https://doi.org/10.1051/jnwpu/20224051055

基于深度强化学习的端到端无人机避障决策

张云燕[1], 魏瑶[2], 刘昊[2], 杨尧[3]

(1.西北工业大学 电子信息学院, 陕西 西安 710072; 2.西北工业大学 航天学院, 陕西 西安 710072;
3.西北工业大学 无人系统技术研究院, 陕西 西安 710072)

摘 要:针对传统无人机避障算法需要构建离线三维地图以及速度控制不连续、速度方向选择受限的问题,基于深度确定性策略梯度(deep deterministic policy gradient,DDPG)的深度强化学习算法,对无人机连续型动作输出的端到端避障决策方法展开研究。建立了基于 DDPG 算法的端到端决策控制模型,该模型可以根据感知得到的连续状态信息输出连续控制变量即无人机避障动作;在 UE4+Airsim 的平台下进行了训练验证表明该模型可以实现端到端的无人机避障决策,与数据来源相同的三维向量场直方图(three dimensional vector field histogram,3DVFH)避障算法模型进行了对比分析,实验表明 DDPG 算法对无人机的避障轨迹有更好的优化效果。

关 键 词:无人机;避障;DDPG;强化学习
中图分类号:TP312 **文献标志码**:A **文章编号**:1000-2758(2022)05-1055-10

小型多旋翼无人机因为其体积小、动作灵活、适宜在复杂环境下完成机动,从而广泛应用于各种军用以及民用领域。科学技术的发展,也使得针对无人机的导航指导与控制技术得到了更大提升。然而,在陌生环境中实时进行自主导航却是极具挑战性的问题。在结构不能确定、光线条件多变的未知三维环境中,实现自主规避障碍物仍然是亟待解决的问题[1]。

传统的无人机避障算法[2-6]需要构建离线三维地图,在全局地图的基础上以障碍点为约束,采用路径搜索算法计算出最优路径。有的避障算法[7-8]虽然避免了繁杂的地图构建工作,但是需要手动调整大量的参数且机器人在避障的过程中不能利用避障经验进行自我迭代。随着机器学习的发展,研究人员将有监督学习引入无人机的避障之中,将避障看作是一个基于监督学习[9]的分类问题,但需要对每个样本标签进行标注,这样无疑费时费力。在无人机避障过程中,如何能够充分利用无人机三维空间信息,避免环境构建工作,简化算法模型,提升无人机自主避障能力以及无人机避障效率,最终实现无人机更加安全高效地到达指定目的地仍然是目前需要解决的问题。

随着机器学习的发展,强化学习[10]被科研人员应用于无人机的控制之中。强化学习主要通过与外界交互来优化自身的行为,它比传统机器学习更具优势:①训练前不需要对数据进行标注处理,能更有效地解决环境中存在的特殊情况;②可把整个系统作为一个整体,实现端到端输入输出,从而使其中的一些模块具有更强的鲁棒性;③强化学习与其他机器学习方法相比较更容易学习到一系列行为。因此,使用强化学习算法的优势在于不依赖于传统非机器学习所需要的离线地图,以及监督学习所需要的人工标注的数据集,通过深度学习模型学习输入数据和输出动作的映射关系,使智能体具备处理高维连续空间下的决策问题的能力,避免复杂离线地图构建工作。

基于此,本文以三维环境下无人机避障问题为研究对象,研究了在含有建筑、树木、车辆等静态障碍物的陌生环境下,通过实时感知周围环境信息实现端到端的避障决策控制。本文的主要贡献在于:根据无人机探测能力、场景复杂度和障碍物的几何特性等因素,设计了 3DVFH[11]算法,利用无人机的

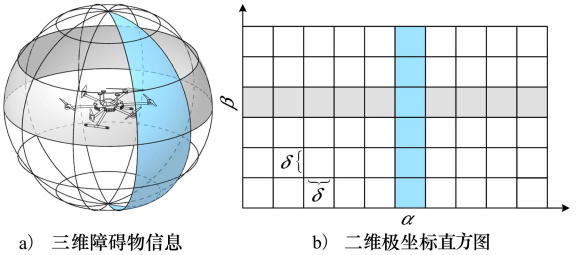
收稿日期:2021-12-14 **基金项目**:陕西省自然科学基金青年项目(2021JQ-075)与上海航天科技创新基金(SAST2020-070)资助
作者简介:张云燕(1982—),西北工业大学助理研究员,主要从事无人系统控制与决策研究。e-mail:yyzhangppwk@163.com

探测视场角来约束速度范围,在三维环境内直接进行无人机路径规划,避免三维空间信息的缺失问题;针对 3DVFH 算法中速度控制不连续和速度方向选择受限的问题设计了 DDPG[12] 网络处理连续状态空间并输出连续控制变量。其中状态空间仅为深度图像数据和位置信息,易迁移至实际场景中,并且根据深度图像数据和无人机通过窗口定义价值函数提升算法收敛率,避免算法产生冗余陷入局部最优;最后,分别在仿真环境及实际环境中对 2 组避障算法进行测试。测试结果表明 2 组避障算法均可实现避障功能,并且 DDPG 算法优化了飞行轨迹,缩短了飞行距离,提高飞行的安全性,具有一定的应用价值。

1 算法原理

1.1 基于 3DVFH 的无人机避障

3DVFH 算法由 Vanneste 等于 2014 年提出,该算法将二维向量场直方图应用于无人机三维飞行场景之中。算法假设无人机探测范围是以无人机中心为球心,直径为 d_u 的球面,以方位角 α 和俯仰角 β 作为横纵坐标,单位角 δ 为间隔,将球体内的障碍物信息转换为二值化的二维极坐标直方图,如图 1 所示,二值化结果用于判断区域能否通过。算法以与目标点夹角、最小避障角度以及与当前速度夹角作为权重,最终确定无人机避障方向。



a) 三维障碍物信息 b) 二维极坐标直方图

图 1 二维极坐标直方图转换

3DVFH 算法以栅格化的形式将无人机周围空间的障碍物信息进行降维处理,适合在复杂场景中应用,算法实现较为简单。但在实际应用中存在以下 2 个问题:

1) 根据 3DVFH 的原理和步骤可以看出,算法以建立在无人机中心的三维极坐标系为基础,输出下一步的运动方向,由当前运动姿态转换到该运动方向的线速度、角速度等这类动态信息均无法给出,速度连续性较差,会导致轨迹不够平滑,且难以进行轨迹优化。

2) 3DVFH 使用了障碍物栅格相对无人机中心的方位来直接限定运动方向,使用了障碍物栅格距无人机中心的距离计算权重,间接反映距离对运动的影响程度。但由于极坐标直方图的二值化过程受阈值影响极大,会损失完整的距离信息。这导致阈值难以调参,使得无人机难以接近靠近障碍物的位置,因为这些方向被提前排除。减小直方图通过窗口可以在一定程度缓解该问题,但同时也会降低无人机避障安全性。

1.2 基于 DDPG 的无人机避障

为了解决 1.1 节中 3DVFH 算法使得无人机速度连续性差、避障轨迹不平滑问题,引入基于无人机视觉输入的、具有连续动作输出的深度强化学习方法来优化无人机避障轨迹,提高避障安全性。无人机的避障决策问题可以认为是在连续动作空间中的控制问题。DDPG 采用了 DQN(deep q-network 深度 Q 网络)的思想和基于行动者-评论家[13](Actor-Critic)的算法结构,可以用来处理连续动作空间中的决策训练问题,其基于深度学习的部分使深度强化学习中的网络参数通过深度神经网络进行拟合与泛化,其策略梯度算法可以使智能体在连续的动作空间里依照学习结果选择合适的机动策略,并以确定性来避免策略梯度随机选择动作策略。

传统的 DQN 强化学习算法,尽管解决了高维的环境观察空间问题,却只能处理离散和低维的动作空间。因为它依赖于寻找使动作值函数最大化的动作,在连续值的情况下,需要在每一步进行迭代优化,从而导致离散动作数量爆炸。在连续的动作场景下,即需要智能体(无人机)的动作进行连续变化时,DDPG 采用 Actor-Critic 网络结构,并在 Actor 网络最后的输出层加入 tanh 激活层使得动作输出限制在[-1,1]之间,并根据需要进行缩放,保证输出为一个具体的浮点数,从而对智能体进行连续动作控制。在确定性策略的情况下没有概率的影响,当神经网络的参数固定下来的情况下,对于输入的同一个状态,必然只会输出同样的动作。

因此,本文将深度确定性策略梯度强化学习算法应用于无人机避障决策过程,以提高无人机动作连续性,平滑运动轨迹,实现避障决策优化。

1.2.1 DDPG 算法原理

本文中,将无人机的观察信息即当前时刻无人

机得到的深度图片以及无人机与目标之间的距离共同组合成为该时刻的状态信息 s_t 输入至神经网络。Actor 网络根据无人机此时状态向无人机提供相应的动作 a_t,无人机执行该动作与环境进行交互并根据此时得到的下一个状态 s' 获得奖励。无人机的目标便是通过学习最优策略 $\pi: S \rightarrow A$,来最大化累积折扣奖励 R_t。

$$R_t = \sum_{i=t}^{T} \gamma^{t-i} r_t \tag{1}$$

式中:r_t 为当前时刻无人机做出动作 a_t 与环境交互后得到的奖励值;γ 为折扣因子。

DDPG 采用具有连续动作空间的 Actor-Critic 网络结构,网络部分由 Actor 网络 $\mu(s \mid \theta^{\mu})$、Actor 网络对应的 Actor 目标网络 $\mu'(s \mid \theta^{\mu'})$、Critic 网络 $Q(s, a \mid \theta^{Q})$、Critic 网络对应的 Critic 目标网络 $Q'(s, a \mid \theta^{Q'})$,其中 $\theta^{\mu}, \theta^{\mu'}, \theta^{Q}, \theta^{Q'}$ 分别为对应网络的网络参数。

Critic 网络根据当前状态动作来逼近计算动作极值函数,其更新方式与 DQN 相似,通过最小化损失函数学习最优动作价值函数。如公式(2)所示,为 Critic 网络的损失函数计算方法

$$\begin{aligned} L(\theta) &= \mathrm{E}_{s,a,r}[(Q^{\wedge} - Q(s, a_t \mid \theta^{Q}))^2] \\ Q^{\wedge} &= r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'}) \end{aligned} \tag{2}$$

Actor 网络确定性地将状态 s_t 映射到动作 a_t,即基于当前环境状态给出无人机确定性动作策略。Actor 网络依赖于 Critic 网络所给出的 Q 值估计,通过梯度上升对其网络参数 $\theta^{\mu'}$ 进行更新。

$$\begin{aligned} &\nabla_{\theta^{\mu}} J \approx \mathrm{E}_s[\nabla_{\theta^{\mu}} Q(s, a \mid \theta^{Q}) \mid_{s=s_t, a=\mu(s_t \mid \theta^{\mu})}] = \\ &\mathrm{E}_s[\nabla_a Q(s, a \mid \theta^{Q}) \mid_{s=s_t, a=\mu(s_t)} \cdot \\ &g \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu}) \mid_{s=s_t}] \end{aligned} \tag{3}$$

为了避免自举算法过高估计价值函数,DDPG 采用软目标更新策略,利用 Actor 目标网络 $\mu'(s \mid \theta^{\mu'})$ 和 Critic 目标网络 $Q'(s, a \mid \theta^{Q'})$ 来计算目标价值。Actor 目标网络、Critic 目标网络的更新方式为如公式(4) 所示,其中 τ 为滑动平均系数,其取值通常小于 1。

$$\begin{cases} \theta^{\mu'} = \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \\ \theta^{Q'} = \tau\theta^{Q} + (1-\tau)\theta^{Q'} \end{cases} \tag{4}$$

1.2.2　网络结构设计

DDPG 所采用的 Actor-Critic 结构如图 2 所示,2 个网络的交互过程为:智能体感知得到所处环境的状态信息,Actor 网络根据此时的状态信息通过策略梯度计算出动作策略,智能体执行动作通过环境得到奖励以及下一个状态信息,该过程是与环境的一次完整交互过程。Critic 网络根据奖励值计算当前状态、下一状态的动作价值函数并根据 $\Delta_{\text{TD error}}$($\Delta_{\text{TD error}}$为 Critic 网络所估计当前状态 Q 值与目标 Q 址网线所估计下一状态 Q 值之差)进行更新,然后行动者网络沿着最大化 Critic 网络计算的 Q 值方向进行更新。该网络结构能够在单步更新参数的同时控制连续变量,并借鉴了 DQN 算法的经验回放(experience replay)机制,从经验回放池中随机采样,打乱状态之间的相关性,加快收敛速度,提高数据利用率。

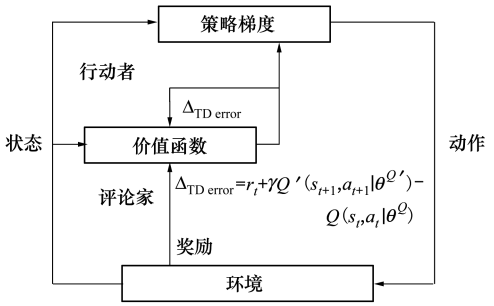


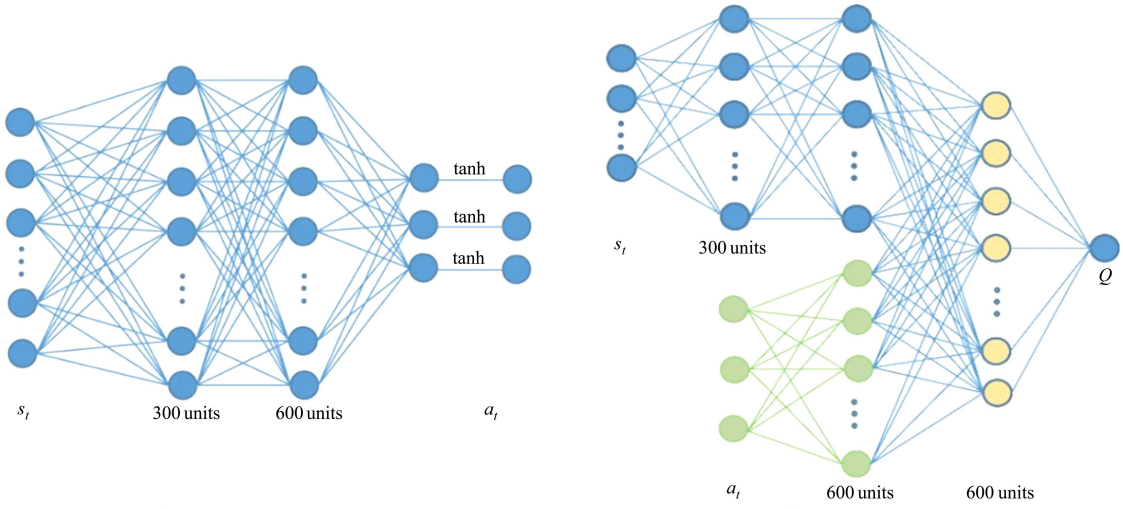
图 2　Actor-Critic 架构

本文采用的 Actor 和 Critic 网络具体结构如图 3 所示。Actor 网络通过输入此时无人机状态信息 ,从而给出无人机基于该状态的动作 。其中 Actor 网络由 2 个全连接层搭建形成,每层神经元个数分别为 300,600。各隐藏层间使用 ReLU 激活函数,输出层使用 tanh 激活函数规范无人机速度和旋转角速度,输出范围为(-1,1)从而保证输出的无人机动作为一个连续变化值。Actor 目标网络结构与此相同。

Critic 网络分为两部分,分别为对状态信息的处理以及对动作信息进行处理,输出为该时刻无人机的状态动作价值函数 $Q(s_t, a_t)$,用来评价无人机基于此时状态给出动作的好坏。$Q(s_t, a_t)$ 动作价值函数越高表明在在该状态下 Actor 网络所选取的动作越好。其中状态信息 s_t 经过 2 个全连接层后得到特征,与动作 a_t 通过一个隐藏层后得到的特征逐级相加,通过最后的全连接层输出 Q 值。

整个 DDPG 算法网络结构如图 4 所示,DDPG 算法包括 4 个网络,分别是 Actor 网络、Actor 目标网络、Critic 网络、Critic 目标网络。其中 Actor 具体网络结构、Critic 网络结构如图 3a)~3b)所示,Actor 目标网络、Critic 目标网络分别与 Actor 网络、Critic 网

络结构保持一致,但网络参数由超参数 τ 和 Actor 网络、Critic 网络参数共同实现更新。



a) Actor网络结构 b) Critic 网络结构

图 3 Actor-Critic 结构图

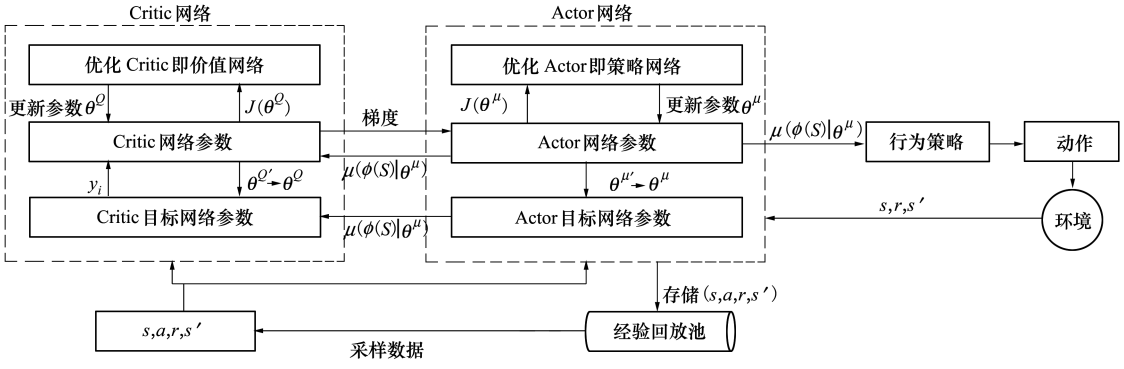


图 4 DDPG 算法结构图

· Actor 网络:基于目前的状态 s 选择合适的动作 a,更新策略网络参数 θ^μ,从而生成与环境相关的 (s',r);

· Actor 目标网络:基于从经验池中抽取的下一个状态 s',选择下一个合适的动作 a' 并保证其为最优,$\theta^{\mu'}$ 则每隔一段时间从 θ^μ 更新;

· Critic 网络:负责价值网络参数 θ^Q 的迭代更新,负责计算当前 Q 值 $Q(s,a\mid\theta^Q)$。对于第 i 次迭代,目标 Q 值 $Q^{\wedge}=r+rQ'(s',a'\mid\theta^{Q'})$;

· Critic 目标网络:负责计算目标 Q 值中的 $Q'(s',a'\mid\theta^{Q'})$ 部分。网络参数 $\theta^{Q'}$ 定期从 θ^Q 复制。

1.2.3 DDPG 算法设计

1) 图像预处理

本文选用深度摄像机成像分辨率为 640×480,FOV 约为 80°×60°,取特征向量大小为 16×16,去除图像边缘八像素宽度以定位图像中心,并将深度图像缩放处理,最终得到 39×29 分辨率的处理后图像(如图 5 所示),其单个像素对应 2°×2°的视场角。为保留距无人机最近的障碍物信息,每个像素点灰度取缩小前对应区域内的最小值。

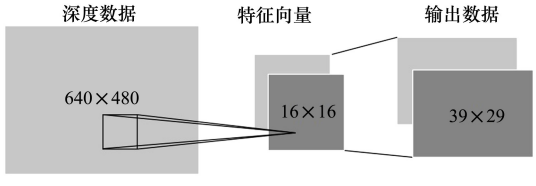


图 5 深度图像处理示意图

深度摄像机的有效探测范围为 10 m,取 7 m 为避障响应距离,该距离对应灰度值约为 178。图像

二值化过程与 3DVFH 算法一致,对于处理后图像,灰度值低于 178 的像素点 $H^b=1$,高于 178 的像素点 $H^b=0$。

对于通过窗口的确定,根据本文所选无人机尺寸和深度探测范围,选择距无人机中心 7 m,宽×高为 2 m×1 m 的区域作为通过窗口范围,根据视场角换算并缩小处理,得到像素大小约为 9×5 的区域,如图 6 所示。

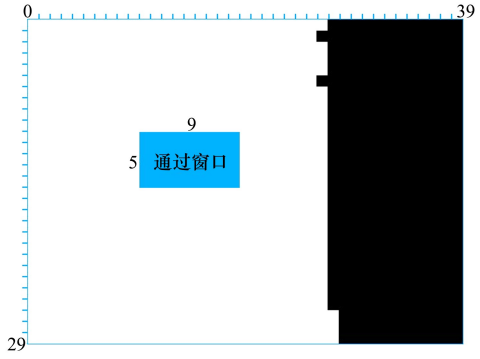


图 6　图像处理结果

2) 状态空间建模

状态空间 S 中包含了环境中智能体的属性和信息,并以此为依据来选取要执行的动作。为了实现无人机的避障,需要获取的信息分别为:无人机采集得到深度图像二值化结果 $H^b(i,j)$、与目标点的距离 $D_t\{\mathrm{d}x,\mathrm{d}y,\mathrm{d}z\}$ 和碰撞标记 S_{col},具体内容见表 1。

表 1　状态空间表

符号定义	状态描述	取值范围
H^b	深度二值化结果	0 或 1
D_t	无人机与目标点距离	由场景尺寸确定
S_{col}	无人机是否发生碰撞	0 或 1

3) 动作空间设计

DDPG 算法可以处理动作连续情况下的决策问题,此处选择无人机的 2 个状态:速度 v(m/s) 以及偏航角速度 ω_z(rad/s) 作为主要动作,以避免利用位置信息需要全局定位的问题,具体内容见表 2。在此,把速度和角速度进行映射处理,再进行计算,使它们的值更适用于神经网络。

表 2　动作空间表

符号定义	动作描述	取值范围
v_y	无人机 Oy_d 轴方向速度	(−0.58,0.58)
v_z	无人机 Oz_d 轴方向速度	(−0.45,0.45)
ω_z	无人机绕 Oz_d 轴旋转角速度	(−0.8,0.8)

4) 奖励函数设计

本文研究的目的是保证无人机在陌生环境中“存活时间”足够久,同时也要在飞行过程中尽量避开障碍物到达指定目标点,而不是原地不动,或者陷入局部最优的状态。这就意味着奖励函数的设计既要激励无人机从原点到达目标点,也要惩罚无人机产生碰撞。在本文中,奖励函数的设计参考了无人机的状态空间和动作空间。状态空间中,H^b 为深度二值化结果,当 H^b_Ω 均为 0 时,说明无人机速度方向上通过窗口无障碍物,定义该状态为 s_{free};反之,$\sum H^b_\Omega \neq 0$ 时,无人机速度方向上的通过窗口内有障碍物,定义该状态为 s_{block}。这样便确定了 H^b 和奖励 R 的关系。完整奖励函数如表 3 所示,当无人机的状态从 s_{free} 转移到 s_{block},表示无人机在飞行过程中遇见了障碍物,说明此时路径选择出现问题,则给予无人机一定程度的惩罚。反之,则给予无人机 + 3 的奖励。$s_{\mathrm{block}}\rightarrow s_{\mathrm{block}}$,$\sum H^b_\Omega\uparrow$ 表示当无人机遇见障碍物时发生状态转移后仍旧遇见障碍物,给予无人机 − 2 的惩罚,当远离障碍物时则给予激励。$D_t=0$ 表示无人机到达指定目标点,给予 + 100 奖励。S_{col} 为碰撞标志位,当 $S_{\mathrm{col}}=1$ 时表示无人机已经产生碰撞,应当对无人机进行惩罚。

表 3　奖励函数表

状态	奖励
$s_{\mathrm{free}}\rightarrow s_{\mathrm{block}}$	−3
$s_{\mathrm{block}}\rightarrow s_{\mathrm{free}}$	+3
$s_{\mathrm{block}}\rightarrow s_{\mathrm{block}}$, $\sum H^b_\Omega\uparrow$	−2
$s_{\mathrm{block}}\rightarrow s_{\mathrm{block}}$, $\sum H^b_\Omega\downarrow$	+2
$D_t=0$	+100
$S_{\mathrm{col}}=1$	−50

1.2.4　算法实现步骤

算法实现步骤如下:

输入:Actor 网络、Actor 目标网络、Critic 网络、

Critic 目标网络参数(θ^{μ},$\theta^{\mu'}$,θ^{Q},$\theta^{Q'}$),折扣因子 γ、软更新系数 τ、批量梯度下降的样本数 m、目标网络参数更新频率 C、最大迭代次数 T、随机噪声函数 N。

输出:最优 Actor 网络参数 θ^{μ},Critic 网络参数 θ^{Q}。

步骤 1　随机初始化 θ^{μ},$\theta^{Q'}=\theta^{Q}$,θ^{Q},$\theta^{\mu'}=\theta^{\mu}$。清空经验回放集合 $\boldsymbol{D}$;

步骤 2　对 i 从 1 到 T,循环执行以下步骤:

① 初始化 $\boldsymbol{S}$ 为当前状态序列的第一个状态,得到其特征向量 $\boldsymbol{\phi}(\boldsymbol{S})$。

② 在 Actor 网络基于状态 $\boldsymbol{S}$ 得到动作 $A=\mu(\boldsymbol{\phi}(\boldsymbol{S})\mid\theta^{\mu})+N$。

③ 执行动作 A,得到新状态 $\boldsymbol{S}'$,奖励 R,终止状态标志 is_ end。

④ 将五元组{$\boldsymbol{\phi}(\boldsymbol{S})$,$A$,$R$,$\boldsymbol{\phi}(\boldsymbol{S}')$,is_ end} 存入经验回放集合 $\boldsymbol{D}$。

⑤ 令 $\boldsymbol{S}=\boldsymbol{S}'$。

⑥ 从经验回放集合 $\boldsymbol{D}$ 中采样 m 个样本{$\boldsymbol{\phi}(\boldsymbol{S}_j)$,$A_j$,$R_j$,$\boldsymbol{\phi}(\boldsymbol{S}'_j)$,is_ end$_j$},$j=1,2,\cdots,m$ 计算目标 Q 值 $Q^{\wedge}$:

$$Q^{\wedge}=\begin{cases}R_j & \text{is\_ end}_j \text{ is true}\\ R_j+\gamma Q'(\boldsymbol{\phi}(\boldsymbol{S}'_j),\mu'(\boldsymbol{\phi}(\boldsymbol{S}'_j))\mid\theta^{Q'} & \text{is\_ end}_j \text{ is false}\end{cases}$$

⑦ 更新 Critic 网络参数 θ^{Q}。

⑧ 更新 Actor 网络参数 θ^{μ}。

⑨ 若 $T\%C=1$(表明此时 C 能够被 T 整除,网络按照该频率更新 Critic 目标网络和 Actor 目标网络参数),更新 Critic 目标网络和 Actor 目标网络参数。

⑩ 若 $\boldsymbol{S}'$ 为终止状态,本轮循环结束,否则转到步骤②继续训练。

表 4　参数配置表

参数	类型/值
Actor Optimizer	Adam,$\alpha=0.000\ 1$
Critic Optimizer	Adam,$\alpha=0.000\ 1$
γ	0.99
τ	0.001
D	500 000
m	300
T	9 000
N	Ornstein-Uhlenbeck

2　仿真及实验研究

2.1　仿真实验场景

微软公司开源了一个用于无人机模拟测试的高仿真软件 AirSim[14]。该软件依赖于虚幻引擎可以提供逼真的测试场景,并模拟其动力传感(如图 7 所示)。AirSim 支持开源和跨平台开发,因此本文选用 AirSim 作为仿真系统的基础,开展无人机避障算法研究。

图 7　AirSim 仿真效果图

仿真场景试验在仿真计算机中完成,计算机软件配置:Window10　CUDA10. 1, tensorflow2. 3. 0, Python3.7。

仿真场景的复杂度较高,俯视图方向能较为直观地展现无人机飞行轨迹(图 8 为本次仿真场景左透视图)。同时为保留障碍物高度信息,本文对仿真场景俯视图进行深度化处理,令障碍物高度对应障碍物投影的灰度值,灰度值越低,表示障碍物越高,效果如图 9b)所示。同时,为提升 DDPG 算法训练速度,将仿真场景约束为图 9a)所示的 100 m×100 m范围内,当无人机飞出该范围视为发生碰撞。

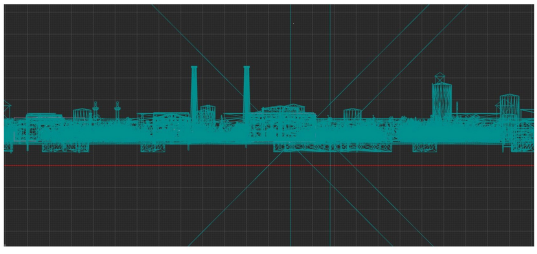
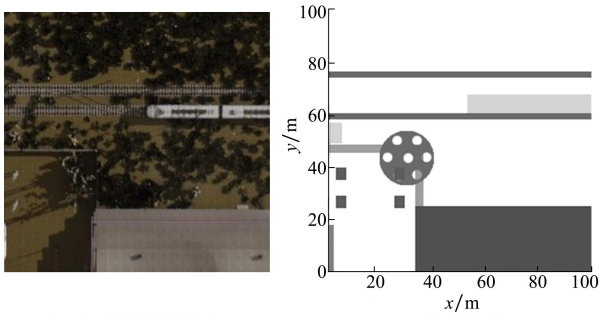


图 8　仿真场景左透视图

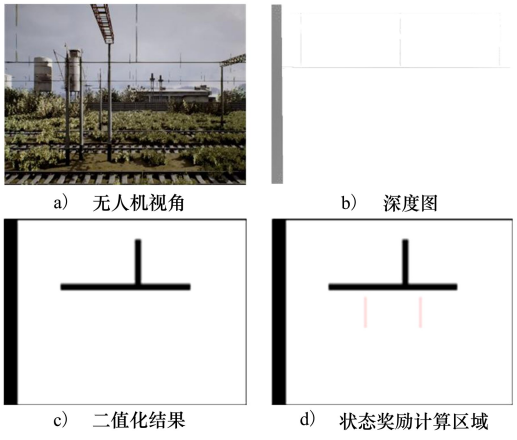


a)　约束区域俯视图　　b)　深度效果

图 9　俯视图处理效果

2.2　仿真场景测试

DDPG 的算法实验过程如图 10 所示,当无人机获取的深度图像信息为图 10b)时,经由算法处理得到的二值化极性直方图为图 10c),根据当前速度方向对应的通过窗口区域进行奖励函数区域计算。对于图 10c)的二值化结果,若当前速度指向正前方,则当前深度图像代表状态的奖励由图 10d)的红框区域确定,此时,无人机处于 s_{free} 状态。



a)　无人机视角　　b)　深度图

c)　二值化结果　　d)　状态奖励计算区域

图 10　DDPG 算法状态奖励计算过程

无人机训练过程如图 11 所示,其中 Cum reward 为每一训练回合的累积奖励值,Mean reward 为趋势线,表明每十个回合奖励值的平均值拟合曲线。本次训练共计 9 000 次,从图中可以看出随着训练次数增加,累积奖励和平均奖励都在逐渐增加,并且保持相对稳定。在前 2 000 次训练过程中,无人机处于探索状态因此其平均奖励均小于 0,但随着后期神经网络开始训练,其累积奖励在逐步提升并保持在 100 左右附近,表明此时无人机已经能够躲避障碍物并且安全到达指定地点。需要注意的是,可能在训练过程中出现无人机无法到达目标点的现象,由于无人机的探索率是随着训练次数递减的,因此,该现象是正常的,并不影响最终无人机躲避障碍物到达指定目标点的能力。并且由于探索阶段无人机只进行探索,其对应的神经网络并不进行梯度更新,此时网络输出的策略动作可能会在相邻的 2 个控制周期内出现突变。自由探索阶段是为了将更多的训练信息存入经验回放池中,从而保证在后续的网络训练阶段,神经网络可以更快学习得到好的策略。此时的动作突变并不会对无人机的训练产生影响。当神经网络开始训练时,Actor 网络会沿着使 Critic 网络给出更高 Q 值的方向进行梯度更新。随着网络的训练收敛,策略所输出动作在 2 个相邻的控制周期中出现突变的可能性会越来越小。在神经网络参数固定情况下,对于输入的同一个状态,必然只会输出同样的动作,在相邻的 2 个周期内所输出的动作便不会发生异常突变。

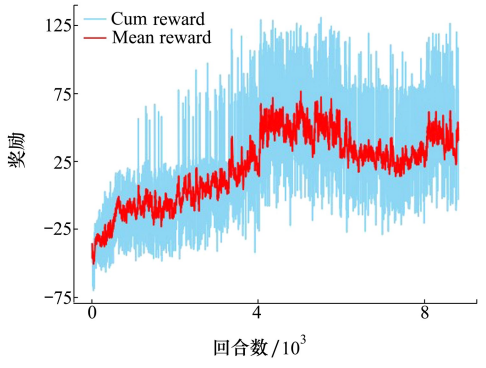


图 11　DDPG 算法训练过程

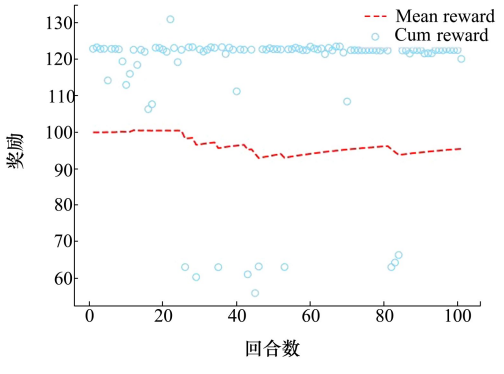
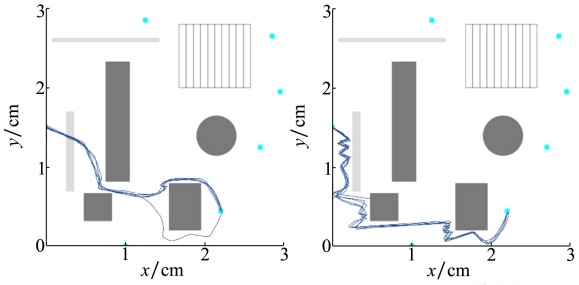


图 12　DDPG 算法测试结果

为更好地展现所提出的算法性能,首先将训练得到的算法在同样的环境下进行了 100 次测试,测试结果如图 12 所示。由于奖励函数设置为当无人机产生碰撞时奖励值为−50,因此当累积奖励大于 0 时表明无人机均可以躲避掉障碍物。当无人机的累

积奖励值大于 100 时表明此时无人机能够躲避掉所有障碍物并到达指定位置。综上所述,在 100 次测试结果中,有 90 次均为成功,无人机躲避障碍物成功率高达 90%,同时本次测试的平均奖励值为 95。

在此基础上,将 DDPG 算法在测试环境中无人机的运动轨迹与基于 3DVFH 的避障算法在同样的仿真场景中轨迹进行对比,2 种算法试验结果的俯视图如图 13 所示。2 种算法起始点与目标点位置设置相同,各试验 5 次,其中,图 13a)为 DDPG 算法 9 000 次迭代后效果。对比结果明显展现出 DDPG 算法飞行轨迹的平滑度高于 3DVFH 算法,这同时缩短了飞行距离,提高了避障效率。



a) DDPG 避障轨迹 b) 3DVFH 避障轨迹

图 13 仿真结果对比

2.3 真实场景测试

针对无人机避障需求,本文确定以“大疆 M600 pro 六旋翼无人机+RealSense D435i 深度摄像机+妙算 Manifold2 机载计算机”组合成完整硬件系统进行试验测试。其中试验无人机和深度摄像机如分别图 14~15 所示。

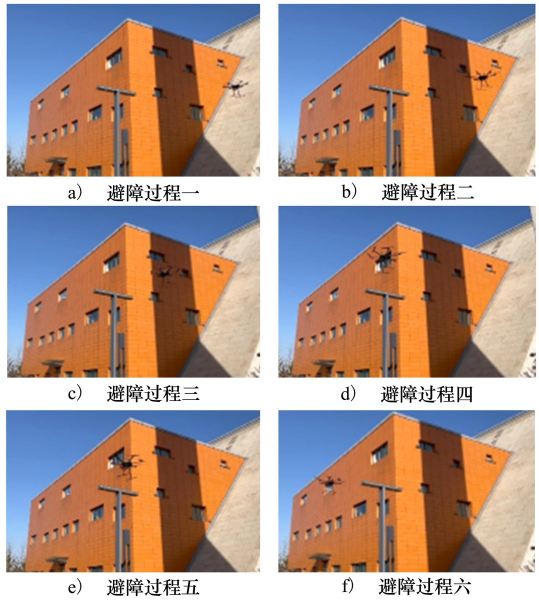
图 14 试验所用无人机

图 15 Intel D435i 深度摄像头

在真实场景试验前,为了保证测试效果,提高试验安全性,需完成以下几个步骤:首先,测试无人机飞控在 Onboard 模式下的坐标系及飞行速度误差,以修正算法的控制效果,提高控制精度,缩小控制效果与仿真场景的差异;其次,确定结合 RealSense D435i 深度摄像机实测时的试验保险措施。真实环境往往比仿真环境更为复杂,当传感器发生异常,输出错误的探测信息可能导致无人机发生撞击。应设置紧急悬停功能防止这类事故发生;最后,将避障系统与无人机平台相结合,观察算法在实际飞行中的过程,并进行适当调整,以达到预期效果。

2.3.1 3DVFH 算法实际验证

在实测环境中,无人机根据 3DVFH 算法实现避障的过程如图 16 所示。由于 3DVFH 算法对速度控制的不连续性,因此当避开障碍物后,速度的突变会导致无人机产生摆动,表现为图 16d)~16e)过程。从图 16a)~16f)为发现障碍物、调整前进方向以躲避障碍物并最终避开障碍物的部分过程。



a) 避障过程一 b) 避障过程二

c) 避障过程三 d) 避障过程四

e) 避障过程五 f) 避障过程六

图 16 3DVFH 算法实际避障过程

2.3.2 DDPG 算法实际验证

经过实测表明本文所设计算法其网络计算效率满足实时控制要求,避障算法生成避障控制指令的控制周期为 55 ms。图 17 为在实测环境中,无人机根据 DDPG 算法实现避障的过程。在 DDPG 算法中,无人机避障时所采取的避障策略均来自仿真环境的训练结果。在该算法中,可明显看出当发现障碍物时,无人机会放慢前进速度进行避障,当避开障碍物后继续高速向前飞行。图 17a)~17f)为发现障碍物、调整前进方向以躲避障碍物并最终避开障碍物的部分过程。

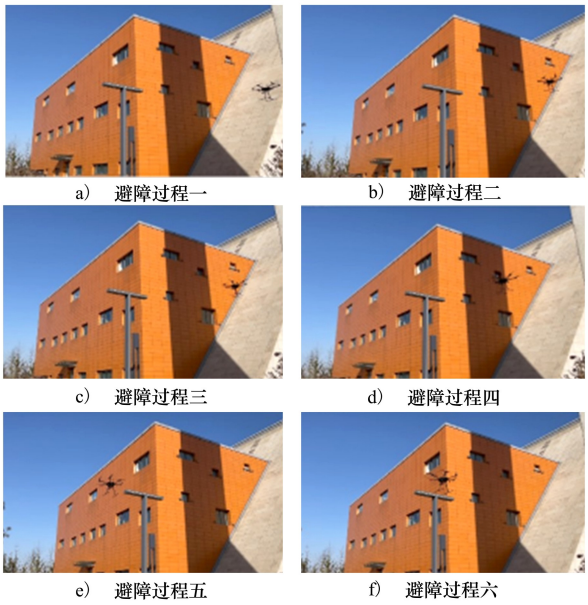


图 17　DDPG 算法实际避障过程

2.3.3　试验结果对比

为直观展现 2 种算法在实测中的效果,试验设置了高空无人机视角拍摄完整飞行过程,2 种算法的避障过程如图 18 所示。试验表明,网络的计算效率满足实际控制要求。其中,蓝色为 3DVFH 算法飞行轨迹,在避障结束段有明显摆动痕迹;红色为 DDPG 算法飞行轨迹,整体较为平滑。

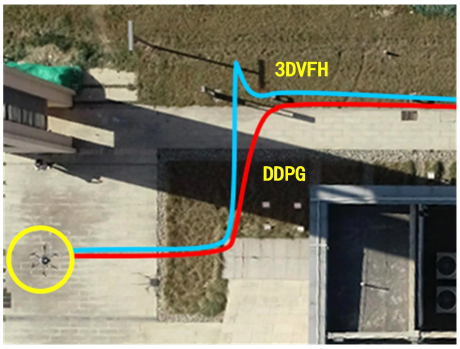


图 18　高空视角下的无人机避障过程

3　结　论

本文以无人机在未知三维环境中对静态障碍物的智能规避为研究背景,使用六旋翼无人机,在无全局地图信息的条件下,采用深度图像感知方案,对无人机避障硬件平台、无人机避障仿真系统、深度强化学习避障算法和传统无人机避障算法进行了研究。

本文根据无人机探测能力、场景复杂度和障碍物的几何特性等因素,结合无人机探测视场角约束速度范围设计了 3DVFH 算法,在三维环境直接进行无人机路径规划。针对 3DVFH 这类传统算法速度连续性差、避障轨迹不平滑问题本文引入深度强化学习中的 DDPG 算法,建立了基于 DDPG 算法的端到端决策控制模型。通过训练表明,在进行了 4 000 个episode 后算法收敛,无人机学习得到了避障能力。在此基础上,将训练收敛的 DDPG 算法模型在仿真环境中进行测试,测试表明无人机躲避障碍物成功率高达 90%。

分别将 3DVFH 算法和 DDPG 算法避障轨迹在仿真环境和实际环境中进行测试,实践表明 3DVFH 算法在避障结束阶段有明显摆动痕迹,DDPG 算法优化平滑了飞行轨迹,缩短了飞行距离,提高飞行的安全性,具有一定的应用价值。

参考文献:

[1]　WEI C, ZHANG F, YIN C, et al. Research on UAV intelligent obstacle avoidance technology during inspection of transmission line[C]//The 2015 International Conference on Applied Mechanics, Mechatronics and Intelligent Systems, 2015: 319-327

[2]　HWANGBO M, KUFFNER J, KANADE T. Efficient two-phase 3D motion planning for small fixed-wing uavs[C]//IEEE International Conference on Robotics and Automation, 2007: 10-14

[3]　HENG L, MEIER L, TANSKANEN P, et al, Autonomous obstacle avoidance and maneuvering on a vision-guided MAV using on-board processing[C]//IEEE International Conference on Robotics and Automation, 2011: 2472-2477

[4]　KUFFNER J, LAVALLE S. RRT-connect: an efficient approach to single query path planning[C]//IEEE International Conference on Robotics and Automation, 2000: 995-1001

[5]　DEITS R, TEDRAKE R. Efficient mixed-integer planning for UAVs in cluttered environments[C]//IEEE International Conference on Robotics and Automation, 2016: 42-49

[6]　SHIM D, CHUNG H, KIM H J, et al. Sastry, autonomous exploration in unknown urban environments for unmanned aerial vehicles[C]//AIAA Guidance, Navigation, and Control Conference and Exhibit, 2005: 1-8

[7]　杨坤山. 基于深顶学习的语义图像分割在三维重建系统中应用与实时化[D]. 成都:电子科技大学,2019

YANG Kunshan. Application and real-time of ivnage sematic segmentation based on deep lecrning in 3D rocomstmction system [D]. Chengdu: University of Electronic Science and Technolgey of China, 2019 (in Chinese)

[8] WAYDO S. Vehicle motion planning using stream functions[C]//IEEE International Conference on Robotics and Automation, 2003, 2: 2484-2491

[9] TAI L, LI S, LIU M. A deep-network solution towards model-less obstacle avoidance[C]//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016: 2759-2764

[10] Sutton R S, Barto A G. Reinforcement learning: An introduction[J]. Cambridge: MIT Press, 2018

[11] VANNESTE S, BELLEKENS B, WEYN M. 3DVFH+: real-time three-dimensional obstacle avoidance using an octomap[C]// Procedings of the 1st International Workshop on Model-Diwen Robot Scitwane Engineering Foundations, 2014

[12] KIM M S, HAN D K, PARK J H, et al. Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay[J]. Applied Sciences, 2020, 10(2): 575

[13] PETERS J, VIJAYAKUMAR S, SCHAAL S. Natural actor-critic[C]//European Conference on Machine Learining, Berlin, 2005: 280-291

[14] SRDJAN J, VICTOR G, ANDREW B. Airway anatomy of airsim high-fidelity simulator[J]. Anesthesiology, 2013, 118(1): 229-230

End-to-end UAV obstacle avoidance decision based on deep reinforcement learning

ZHANG Yunyan[1], WEI Yao[2], LIU Hao[2], YANG Yao[3]

(1.School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China;
2.School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China;
3.Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Aiming at the problem that the traditional UAV obstacle avoidance algorithm needs to build offline three-dimensional maps, discontinuous speed control and limited speed direction selection, we study the end-to-end obstacle avoidance decision method of UAV continuous action output based on DDPG(deep deterministic policy gradient) deep reinforcement learning algorithm. Firstly, an end-to-end decision control model based on DDPG algorithm is established. The model can output continuous control variables, namely UAV obstacle avoidance actions, according to the continuous state information perceived. Secondly, the training verification is carried out on the platform of UE4 + Airsim. The results show that the model can realize the end-to-end UAV obstacle avoidance decision. Finally, the 3DVFH(three dimensional vector field histogram) obstacle avoidance algorithm model with the same data source is compared and analyzed. The experiment shows that DDPG algorithm has better optimization effect on the obstacle avoidance trajectory of UAV.

Keywords: UAV; obstacle avoidance; deep deterministic policy gradient (DDPG); reinforcement learning

引用格式: *张云燕, 魏瑶, 刘昊, 等. 基于深度强化学习的端到端无人机避障决策[J]. 西北工业大学学报, 2022, 40(5): 1055-1064*

ZHANG Yunyan, WEI Yao, LIU Hao, et al. End-to-end UAV obstacle avoidance decision based on deep reinforcement learning[J]. Journal of Northwestern Polytechnical University, 2022, 40(5): 1055-1064 (in Chinese)