM) 引入 TD3 的 Actor 网络中, 提高了 TD3 对于空间管理信息以及时间关联信息的捕捉能力, 同时引入一种连续型对数奖励函数来提高训练稳定性和收敛速度, 并使用带随机干扰的随机任务场景进行训练以提高模型泛化性. 在实验中将 NMPC-MALSTM-TD3 架构与使用 DDPG、SAC、TD3、PPO 算法作为干扰补偿器的架构进行对比, 实验结果表明, NMPC-MALSTM-TD3 架构的综合表现最好, 而且对 NMPC 的稳定性和实时性影响较小.

**关键词**: 深度强化学习; 非线性模型预测控制; TD3; 多头注意力; LSTM

引用格式: 徐博洋,时宏伟.基于深度强化学习的四旋翼无人机抗扰控制.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/9675.html

## Disturbance Rejection Control of Quadrotor UAVs Based on Deep Reinforcement Learning

XU Bo-Yang, SHI Hong-Wei

(College of Computer Science, Sichuan University, Chengdu 610065, China)

**Abstract**: As the demand for unmanned aerial vehicle (UAV) applications continues to expand, the design of disturbance rejection controllers which aim to ensure that UAVs can complete designated tasks as required has received significant attention. Traditional control algorithms widely used currently exhibit good stability but poor disturbance rejection capability. To address this issue, a hybrid disturbance rejection controller based on an improved twin delayed deep deterministic policy gradient (TD3) algorithm is proposed. This method utilizes nonlinear model predictive control (NMPC) as the base controller and introduces a disturbance compensator based on improved TD3 for hybrid control. This approach combines the advantages of the NMPC controller as well as addresses the shortcomings in disturbance rejection of traditional control algorithms. This study introduces a multi-head attention (MA) mechanism and long short-term memory (LSTM) network into the Actor network of TD3, enhancing TD3's ability to capture spatial management information and temporal correlation information. Additionally, a continuous logarithmic reward function is introduced to improve training stability and convergence speed, and training is conducted using random task scenarios with random disturbances to enhance model generalization. In experiments, the NMPC-MALSTM-TD3 architecture is compared with architectures using DDPG, SAC, TD3, and PPO algorithms as disturbance compensators. Experimental results demonstrate that the NMPC-MALSTM-TD3 architecture exhibits the most excellent disturbance rejection capabilities and

① 收稿时间: 2024-04-28; 修改时间: 2024-05-20; 采用时间: 2024-05-31; csa 在线出版时间: 2024-09-24

a smaller influence on the stability and real-time performance of NMPC.

**Key words**: deep reinforcement learning; nonlinear model predictive control; TD3; multi-head attention; LSTM

多旋翼无人机以其轻便、体积小、机动性强、起降方便以及能够实现空中悬停和低空稳定飞行的优点, 已经被广泛应用于航拍摄影、物流、资源勘探等领域[1,2]. 其中, 四旋翼无人机作为一种最常见的多旋翼无人机, 由于其系统的非线性和强耦合特性, 设计一种可靠、稳定而高效的控制算法具有一定的挑战性.

随着无人机应用需求的不断扩展, 保证无人机能够完成预期飞行任务的重要性不言而喻. 为此, 越来越多的研究者正致力于开发鲁棒的无人机控制器. 目前, 已经有多种高效且可靠的控制算法被提出, 包括模型预测控制 (MPC)、自适应控制、比例-积分-微分 (PID) 控制和反步法等. 其中, 模型预测控制尤其适用于处理多变量、多约束的非线性系统控制问题, 将非线性系统引入模型预测控制就得到了非线性模型预测控制 (NMPC). 并且相比于其他算法, MPC 在控制中能直接考虑输入、输出和状态的约束, 并能通过使用预测模型及优化算法来预测和优化未来一段时间内的控制策略, 从而实现更精确和稳定的控制. 此外, 一些高性能计算包的出现让 MPC 的计算速度得到了极大的提升, 保证了其实时性. 传统控制算法已然较好地完成了无人机的控制任务, 但是在一些强干扰的场景中, 传统控制算法无法做出及时响应, 这就导致在某些情况下, 传统控制算法可能会极大地偏离任务轨迹导致任务出现差错.

强化学习作为机器学习中的核心领域之一, 已经被广泛应用在各种领域当中. 在无人机控制领域中, 其也被广泛作为无人机控制方法或抗干扰算法的一部分来使用, 文献[3]设计了一种基于风源视觉特征的迁移强化学习四旋翼无人机前馈补偿器来抵抗风扰, 文献[4]设计了一种基于深度强化学习的前馈补偿器来抵抗四旋翼无人机近距离交叉飞行时所产生的剪切气流造成的突然干扰, 文献[5]设计了一种基于强化学习且支持高动态控制的四旋翼无人机控制方法, 该方法对扰动的抵抗效果较传统控制算法更好, 文献[6]设计了一种带干扰探测和补偿的强化学习控制器. 上述文献中直接利用强化学习算法作为控制方案的不确定性较大, 在未知任务下的表现可能会出现较大偏差, 相比之下, 一些传统控制算法虽然抗干扰能力可能不及强化学习算法, 但在未知任务下的稳定性要优于强化学习算法. 另外, 上述文献中提出的强化学习抗干扰补偿器均存在一定的场景限制, 不能广泛地应用在不同的任务场景下.

鉴于此, 本文采用非线性模型预测控制方法作为基础控制器, 以保证每个场景任务下的控制稳定性, 同时引入一个深度强化学习干扰补偿器来抵消不同场景下的干扰, 从而保证在干扰场景下的稳定运行. 同时, 本文将多头注意力机制 (MA) 以及长短期记忆网络 (long short-term memory, LSTM) 引入 TD3 算法的 Actor 网络当中, 提出了一种基于 MALSTM 的改进 TD3 算法作为干扰补偿器, 提高了模型处理空间以及时间信息的能力. 此外, 本文还提出了一种带边界的可控尺度连续型对数奖励函数, 以提高模型的收敛速度和训练稳定性, 并利用带随机干扰的随机空间训练策略来保证模型的泛化性. 实验结果表明, 本文提出的改进模型能够有效地对不同场景下的随机干扰进行抵抗, 并且对传统控制算法的实时性和稳定性影响较小.

## 1 四旋翼无人机建模

四旋翼无人机是一个欠驱动、非线性、高耦合的系统, 其包含 $X$、$Y$、$Z$ 轴平移运动以及滚转 (roll)、俯仰 (pitch) 和偏航 (yaw) 这 3 个旋转运动, 共 6 个自由度[7], 但只能依靠其自身的 4 个螺旋桨转动产生的推力来进行位置和姿态控制, 建立四旋翼无人机动力学模型之前, 需要建立描述模型所需的无人机结构和坐标系, 具体结构如图 1 所示.

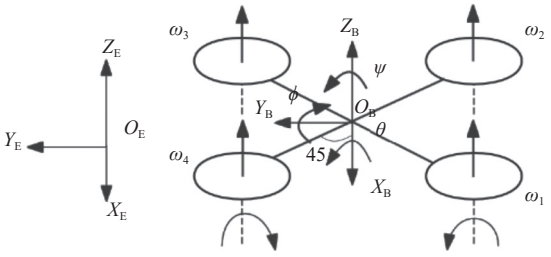


图 1 四旋翼无人机模型

图 1 中 $O_E$-$X_EY_EZ_E$ 代表惯性 (地球) 坐标系, $O_B$-$X_BY_BZ_B$ 代表机体坐标系, 机体坐标系与无人机臂成45°

夹角. 建立动力学模型时将四旋翼无人机视为一个刚体, 且假设其几何中心处位于其重心, 为了计算方便同时忽略空气阻力. 在建立模型以及后续姿态解算时采用四元数的方式进行姿态表示, 其避免了使用欧拉角表示时可能产生的万向锁等问题[8]. 四元数被定义为一个超复数, 其由 1 个标量$q_0$和 3 个矢量$q_{1:3}=\begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^{\mathrm{T}}$构成[9], 因此它可以被表示为:

$$q=\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^{\mathrm{T}} \tag{1}$$

四旋翼无人机 4 个螺旋桨推力差异会产生扭矩. 当需要改变姿态时, 通过改变不同螺旋桨的转速, 使各旋翼产生不同的推力, 从而产生扭矩, 改变无人机的姿态. 这里绕机体坐标系的扭矩$\tau_x$、$\tau_y$、$\tau_z$表示为:

$$\tau_x=l\cos\left(\frac{\pi}{4}\right)\cdot(u_2+u_3-u_1-u_4)F_m \tag{2}$$

$$\tau_y=l\cos\left(\frac{\pi}{4}\right)\cdot(u_3+u_4-u_1-u_2)F_m \tag{3}$$

$$\tau_z=c(u_2+u_4-u_1-u_3)F_m \tag{4}$$

其中, $l$表示螺旋桨电机中心到无人机重心的距离, $c$表示扭矩系数, $F_m$表示单螺旋桨电机的最大推力, $u_i\ (i=1,2,3,4)$表示每个螺旋桨电机的激活水平, $u_i\in[0,1]$, 其与最大推力的乘积为对应电机产生的推力$F_i=u_iF_m$.

利用 Newton-Euler 法建立四旋翼无人机的动力学方程时, 在表示速度动力学方程时会用到从机体坐标系到惯性坐标系下的正交旋转矩阵$R_B^E$, 此外, 在表示四元数动力方程时还会用到角速度矩阵$R_\omega$, 基于四元数姿态表示下的正交旋转矩阵以及角速度矩阵表示如下:

$$R_B^E=\begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_2q_3+2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix} \tag{5}$$

$$R_\omega=\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{6}$$

其中, $\omega_x$、$\omega_y$、$\omega_z$分别表示机体坐标系下无人机绕$x$、$y$、$z$轴的角速度. 基于上述公式, 利用 Newton-Euler 法建立的非线性四旋翼无人机动力学方程表示如式 (7) 所示.

$$\begin{cases} \dot{p}=v \\ \dot{q}=\dfrac{1}{2}R_\omega q \\ \dot{v}=R_B^Ea-\boldsymbol{g} \\ \dot{\omega}_x=\dfrac{1}{J_x}\left(\tau_x+(J_y-J_z)\omega_y\omega_z\right) \\ \dot{\omega}_y=\dfrac{1}{J_y}\left(\tau_y+(J_z-J_x)\omega_z\omega_x\right) \\ \dot{\omega}_z=\dfrac{1}{J_z}\left(\tau_z+(J_x-J_y)\omega_x\omega_y\right) \end{cases} \tag{7}$$

其中, $p=(x,y,z)^{\mathrm{T}}$表示无人机在惯性系下的位置, $v=\left(v_x,v_y,v_z\right)^{\mathrm{T}}$表示无人机在惯性系下的速度, $J_x,J_y,J_z$分别表示无人机绕机体系$x,y,z$轴的转动惯量. $a$为机体系下的无人机加速度向量, $\boldsymbol{g}$为地球坐标系下的重力加速度向量, 具体表示如式 (8)、式 (9) 所示, 其中$m$表示无人机质量, g表示重力加速度.

$$a=\begin{bmatrix} 0 & 0 & \dfrac{1}{m}\sum F_i \end{bmatrix}^{\mathrm{T}} \tag{8}$$

$$\boldsymbol{g}=\begin{bmatrix} 0 & 0 & \mathrm{g} \end{bmatrix}^{\mathrm{T}} \tag{9}$$

## 2 NMPC 控制器设计

NMPC 是一种先进的反馈控制策略, 它基于动态优化反馈控制策略. 该策略的整体性能主要取决于系统模型的准确性. NMPC 通过优化模型在有限时间域内的行为来计算当前最佳控制输入. 最优系统行为的预测来自开环在线优化, NMPC 将其形式化为受约束的有限范围内最优控制问题 (OCP)[10].

### 2.1 离散时间非线性系统

在非线性模型预测控制中需要建立离散时间非线性系统模型, 一般的非线性状态空间方程可以表示为:

$$\begin{cases} \dot{X}=f(X(t),U(t)) \\ Y(t)=X(t) \end{cases} \tag{10}$$

其中, $X$表示系统状态变量, $U$表示系统控制输入, $Y$表示系统输出变量. 在本文中系统状态$X$定义为 13 个相关物理量, 系统控制输入$U$定义为 4 个螺旋桨的激活程度, $\dot{X}=f(X(t),U(t))$定义为基于式 (7) 的状态空间方程, 其中系统状态$X$表示如下:

$$X=\begin{bmatrix} x & y & z & q_0 & q_1 & q_2 & q_3 \\ v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathrm{T}} \tag{11}$$

为了方便将上述非线性状态空间方程应用到计算

机当中进行计算, 需要将上述非线性状态空间方程转化为离散时间非线性系统模型, 设定采样时间$\Delta t$, $k+1$时刻的系统状态可以表示为:

$$\begin{cases} X_{k+1} = X_k + \Delta t \cdot f(X_k, U_k) \\ Y_k = X_k \end{cases} \tag{12}$$

以位置状态空间方程为例, 在采样时间$\Delta t$的条件下, $t$时刻对应下一采样时间的位置状态转移方程可以表示为:

$$\begin{cases} x(t+\Delta t) = x(t) + \Delta t \cdot v_x(t) \\ y(t+\Delta t) = y(t) + \Delta t \cdot v_y(t) \\ z(t+\Delta t) = z(t) + \Delta t \cdot v_z(t) \\ v_x(t+\Delta t) = v_x(t) + \Delta t \cdot \dot{v}(t) \\ v_y(t+\Delta t) = v_y(t) + \Delta t \cdot \dot{v}(t) \\ v_z(t+\Delta t) = v_z(t) + \Delta t \cdot \dot{v}(t) \end{cases} \tag{13}$$

### 2.2 最优控制问题

作为基础控制方法, 非线性模型预测控制仅以期望轨迹跟踪为目的而不涉及其他任务, 为此, 简化NMPC 的控制约束, 参考 MPC 标准形式, 以不带终端误差约束的系统状态约束以及控制输入界限约束为基础, 构建第$k$个采样时间下的最优控制问题以求得系统控制输入:

$$\begin{cases} \min \dfrac{1}{2} \displaystyle\sum_{i=1}^{N} |\tilde{Y}_{k+i} - \bar{Y}_{k+i}|^2_{Q_t} + |U_{k+i-1}|^2_{Q_r} \\ \text{s.t.} \begin{cases} X_{k+1} = X_k + \Delta t \cdot f(X_k, U_k) \\ Y_k = X_k \\ X_0 = X_{\text{init}} \\ u_{\min} \leqslant u \leqslant u_{\max} \end{cases} \end{cases} \tag{14}$$

其中, $\tilde{Y}_k$表示第$k$个采样时间下的预测状态, $\bar{Y}_k$表示第$k$个采样时间下的期望状态, $Q_t$表示系统状态误差加权矩阵, $Q_r$表示控制输入加权矩阵, $X_{\text{init}}$表示控制开始时刻初始系统状态, $u_{\max}, u_{\min}$分别表示控制输入的上下限, $N$表示预测时域长度.

为了满足实时性要求, 针对上述最优控制问题, 利用 CasADi[11]和 ACADOS[12]进行求解, 其主要思路是首先利用多重打靶法将最优控制问题转化为非线性二次规划问题, 再利用实时迭代 (real time iteration, RTI) 策略下的序列二次规划 (sequential quadratic programming, SQP) 进行求解[13]. 上述最优控制问题求解后得到控制量序列, 取第 1 个序列作为四旋翼无人机的控制量.

在 ACADOS 框架下, 为了高效求解 MPC 问题中的二次规划问题, 本文选择使用其提供的内点法求解器 HPIPM[14]进行求解, HPIPM 对于实时性要求较高的场景非常适用, 并且在应用 HPIPM 时采用全局缩减策略, 使 HPIPM 处理大规模问题时能够显著减少问题的维度, 从而提高求解速度; Hessian 矩阵的近似方法采用 Gauss-Newton 近似, 其简化了 Hessian 矩阵的计算, 减少了计算负担的同时也保持了优化问题的求解精度; 此外, 采用四阶龙格库塔方法对系统方程进行数值积分, 来模拟从当前时刻到未来某一时刻系统的状态变化.

## 3 基于改进 TD3 的深度强化学习补偿器

在非线性模型预测控制中, 当遇到强干扰时 NMPC 控制器无法及时作出反应, 为了对抗干扰的同时不影响 NMPC 控制器的实时性, 一种基于改进 TD3 的端到端控制补偿器被提出. 该补偿器基于当前场景的领航机状态以及被控机状态得到四旋翼无人机的基础控制量补偿$\Delta u_i$, 其本质是为了让被控机状态尽量贴近领航机状态从而达到抗干扰的效果, 这里的领航机状态集可以是领航机在该场景下任意一次平稳控制的预置状态集.

### 3.1 强化学习基础

一个标准强化学习问题可以被抽象为一个马尔可夫决策过程 (Markov decision process, MDP). 马尔可夫决策过程由元组$(S, A, P, r, \gamma)$组成, 其中$S$代表智能体的状态空间集合、$A$代表智能体能采取的动作集合、$\gamma$代表折扣因子、$r$代表一个奖励函数、$P$代表一个状态转移函数. $\gamma \in [0,1)$代表未来奖励占比, 决定了未来奖励对于当前决策的影响大小, $r(s,a)$表示在状态$s$下采取动作$a$所产生的奖励, $P(s' \mid s,a)$表示在状态$s$下采取动作$a$之后状态转移至$s'$的概率. 基于折扣因子, 定义马尔可夫决策过程中的回报$G_t$为从第$t$时刻的状态$S_t$到终止状态期间的所有折扣奖励之和, 表示为:

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots = \sum_{k=0} \gamma^k R_{t+k} \tag{15}$$

智能体的策略 (policy) 通常是一个函数$\pi(a \mid s)$, 其被定义为某一状态$s$下采取某一动作$a$的概率. 当一个策略是确定性策略 (deterministic policy) 时, 其在处于某个状态下时只会输出某一个确定性动作, 即在该状

态下采取该动作的概率为 1. 策略函数一般被表示为:

$$\pi(a \mid s) = P(A_t = a \mid S_t = s) \tag{16}$$

在强化学习中, 定义状态价值函数 (state-value function) 和动作价值函数 (action-value function) 来评估当前所处状态$s$的好坏以及处于某一状态$s$下采取某一动作$a$的好坏. 状态价值函数$V_\pi(s)$被定义为在状态$s$下遵循策略$\pi$所得到的期望回报, 动作价值函数$Q^\pi(s,a)$被定义为基于策略$\pi$时在状态$s$下执行动作$a$所得到的期望回报, 它们分别表示为:

$$V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] \tag{17}$$

$$Q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \tag{18}$$

强化学习的目的是找到期望累计奖励最大时的最优控制策略[15], 为了实现该目的, 在一些强化学习方法, 例如在采用 Actor-Critic (AC) 架构的算法中, 利用神经网络拟合一个 Critic 网络作为$Q$函数并利用神经网络拟合一个 Actor 网络作为策略函数$\pi$. 其中 Critic 提供当前策略下所采取动作的长期期望价值估计, 而 Actor 直接与环境交互并根据反馈调整策略. Actor 无法直接获得每个动作所产生的长期影响, 故 Critic 提供的期望价值估计可以指导 Actor 向更有利的动作选择方向更新策略, Actor 通过最大化$Q$值来更新策略, 从而间接地最大化长期回报.

### 3.2 双延迟深度确定性策略梯度 (TD3) 算法

深度确定性策略梯度 (deep deterministic policy gradient, DDPG)[16]算法是 DeepMind 于 2016 年提出的一种深度强化学习算法, 该算法结合了深度神经网络和确定性策略梯度方法, 可以被用于解决高维、连续动作空间下的控制问题, 但该算法学习过程中稳定性不佳且存在高估的问题. 在此基础上 Fujimoto 等人于 2018 年提出了双延迟深度确定性策略梯度算法 (twin delay DDPG, TD3)[17], 旨在改进 DDPG 算法中存在的高估等问题.

TD3 算法是一种 Actor-Critic (AC) 架构的强化学习算法, 其中包含了一个目标 Actor、两个目标 Critic 网络以及一个 Actor、两个 Critic 主网络, 引入目标网络是为了减少训练中的不稳定性, 相较于 DDPG 额外引入了一个 Critic 网络以解决高估的问题. TD3 首先从经验回放池中取出批次样本$(s_i, a_i, r_i, s_{i+1})$, 并利用目标 Actor 网络$\pi_{\phi'}$选取下一状态采取的动作$\tilde{a}_{i+1}$, TD3 额外加入了一个随机噪声平滑$\varepsilon$以减少价值函数的过度估计:

$$\tilde{a}_{i+1} = \pi_{\phi'}(s_{i+1}) \tag{19}$$

$$\tilde{a}_{i+1} = \tilde{a}_{i+1} + \varepsilon, \varepsilon \sim clip(N(0,\sigma), -c, c) \tag{20}$$

TD3 在基于时序差分法 (temporal difference, TD) 计算目标$Q$值时, 选取通过 Critic 网络计算得到的值中较小的一个进行计算, 以避免高估的问题, 并利用 MSE 损失基于梯度下降法对主 Critic 网络参数$\theta_j$ $(j = 1,2)$进行更新:

$$y = r_i + \gamma \min_{j=1,2} Q_{\theta'_j}(s_{i+1}, \tilde{a}_{i+1}) \tag{21}$$

$$\theta_j \leftarrow \arg\min_{\theta_j} N^{-1} \sum \left(y - Q_{\theta_j}(s_i, a_i)\right)^2 \tag{22}$$

为了进一步减少$Q$值的估计偏差, 同时保证学习的稳定性, TD3 在主 Critic 网络更新更新多次之后, 才更新一次主 Actor 和 Critic 网络. 其中 Actor 网络根据确定性策略梯度来延迟更新网络参数$\phi$:

$$\nabla_\phi J(\phi) = \frac{1}{N} \sum \nabla_a Q_{\theta_1}(s_i, a) \big| a = \pi_\phi(s_i) \nabla_\phi \pi_\phi(s_i) \tag{23}$$

Critic 网络基于一个目标平滑系数$\tau \in (0,1)$来延迟更新网络参数$\theta'_j$ $(j = 1,2)$:

$$\begin{cases} \theta'_j \leftarrow \tau\theta_j + (1-\tau)\theta'_j \\ \phi'_j \leftarrow \tau\phi_j + (1-\tau)\phi'_j \end{cases} \tag{24}$$

### 3.3 状态、动作空间

该问题下的状态$S$被定义为领航机状态$S_l$与被控机状态$S_c$的组合, 共包含 24 个特征值:

$$S = (S_l \quad S_c) \in R^{24} \tag{25}$$

另外, 在整个强化学习框架中, 四旋翼无人机的姿态表示被转换为基于欧拉角的表示, 状态$S$可以被具体表示为基于领航机的位置$p_l$、欧拉角$\Theta_l$、速度$v_l$、角速度$\Omega_l$以及被控机的位置$p_c$、欧拉角$\Theta_c$、速度$v_c$、角速度$\Omega_c$所组成的集合:

$$S = \{p_l, \Theta_l, v_l, \Omega_l, p_c, \Theta_c, v_c, \Omega_c\} \tag{26}$$

动作空间为螺旋桨电机激活程度的补偿量, 表示为:

$$a = [\Delta u_1, \Delta u_2, \Delta u_3, \Delta u_4] \tag{27}$$

其中, $\Delta u_i \in [-1/5, 1/5]$ $(i = 1,2,3,4)$表示 4 个电机的激活程度补偿量, 四旋翼无人机的最终控制量可表示为:

$$u_i = u_i + \Delta u_i \tag{28}$$

### 3.4 奖励设计

奖励设计在强化学习任务中起到至关重要的作用, 奖励设计的好坏直接决定了强化学习的训练结果. 为了提高模型的稳定性和收敛速度, 提出一种带边界的可控尺度连续型对数奖励函数, 该奖励函数可以设置每个物理量的奖励边界, 奖励边界可以将每个物理量的误差映射到统一的尺度上. 每项奖励$r_i\,(i=1,2,3,4)$以领航机与被控机之间的状态误差为基础建立, 共包含两机的位置、欧拉角、速度以及角速度误差, 它们对应的奖励边界分别为0.01、$\ln(\pi/12)$、0.05、$\ln(\pi/12)$, 对应的奖励尺度分别为10、4、3、2, 每项奖励具体表示为:

$$r_i = \begin{cases} -\lambda_i(\ln d_i - \ln b_i), & d_i > 0 \\ -\lambda_i(\ln 0.01 - \ln b_i), & d_i = 0 \end{cases} \tag{29}$$

其中, $d_i$表示领航机与被控机的物理量在某一时刻下的欧氏距离, $\lambda_i$表示对应物理量的奖励尺度, $b_i$表示对应物理量的奖励边界. 最终奖励表示为每项奖励的总和:

$$reward = -\sum_{i=1}^{4} \lambda_i(\ln d_i - \ln b_i) \tag{30}$$

### 3.5 训练场景设计

为了使模型能够得到更多的探索并增加其泛化能力, 训练时, 每局均采用带随机干扰的随机空间场景进行训练, 如图 2 所示. 另外, 每局领航机状态均使用前文所述的 NMPC 控制器在无干扰情况下进行控制后得到. 训练中, 完全随机干扰以 5% 的概率在训练过程中的每一时间刻随机出现一个固定时间, 固定时间长度为总时长的 1/16, 处于干扰阶段时不会触发下一次干扰, 且每一段干扰大小固定但不同, 随机干扰的生成方法表示为$u_{\rm k} = u_k + \varepsilon_k, \varepsilon_k \sim U(m_1, m_2)$, 其中$u_k \in (0,1)$ $(k=1,2,3,4)$代表四旋翼无人机对应螺旋桨的激活程度, $m_1$和$m_2$代表所服从均匀分布的上下限.

另外, 每局采用随机参考轨迹策略时, 起始点在一定范围内使用均匀分布抽样得到, 且每一时刻轨迹点按照某种方式随机生成, 每一时间刻的参考轨迹点生成方法表示为:

$$\Delta s = \frac{l_s}{M} \tag{31}$$

$$\begin{cases} x_{i+1} = x_i + \Delta s \cdot \sin(\beta_i) \cdot \cos(\beta_i'), & i > 0 \\ y_{i+1} = y_i + \Delta s \cdot \sin(\beta_i) \cdot \sin(\beta_i'), & i > 0 \\ z_{i+1} = z_i + \Delta s \cdot \cos(\beta_i), & i > 0 \\ x_0 \sim U(x_{\min}, x_{\max}) \\ y_0 \sim U(y_{\min}, y_{\max}) \\ z_0 \sim U(z_{\min}, z_{\max}) \end{cases} \tag{32}$$

其中, $\beta$代表极角, 用来控制关于$z$轴的倾斜程度, $\beta'$代表方位角, 用来控制$xoy$平面的偏转程度, $\Delta s$代表单位时间下的行走距离, $l_s$代表路径总长度, $M$代表单局训练的总步数. 另外, 对于每一时间刻, $\beta_i$和$\beta_i'$满足下面的均匀分布, 其中$n$代表随机程度:

$$\begin{cases} \beta_i' \sim U(-n, n), & i \neq 0 \\ \beta_i \sim U(-n, n), & i \neq 0 \\ \beta_0' \sim U(0, 2\pi), & i = 0 \\ \beta_0 \sim U(0, \pi), & i = 0 \end{cases} \tag{33}$$

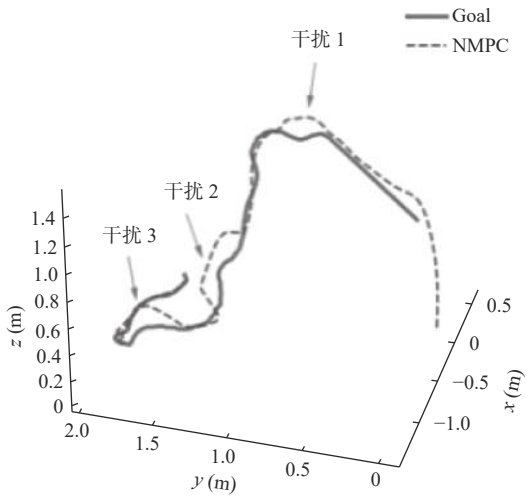


图 2 训练任务场景示例

### 3.6 MALSTM-TD3 补偿器

TD3 的 Actor 网络输入中包含领航机和被控机的位置、欧拉角、速度以及角速度, 为了使模型能够更加充分地学习不同物理量之间以及领航机与被控机之间的关系, 引入多头注意力机制 (multi-head attention, MA) 进行处理. 多头注意力机制能够处理多个序列, 从而发掘不同序列之间的相互关系. 此外, 它还能关注每个序列中不同位置的信息, 进而揭示序列内部的结构性关联. 多头注意力机制通过将注意力分解为多个“头”, 使得每个头可以独立学习序列的不同特征表达, 这种分解方式允许模型在处理信息时获得更为丰富的上下文信息, 从而改善对信息的整体理解. 另外, 在无

人机控制过程中, 存在显著的时间关联性, 为了有效捕获这种时间依赖性, 引入长短期记忆网络 (long short-term memory, LSTM), LSTM 是一种专门用于处理和记忆时间序列数据中的长期依赖关系的循环神经网络.

对于每一时间刻下的状态$S_t$, 在进行多头注意力机制处理前, 需要将原始输入$S=(S_l \quad S_c)\in R^{24}$按领航机与被控机的状态分组为位置$p=(p_l \quad p_c)$、欧拉角$\Theta=(\Theta_l \quad \Theta_c)$、速度$v=(v_l \quad v_c)$以及角速度$\Omega=(\Omega_l \quad \Omega_c)$这 4 个 1×6 的输入向量, 并将它们分别输入不同的全连接层映射到高维特征空间. 之后, 将 4 个全连接层的 1×128 输出按照不同的特征序列组合为 1×4×128 的输入向量输入到多头注意力机制中, 并利用多头注意力机制提取相关空间特征, 本文所使用的的多头注意力机制包含 4 个注意力头. 随后, 将多头注意力产生的 1×4×128 输出向量输入 LSTM 中, 利用 LSTM 处理 4 种特征的前后关联时间序列信息, 进一步优化对动态环境的理解和预测能力. 最后, 将通过 LSTM 得到的 1×256 的特征向量通过两层全连接层进一步提取特征后转换为网络输出. 多头注意力机制与 LSTM 进行融合, 能够让网络从输入状态中捕捉到训练前后时间刻中复杂的空间和时间关系. 引入多头注意力机制以及 LSTM 后的 Actor 网络架构如图 3 所示.

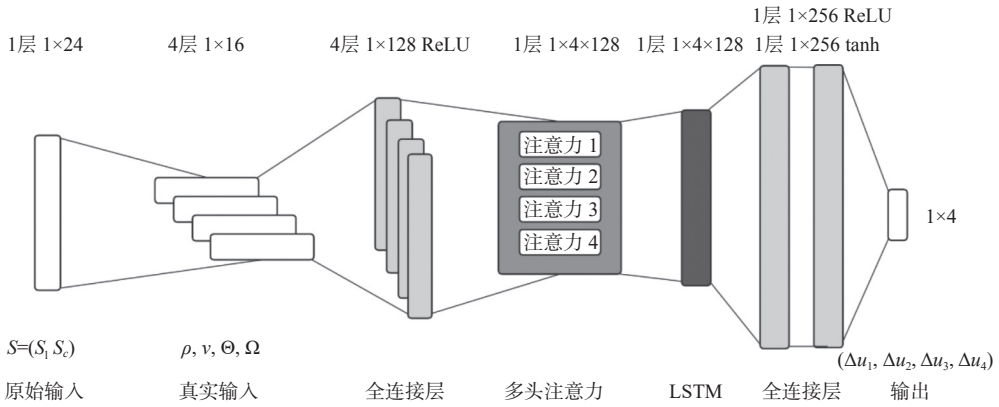


图 3 MALSTM-Actor 网络架构

### 3.7 整体架构

NMPC-MALSTM-TD3 控制器的整体架构如图 4 所示, 图中白色虚线框中的架构为引入 MALSTM 后的 TD3 算法单步训练过程, 详细原理见第 3.2 节所示, 其中策略函数与目标策略函数均被替换为 MALSTM-Actor 网络架构, 领航机$t+1$时刻的状态被用于生成一个经验回放池的 Transition. 图中灰色虚线框内为具体应用架构, 其中包含一个基础 NMPC 控制器、一个训练得到的稳定 MALSTM 策略网络以及一个基于当前场景的预置领航机状态集. $p_{\text{ref}}$表示任务参考轨迹集合, 其大小与预测控制时域长度相同. 另外, 需要注意的是, 在应用架构中, 领航机状态集仅提供$t$时刻的状态$s_l$.在实际应用中, 非线性模型预测控制器根据参考轨迹集合得到一个基础控制量$u$, 然后再利用 MALSTM-TD3 补偿器根据当前时刻的被控机与领航机状态生成一个补偿量$a$, 最终两者求和可得到四旋翼无人机的最终控制量$u$.

## 4 仿真实验与分析

### 4.1 仿真实验配置

本文利用不同的强化学习方法在相同场景下进行了训练, 包括 TD3、DDPG、SAC、PPO, 以及本文提出的 MALSTM-TD3, 对比了它们的效果和差异, 并在不同场景下测试了基于 TD3、DDPG、SAC、MALSTM-TD3 的模型稳定性以及抗干扰效果, 同时测试了引入 MALSTM-TD3 给系统带来的实时性影响. 本文所采用的仿真平台基于 Windows WSL2 创建的 Ubuntu 22.04.4LTS 系统下的 3.8 版本 Python 环境, CUDA 版本为 12.4, PyTorch 版本为 12.1, acados 计算平台版本为 0.3.0, CasADi 版本 3.6.4; 硬件运行环境采用 AMD 平台 R7-5800H 处理器, 32GB 内存, NVIDIA RTX3070 Laptop GPU. 仿真实验中 NMPC 的采样时间为 0.05 s, 仿真时间为 10 s, 预测时域长度为 10, 系统状态误差权重矩阵$Q_t$和控制输入权重矩阵$Q_r$表示为:

$$Q_t=\text{diag}\left(\left[\begin{array}{cccccccc} 10 & 10 & 10 & 0.1 & 0.1 & 0.1 & 0.1 & \\ 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & 0.05 & \end{array}\right]\right) \tag{34}$$

$$Q_r = \mathrm{diag}\left(\begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}\right) \tag{35}$$

另外, 仿真所使用的强化学习模型参数基本相同, 除 MALSTM-TD3 以外的模型架构均相同且处于训练任务下的较优水平, 经过实验测试采用 4 层或以上的全连接层效果较差, 对于该问题采用 3 层 256 维全连接层网络架构为较优架构. 仿真实验中四旋翼无人机参数、MALSTM-TD3 算法训练参数以及训练场景相关参数分别如表 1、表 2、表 3 所示.

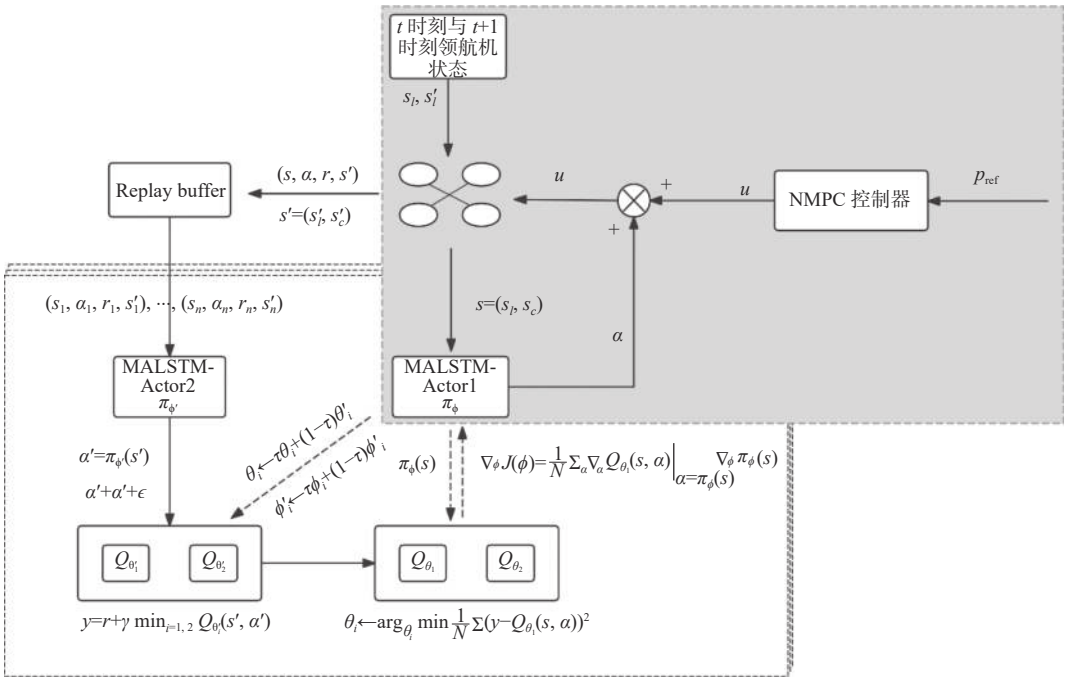


图 4　整体架构图

表 1　四旋翼无人机参数

| 参数 | 值 | 单位 | 参数 | 值 | 单位 |
|---|---|---|---|---|---|
| $m$ | 1.0 | kg | $F_m$ | 20 | N |
| $J_x$ | 0.03 | $\mathrm{kg\cdot m^2}$ | $l$ | 0.235 | m |
| $J_y$ | 0.03 | $\mathrm{kg\cdot m^2}$ | $c$ | 0.013 | m |
| $J_z$ | 0.06 | $\mathrm{kg\cdot m^2}$ | g | 9.81 | $\mathrm{m\cdot s^{-2}}$ |

表 2　MALSTM-TD3 算法训练参数

| 参数 | 值 | 参数 | 值 |
|---|---|---|---|
| 折扣因子 | 0.99 | 目标策略噪声标准差 | 0.15 |
| 通用学习率 | 0.001 | 噪声剪裁阈值 | 0.35 |
| 经验回放池容量 | 1 000 000 | 目标平滑系数 | 0.005 |
| 小批量大小 | 128 | 最大训练局数 | 3 200 |
| 探索噪声 | 0.1 | 最大单局步数 | 200 |

表 3　训练场景参数

| 参数 | 值 | 参数 | 值 |
|---|---|---|---|
| $m_1$ | 0.06 | $y_{min}$ | 0.2 |
| $m_2$ | 0.12 | $y_{max}$ | 0.5 |
| $l_s$ | 5 | $z_{min}$ | 0.25 |
| $x_{min}$ | 0.2 | $z_{max}$ | 1.0 |
| $x_{max}$ | 0.5 | $n$ | 0.4 |

### 4.2　仿真实验结果与分析

本文以不同强化学习方法作为补偿器进行训练, 并对比了它们的奖励趋势初步分析不同模型的优劣, 训练过程中单局奖励为该回合每一采样时间下的奖励总和, 训练得到的奖励如图 5 所示, 图中阴影部分代表真实奖励变化趋势, 实线代表真实奖励进行移动平均后的平均奖励变化趋势. 相比于 TD3 和 MALSTM-TD3, DDPG、SAC 和 PPO 算法在该场景下的表现较差, DDPG 在 600 局左右奖励出现了大幅下降, SAC 整体训练稳定在 3 000 局左右收敛到一个比较差的水平. TD3 于 2 300 局左右收敛, MALSTM-TD3 于 1900 局左右收敛, 相较于 TD3, MALSTM-TD3 奖励数值整体高出 TD3 约 500. MALSTM-TD3 相比 TD3 收敛速度有一定的提高、训练过程更加平稳, TD3 与 MALSTM-TD3 较其他算法的表现更加优秀, 整体收敛于一个较高的水平.

由于 PPO 算法的效果不佳, 故仅利用上述训练得到的 DDPG、TD3、SAC 以及 MALSTM-TD3 模型作

为补偿器分别在一个六边形场景下进行测试, 且 DDPG 采用训练 600 局时的参数. 在仿真中, 于第 7 s 左右加入一个$x$轴正上方向的强干扰, 来测试不同模型的抗干扰能力. 包括单 NMPC 控制在内的模型在该场景下的表现如图 6 所示, 不同算法下的控制状态曲线如图 7–图 12 所示.

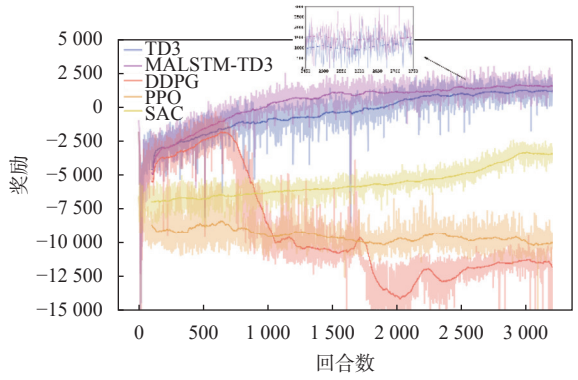


图 5 每局奖励变化

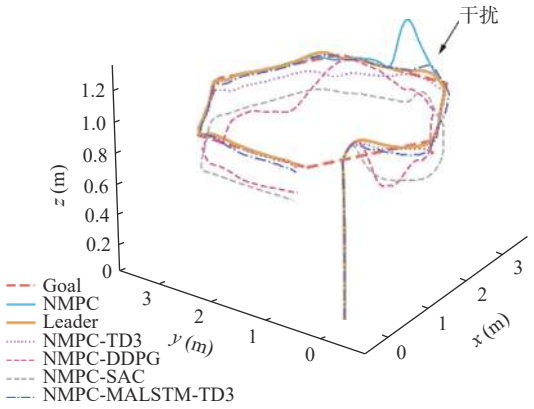


图 6 六边形轨迹任务表现

基于图 6 分析, 单 NMPC 控制器下的无人机状态在干扰处均处产生了较大的偏差. 在基于不同补偿器的控制算法下的位置控制表现中, 不同算法下的四旋翼无人机的$x$轴方向的位置表现如图 7 所示, 不同算法的表现均比较稳定且在干扰处可以得到较好的抵抗效果, 但是 DDPG 在非干扰区域会产生一定的偏差; $y$轴方向位置表现如图 8 所示, 其没有受到干扰的影响故不存在显著差距, 但 DDPG 仍然存在一些偏差; $z$轴方向上的位置表现如图 9 所示, 领航机在原始控制中产生了一些波动, DDPG 与 SAC 在全程上都相较领航机产生了较大的偏离, TD3 在第 8–16 s 中产生了比较大的偏离, 相较之下 MALSTM-TD3 能够全程都比较稳定的贴合领航机状态飞行, 且在干扰处的抵抗效果较好.

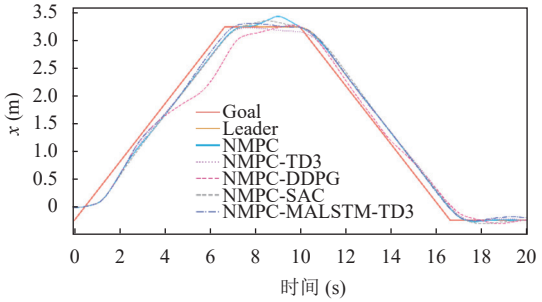


图 7 $x$轴位置变化曲线

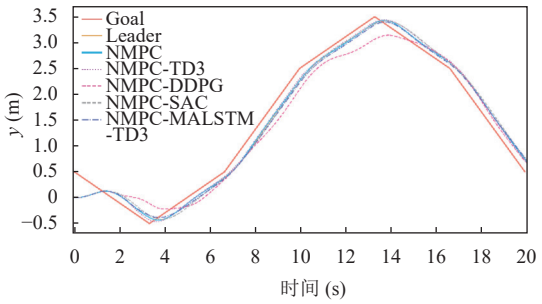


图 8 $y$轴位置变化曲线

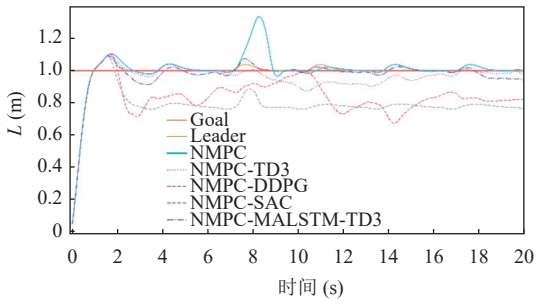


图 9 $z$轴位置变化曲线

在基于不同补偿器的控制算法下的姿态控制表现中, 无人机的滚转角姿态表现如图 10 所示, DDPG 在整体表现中出现了较大的偏移, SAC 在滚转角控制上的表现整体都较为稳定, TD3 在 10–13 s 间出现了明显的波动, MALSTM-TD3 在 3 s 左右出现了一个明显的抖动但整体状态更加稳定, TD3 与 MALSTM-TD3 在干扰处均有较好的抵抗效果; 无人机的俯仰角姿态表现如图 11 所示, 在俯仰角的控制表现上不同模型的效果与在滚转角姿态控制的表现较为类似; 无人机的偏航角控制表现如图 12 所示, 相比之下 MALSTM-TD3 虽也出现了偏差但较 DDPG 与 TD3 更好, 在此场景下

SAC 的表现最佳, 全程都能够稳定贴合领航机状态飞行.

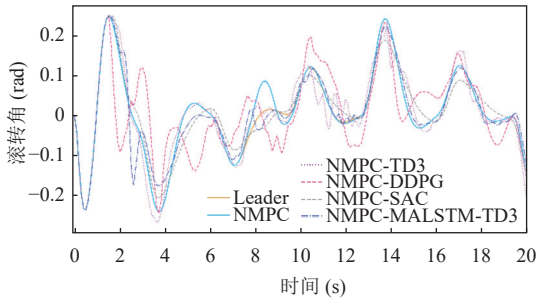


图 10 滚转角姿态变化曲线

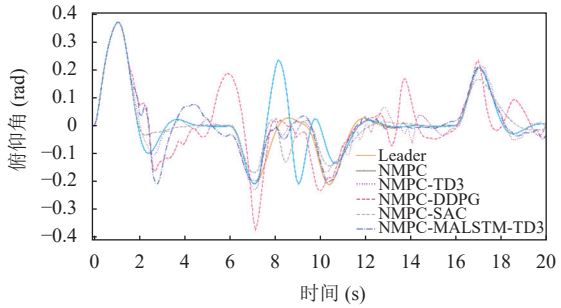


图 11 俯仰角姿态变化曲线

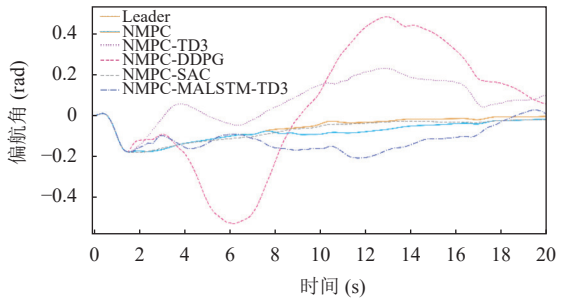


图 12 偏航角姿态变化曲线

另外, 基于不同算法的补偿器在该场景下与领航机的平均状态误差表现如表 4 所示, 可以看出在主要的位置跟踪任务中除了 DDPG 以外其他算法的$x$轴、$y$轴方向位置的状态差距并不大, 而在$z$轴方向的控制上 MALSTM-TD3 要远远优于其他算法. 在姿态控制表现上, 除 DDPG 以外其他算法在滚转角和俯仰角上的表现差距并不大, SAC 相较其他算法在姿态控制上的表现更好, MALSTM-TD3 相较 TD3 在偏航角的控制表现上优化了一倍. 在主要的位置控制任务上, MALSTM-TD3 相较其他算法的表现较好, 但是其在偏航角的控制上较 SAC 有较大差距, 综合来看, MALSTM-TD3 较其他算法的效果更优. 除此之外, 从表中也可以看出, SAC 在该任务上的潜力很大, 尽管在$z$轴方向的位置控制较差, 但在其他状态控制上效果较好.

表 4 不同算法在六边形轨迹任务下的平均状态误差

| 模型 | $x$ (m) | $y$ (m) | $z$ (m) | $\phi$ (rad) | $\theta$ (rad) | $\psi$ (rad) |
|---|---|---|---|---|---|---|
| TD3 | **0.3888** | **0.3753** | 0.6474 | 0.3402 | **0.3313** | 2.1942 |
| DDPG | 1.5957 | 1.8181 | 2.6195 | 0.9255 | 0.9144 | 3.8705 |
| SAC | 0.4064 | 0.4277 | 3.3927 | 0.3568 | 0.4031 | **0.1561** |
| MALSTM-TD3 | 0.4056 | 0.4006 | **0.3303** | **0.2892** | 0.4030 | 1.1201 |

为了确保模型能够被应用在不同的场景中, 将不同模型分别在圆形以及方形轨迹场景下进行测试, 并在其中某段施加一个相同的$x$轴正上方向的强干扰, 不同模型在圆形轨迹任务下的表现如图 13 所示, 在方形轨迹任务下的表现如图 14 所示. 可以看到在不同场景下, NMPC-MALSTM-TD3 控制器均可以达到比较好的抗干扰效果, 且稳定性较好, 不会出现较大波动.

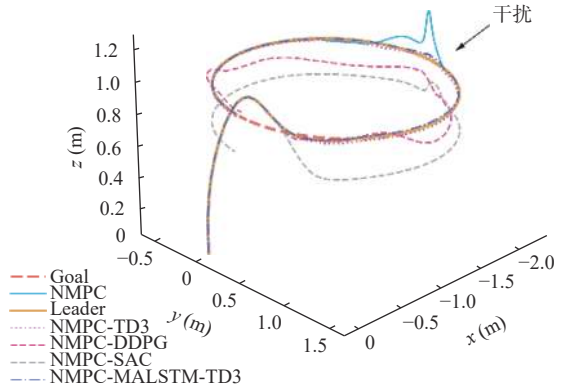


图 13 圆形轨迹任务表现

另外, 在基于 R7-5800H 的单核实验平台下进行实时性测试, 将任务全部放于 CPU 中进行处理, 分别利用 NMPC-MALSTM-TD3 与 NMPC 在六边形轨迹任务下分别进行 10 局控制任务并记录每局的程序运行时间后进行平均得到平均运行时间, 其中 NMPC 控制器的平均运行时间为 2.11 s, NMPC-MALSTM-TD3 控制器的平均运行时间为 2.29 s. 根据实验结果, 相比于仿真时长 10 s, 该方法能在 2 s 左右就完成控制决策, 且引入 MALSTM-TD3 补偿器以后的时间复杂度增加不超过 10%, 在本文的实验平台中具有较好的实时性. 然而主流的 PC 级计算平台并不适用于四旋翼无人机控制这样的移动计算任务, 需要将其与移动计算平台进行对比, 表 5 为本文的计算平台与近年来主流移动计算平台的单核计算能力的相对分数.

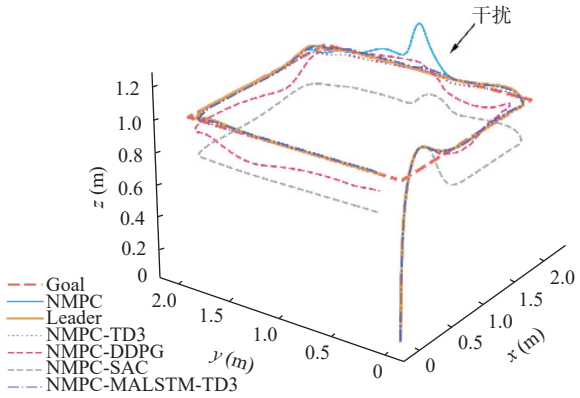


图 14 方形轨迹任务表现

表 5 不同计算平台单核相对测试分数

| 计算平台 | 相对测试分数 | 发布年份 |
|---|---|---|
| R7-5800H | 1 | 2021 |
| SM8250 | 0.800 | 2019 |
| SM8350 | 0.802 | 2020 |
| SM8450 | 0.797 | 2021 |
| SM8550 | 0.956 | 2022 |
| SM8650 | 0.993 | 2023 |

理论上, 只要程序运行时间不超过仿真时间, 就可以认为计算平台性能足以应对实际场景, 然而计算平台不仅仅承担控制计算, 可能还承担其他计算任务, 故这里以运行时间 5 s 作为计算性能允许的最小值, 那么以上面实验结果为例, 相对测试分数值至少为 0.458 才满足要求. 而根据表 5, 近几年主流移动计算平台的计算性能要远远高于该标准, 理论上本文提出的控制方法可以被应用于实际四旋翼无人机控制领域当中. 事实上, 移动计算平台的计算性能已经能够达到一些主流 PC 级计算平台的水平, 且算力出现了较大的冗余, 采用一些旧的移动计算平台不仅能够完成相关控制任务, 成本也能够得到控制.

## 5 结论与展望

本文针对 NMPC 控制器在抗干扰方面的不足进行了改进, 提出了一种基于深度强化学习的混合抗干扰控制器, 用于解决 NMPC 控制器在遇到干扰时产生的剧烈波动, 在此基础上对 TD3 算法进行了改进, 引入了 MALSTM-Actor 网络, 将多头注意力机制与 LSTM 进行融合以后加入 Actor 网络, 提升了模型对于空间和时间信息的表达能力, 并提出了一种带边界的可控尺度连续型对数奖励函数让模型训练稳定性得到一定提升, 同时采用了一种带随机干扰的随机场景训练策略来提升模型的泛化性. 仿真实验结果表明, 基于 MALSTM-TD3 的补偿器在综合表现上最佳, 特别是在 $z$ 轴位置的稳定性方面比较突出, 并且, 基于深度强化学习的端到端补偿器对原控制器的实时性影响较小可以轻易引入并应用在原控制器当中. 本文提出的补偿器效果较好, 理论上可以应用于其他传统控制算法当中, 该想法有待进一步研究验证. 与此同时, 该模型也存在一些不足, 例如应用模型前, 必须要使用领航机在任务场景中进行一次稳定控制, 并得到相应的状态集才可以使用该模型, 未来可以进一步研究无领航机下的抗干扰补偿器, 以使其能够得到更加广泛和灵活的应用.

## 参考文献

1 Chen LL, Liu ZB, Gao HG, *et al*. Robust adaptive recursive sliding mode attitude control for a quadrotor with unknown disturbances. ISA Transactions, 2022, 122: 114–125. [doi: 10.1016/j.isatra.2021.04.046]

2 Zou X, Liu ZB, Zhao W, *et al*. Optimal hovering control of a tail-sitter via model-free fast terminal slide mode controller and cuckoo search algorithm. Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS). Athens: IEEE, 2021. 978–984.

3 Song FL, Li Z, Yu XH. A feedforward quadrotor disturbance rejection method for visually identified gust sources based on transfer reinforcement learning. IEEE Transactions on Aerospace and Electronic Systems, 2023, 59(5): 6612–6623.

4 Song FL, Li Z, Yang SC, *et al*. Anti-disturbance compensation for quadrotor close crossing flight based on deep reinforcement learning. IEEE Transactions on Industrial Electronics, 2023, 70(3): 3013–3023. [doi: 10.1109/TIE.2022.3172764]

5 Li MJ, Cai ZH, Zhao J, *et al*. Disturbance rejection and high dynamic quadrotor control based on reinforcement learning and supervised learning. Neural Computing and Applications, 2022, 34(13): 11141–11161. [doi: 10.1007/s00521-022-07033-7]

6 Pi CH, Ye WY, Cheng S. Robust quadrotor control through reinforcement learning with disturbance compensation. Applied Sciences, 2021, 11(7): 3257. [doi: 10.3390/app11073257]

7 范文茹, 刘权威, 田栢苓. 基于干扰补偿的四旋翼无人机轨迹跟踪控制. 现代防御技术, 2024, 52(2): 87–93.

8 Islam M, Okasha M, Sulaeman E. A model predictive control

(MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking. International Journal of Control, Automation and Systems, 2019, 17(11): 2819–2832. [doi: 10.1007/s12555-018-0860-9]

9 Zhang TH, Kahn G, Levine S, *et al*. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA). Stockholm: IEEE, 2016. 528–535.

10 刘昊. 基于非线性模型预测控制的无人船路径跟踪方法研究 [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2022.

11 Andersson JAE, Gillis J, Horn G, *et al*. CasADi: A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation, 2019, 11(1): 1–36. [doi: 10.1007/s12532-018-0139-4]

12 Verschueren R, Frison G, Kouzoupis D, *et al*. acados—A modular open-source framework for fast embedded optimal control. Mathematical Programming Computation, 2022, 14(1): 147–183. [doi: 10.1007/s12532-021-00208-8]

13 柳子然, 戴梓健, 岳程斐, 等. 基于高斯混合过程的空间机器人任务空间预测控制方法. 系统工程与电子技术, 2023, 45(11): 3597–3605.

14 Frison G, Diehl M. HPIPM: A high-performance quadratic programming framework for model predictive control. IFAC-PapersOnLine, 2020, 53(2): 6563–6569. [doi: 10.1016/j.ifacol.2020.12.073]

15 梁吉, 王立松, 黄昱洲, 等. 基于深度强化学习的四旋翼无人机自主控制方法. 计算机科学, 2023, 50(11A): 220900257. [doi: 10.11896/jsjkx.220900257]

16 Lillicrap T P, Hunt J J, Pritzel A, *et al*. Continuous control with deep reinforcement learning. arXiv:1509.02971, 2015.

17 Fujimoto S, Hoof H, Meger D. Addressing function approximation error in Actor-Critic methods. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 1587–1596.

(校对责编: 张重毅)