基于强化学习的扑翼飞行器路径规划算法

王思鹏, 杜昌平†, 郑 耀

(浙江大学 航空航天学院，杭州 310027)

摘 要: 针对扑翼飞行器机动性能弱的问题,提出一种在未知环境下示教学习辅助的强化学习局部路径规划算法(IL-PPO2). 首先,基于扑翼飞行器的受限视角的双目感知系统,提出一种心形避障算法,降低避障时对扑翼飞行器控制精度的要求,提高避障鲁棒性;其次,根据心形避障算法的特性,提出一种U型障碍的避障策略;最后,提出一种示教学习辅助的强化学习局部路径规划算法,将心形避障算法与局部路径规划算法相结合,实现扑翼飞行器的局部路径规划. 仿真结果表明:与TD3fD强化学习算法相比,IL-PPO2算法能够缩短模型训练时间,路径规划效率与成功率明显高于TD3fD算法;与动态窗口法(DWA)相比,IL-PPO2算法能够提高路径规划的成功率,并且有效融合心形算法,提高路径的平滑程度.

关键词: 扑翼飞行器；局部避障；U型障碍；专家系统；强化学习；路径平滑

中图分类号: TP242 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.1574 **开放科学(资源服务)标识码(OSID):**

引用格式: 王思鹏,杜昌平,郑耀. 基于强化学习的扑翼飞行器路径规划算法[J]. 控制与决策, 2022, 37(4): 851-860.

Local planner for flapping wing micro aerial vehicle based on deep reinforcement learning

WANG Si-peng, DU Chang-ping†, ZHENG Yao

(School of Aeronautics and Astronautics，Zhejiang University，Hangzhou 310027，China)

Abstract: For the poor maneuverability of flapping wing micro aerial vehicles (FWMAVs), a deep reinforcement learning (DRL) based local path planning method (IL-PPO2) is proposed with the assistant of demonstration learning in an unknown environment. Firstly, due to the limited visual angle of a stereo camera on a FWMAV, a “Heart” algorithm is proposed to reduce the requirement for control accuracy and meanwhile improve robustness. Then, according to the characteristics of the Heart algorithm, a U trap avoidance framework is developed. Finally, with the help of demonstration learning, a DRL based local path planning method is put forward, which is realized with the combination of the Heart algorithm and local planner. According to the simulation results, compared to the TD3fD DRL method, the path planning efficiency and success rate of the IL-PPO2 is higher than the TD3fD with shorter training time. Besides, compared to the dynamic window approach (DWA), the success rate of the IL-PPO2 is improved, and the path smoothness is promoted considering the integration of the Heart algorithm.

Keywords: flapping wing micro aerial vehicle (FWMAV)；obstacle avoidance；U trap；expert system；reinforcement learning；trajectory smooth

0 引 言

扑翼飞行器通过像鸟一样扑动翅膀进行飞行,由于其飞行噪声小,隐蔽性强,近年来受到越来越多的关注[1-3]. 路径规划是扑翼飞行器高效完成任务的关键要素之一. 大中型扑翼机需要更加健壮的机械结构和动力系统作为完成飞行任务的支撑,因此与旋翼机和小型扑翼机相比,其质量较大,结构更加复杂,机动性能较弱,且传感器容易受到机体振动的影响,这对路径规划算法提出了更高的要求.

传统的无人机路径规划算法主要分为局部路径规划(local planner)和全局路径规划(global planner). 局部路径规划算法通过当前无人机的状态与传感器采集的数据直接进行反应式避障,如动态窗口法(dynamic windows algorithm, DWA)[4]、3DVFH+[5]等. Tijmons等[6]针对扑翼飞行器设计了一种水滴避障算法,但只实现了避障,没有实现导航

收稿日期: 2020-11-15；**录用日期:** 2021-01-19.
基金项目: 装备预研教育部联合基金重点项目(6141A02011803).
责任编委: 谢晖.
†通讯作者. E-mail: duchangping@zju.edu.cn.

功能. Oleynikova 等[7] 利用 U 视差图建立局部地图, 实现快速高效避障, 但是得到的不是最优路径. Liu 等[8] 利用凸多面体构成飞行走廊, 在飞行走廊范围内进行飞行轨迹的二次优化, 以得到飞行轨迹. 局部路径规划算法内存消耗小, 计算效率高, 但是通常难以找到最优路径. 以 SLAM 为代表的全局路径规划则通过建图和定位, 获得飞行器周围的障碍物情况, 从而进行全局路径规划. Gao 等[9] 利用激光得到的点云建立地图, 并通过受约束二次规划方法寻找最优路径. Burri 等[10] 利用视觉里程计建立三维栅格地图, 在后端进行非线性路径优化, 进而进行全局导航. 全局路径规划可以获得周围地图与障碍物的情况, 获得的路径更优, 但建图和定位所需要的计算量较大, 规划效率低, 同时容易受到光线以及移动障碍物等周围环境的影响.

近几年, 以深度强化学习 (deep reinforcement learning, DRL) 为代表的基于深度学习的路径规划逐渐成为研究热点之一. Lin 等[11] 通过深度学习对专家系统进行模仿, 并通过强化学习进行参数调优, 引导无人机通过窄缝. Wang 等[12] 改进 DRL 的经验池采样算法, 从经验池随机采样改为对历史轨迹进行采样, 以提升模型训练的效果. Shin 等[13] 提出一种 U-Net 图像处理模型, 对图像进行编码处理后输入到强化学习模型中, 以提高模型对于输入图片的信息提取能力. He 等[14] 提出基于模仿学习的强化学习训练策略, 加快了训练速度. Pfeiffer 等[15] 利用全局路径规划算法得到的路径, 对局部路径规划模型进行监督学习. 相比于传统的路径规划算法, DRL 不需要在无人机飞行过程中进行实时优化, 对计算资源消耗较小; 同时可以直接将原始传感器数据输入到网络中, 构建端到端 (end to end, E2E) 网络, 不需要对数据进行进一步提取. 但是, 这种方法存在一定的缺点, 为了保证算法收敛, 需要大量的环境交互数据进行网络训练, 训练时间长, 并且需要进行网络的参数调优. 同时, 与传统方法相比, DRL 没有数学理论上的支持, 难以保证其稳定性和鲁棒性.

鉴于此, 本文提出一种适应于扑翼飞行器飞行特性的基于强化学习的局部路径规划算法. 强化学习算法作为上层控制器, 利用双目相机, 在有限视角下对前方环境进行感知, 在未知环境中为飞行器规划三维路径. 为了加快强化学习训练过程, 采用专家系统进行示教学习, 并将深度图像进行裁剪分割, 减小图像抖动对网络训练的影响, 同时降低网络输入维度, 便于网络收敛. 为了保证飞行器避开障碍物 (collision free), 提出一种适用于扑翼飞行器运动学特性的心形避障算法, 与 DRL 算法相结合, 进而提升局部路径规划算法的性能.

1 心形局部避障算法

1.1 心形算法简介

扑翼飞行器通过翅膀扑动产生推力与升力, 翅膀的振动会对传感器的测量精度以及机载摄像头的画面清晰度产生较大影响, 从而影响扑翼飞行器的姿态和位置控制精度. 此外, 相对于旋翼与固定翼飞行器的动力系统, 扑翼飞行器由翅膀扑动产生的动力较弱, 同时翅膀扑动会对飞行器的平尾和垂尾产生扰流, 导致其机动性能较弱. 基于以上分析, 所提出避障算法设计的出发点为: 设计一种恒转弯半径的避障算法, 在保持恒定前向速度 v_x 与偏航角速率 $\dot{\psi}$ 的情况下, 以半径 $R_{\rm turn}=v_x/\dot{\psi}$ 进行转向避障, 从而降低对于扑翼飞行器机动性能和姿态控制精度的要求.

代尔夫特理工大学的 DelFly 团队在其小型扑翼飞行器的特性基础上, 提出一种水滴避障策略[6]. 该策略利用双目摄像头探测障碍物的位置, 将双目摄像头的基线与飞行器的中轴线成一安装角, 而非垂直安装. 这样便可以在飞行器前向速度方向的一侧形成一个水滴状的区域作为障碍物探测区域, 保证飞行器与障碍物之间不会发生碰撞. 但是这种算法存在明显的缺点: 摄像头偏置安装导致飞行器重心偏移, 影响飞行器的动力学特性, 避障时只能向一侧转弯等.

基于以上问题, 对该水滴避障策略进行改进, 提出一种适应于扑翼飞行器的避障策略, 如图 1 所示. 因为其外部轮廓线形状类似于心形, 称之为心形算法 (heart algorithm). 图 1 中: ∠FOV 为双目相机的视野范围; 绿色点画线为心形算法的有效探测范围; 黑色虚线为扑翼飞行器的前进路线与转向圆, 2 个黑色虚线圆相切, 并与外部的绿色点画线圆弧同心; 心形轮廓的绿色直线点画线与圆弧点画线相切. $R_{\rm turn}$

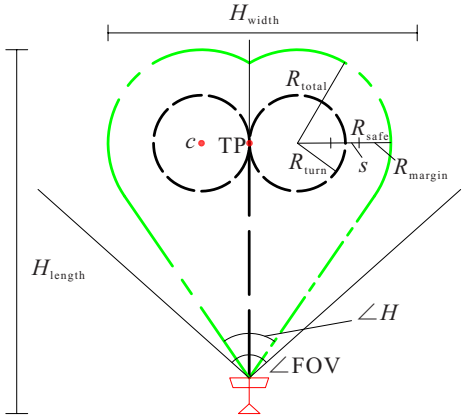


图 1 心形局部避障算法

为设定的扑翼飞行器转弯半径; R_{safe} 为避免飞行器碰撞而设定的安全半径阈值; R_{total} 为心形区域的外部轮廓线半径; s 为扑翼飞行器翼展; R_{margin} 为安全半径裕量; TP (turn point) 为转向点; c 为转向圆圆心.

心形算法的关键之一是检测心形区域范围内是否有障碍物存在, 因此要在双目相机视角 $\angle$FOV 与双目相机有效探测距离 $D_{\max}$ 范围内, 建立合适的心形检测区域. 心形区域的范围确定如图1所示. 根据前文所述, 在恒定前向速度 v_x 与偏航角速率 $\dot{\psi}$ 的作用下, 可以确定飞行器的转弯半径 R_{turn}. D_{TP} 为TP点与相机坐标系原点之间的距离, 是心形区域的关键尺寸参数之一, 需要根据扑翼飞行器尺寸以及相机检测范围等限制条件综合选取. 心形区域外侧圆弧半径由飞行器转弯半径与安全半径确定, 安全半径由扑翼飞行器翼展与安全裕量共同确定. 有

$$R_{\text{total}} = R_{\text{turn}} + R_{\text{safe}} = R_{\text{turn}} + \frac{s}{2} + R_{\text{margin}}, \quad (1)$$

则心形区域的范围可以确定为

$$H_{\text{length}} = D_{\text{TP}} + R_{\text{total}}, \quad (2)$$

$$H_{\text{width}} = 2(R_{\text{total}} + R_{\text{turn}}). \quad (3)$$

心形区域的长度应该在双目相机的有效探测范围内, 应同时满足

$$H_{\text{length}} < D_{\max}. \quad (4)$$

除心形区域的长度应有所限制之外, 心形区域的角度也应限制在相机的有效视角范围内. D_c 表示转向圆圆心与相机坐标系原点之间的距离, 有

$$D_c = \sqrt{R_{\text{turn}}^2 + D_{\text{TP}}^2}\ . \quad (5)$$

因此心形区域的角度可以确定为

$$\angle H = 2\Big(\arctan\Big(\frac{R_{\text{turn}}}{D_{\text{TP}}}\Big) + \arcsin\Big(\frac{R_{\text{total}}}{D_c}\Big)\Big), \quad (6)$$

同时应满足

$$\angle H < \angle \text{FOV}. \quad (7)$$

扑翼飞行器的前向速度在 $2 \sim 4\,\text{m/s}$, 转向角速率在 $\pi/4 \sim \pi/2\,(\text{rad/s})$, 翼展 s 为 $0.8\,\text{m}$. 综合考虑式 (1)~(7) 对心形区域尺寸的限制, 取前向速度为 $2\,\text{m/s}$, 转向角速率为 $\pi/2\,(\text{rad/s})$, 安全半径裕量 $R_{\text{margin}} = s/2 = 0.4\,\text{m}$, TP点距离 $D_{\text{TP}} = 5\,\text{m}$.

1.2　障碍检测策略

通过扑翼飞行器的机载双目相机系统, 可以得到前方环境的视差图 (disparity map), 从而对障碍物的位置与形状等进行三维感知. 为了兼顾图像的处理速度和障碍物的检测精度, 选择尺寸为 $320 \times 240\,\text{pix}$ 的视差图. 同时为了简化计算, 令在相机坐标系下的心形区域中心线与相机坐标系 z 轴重合, 并将视差图沿中心线分割成 $160 \times 240\,\text{pix}$ 的左右两幅图像, 分别进行障碍物检测. 同时, 考虑到扑翼飞行器的宽度远大于高度, 将视差图再次进行裁剪, 取左右两幅子图的中间 $1/3$ 部分 ($160 \times 80\,\text{pix}$) 作为最终障碍物检测的区域.

考虑到视差图的获取与处理过程中存在噪声, 为了减小扑翼飞行器的机体振动对检测效果的影响, 将障碍物检测结果简化为存在障碍物与不存在障碍物两种, 得到视差图并进行图像分割后, 进而得到感兴趣的障碍物检测区域. 在相机参数已知的情况下, 可以得到视差图中每个像素点在相机坐标系下所对应的三维坐标转换关系为

$$\begin{cases} P_z^c = \dfrac{\text{focal} \times \text{base}}{I(x,y)}, \\ P_x^c = \dfrac{P_z^c \times \text{focal}}{x}, \\ P_y^c = \dfrac{P_z^c \times \text{focal}}{y}. \end{cases} \quad (8)$$

其中: focal 为相机焦距; base 为双目相机的基线长度; x、y 分别为像素点在图像坐标系下的横纵坐标; $I(x,y)$ 为该点的像素值.

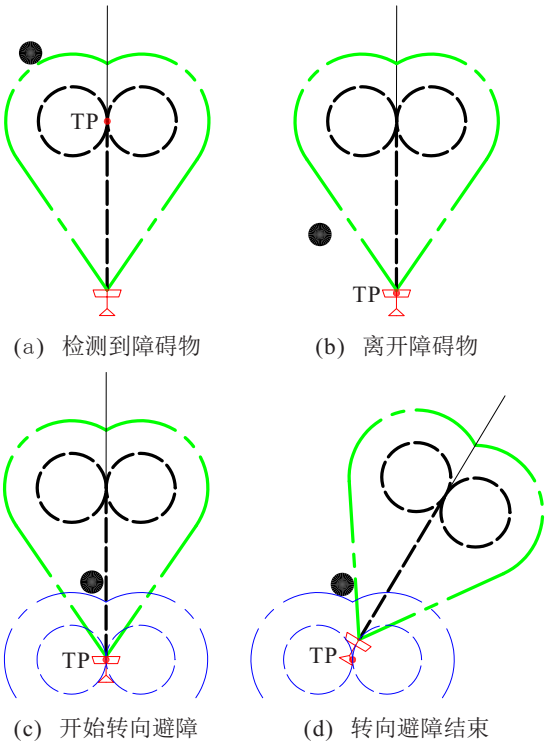
当一侧检测到的在心形区域范围内的点的数量 $N_p > N_{\text{thres}}$ 时, 认为在该侧检测到了障碍物. 实验中, 取判断阈值 $N_{\text{thres}} = 10$. 若在心形区域的一侧检测到了障碍物, 而另一侧未检测到障碍物, 则可以直接判断该侧检测到了障碍物; 若心形区域两侧同时检测到了障碍物, 则无人机向距离目标点较近的一侧转向.

1.3　心形算法避障流程

进行算法实现时, 为了提高避障的鲁棒性, 对前文所述方法进行如下改进. 首先, 在飞行器初始状态时, 应保证心形区域内没有障碍物. 由于算法只检测心形区域内是否有障碍物, 而不会检测障碍物的具体位置, 若初始状态下心形区域内存在障碍物, 则飞行器飞向TP点的过程中有可能与障碍物发生碰撞. 其次, 上文中TP点的位置 D_{TP} 由与相机坐标系原点之间的距离决定, 但是受到扑翼飞行器机体振动以及所在位置信号强度的影响, 飞行器的位置获取精度可能较低, 并存在时延的问题. 因此, 在判断是否到达TP点时, 增加前向速度对于时间积分的判断项. 在算法检测到障碍物并确定TP点时, 记录当前时间戳 t_0. 在飞向TP点的过程中, 若 $\max((t_{\text{curr}} - t_0)v, D_{\text{curr}}) \geqslant D_{\text{TP}}$, 则认为飞行器已经到达了TP点. 其中: v 为飞行器的前向速度; D_{curr} 为传感器得到的

飞行器当前位置与t_0时刻位置之间的距离;$t_{\rm curr}$为当前时刻的时间戳.

在扑翼飞行器飞行过程中,会一直对心形检测范围(绿色点画线)中的障碍物进行检测. 如果没有检测到障碍物,则按照设定的导航算法进行导航,不启动心形避障算法. 心形算法的避障流程如图2所示. 设心形避障算法在心形左侧检测到障碍物,算法将保持飞行器的航向角不变,直至飞行至TP点. 在飞行至TP点后,心形算法将再次对飞行器当前的前方心形区域进行障碍物检测. 如图2(a)所示,若传感器检测到在心形检测范围内存在障碍物,则启动心形局部避障算法;如图2(b)所示,若前方心形区域已经没有障碍物,则飞行器结束本次心形避障算法,转入局部导航算法(图2中所有的TP点均为同一TP点,即图2(a)所确定的TP点,图2(b)~(d)中TP点表示飞行器已经到达图2(a)中的TP点后,TP点相对于飞行器的位置);如图2(c)所示,若前方心形区域内仍然检测到障碍物,则扑翼飞行器将按照图2(d)所示方法进行避障. 扑翼飞行器将在心形区域内未检测到障碍物的一侧,沿着给定的转弯轨迹(蓝色虚线)进行转向,直至当前位置的前方心形区域未检测到障碍物,结束本次心形避障算法,转至局部导航. 其中,蓝色虚线圆即为图2(a)中的转向圆.



(a) 检测到障碍物 (b) 离开障碍物

(c) 开始转向避障 (d) 转向避障结束

图 2 心形算法避障流程

1.4 心形算法分析

心形算法能够保证较高的避障成功率,主要是由于其基本策略保证了飞行器与障碍物之间的最小距离. 无论心形区域的形状参数如何,障碍物的位置在哪里,飞行器与障碍物之间的最小距离都是安全阈值$R_{\rm margin}$. 因此,在安全阈值选取合适的情况下,飞行器能够保证不与障碍物发生碰撞.

算法1 心形算法.

输入: pcl点云数据, target目标位置;

输出: $\dot{\psi}$偏航角速度.

```
S ← getsituation(pcl)
for t ∈ [1, T] do
    pcl ← getpcl( )
    pos ← getpos( )
    switch S do
      case obstacle free
        ψ̇ ← localplanner(pos, target)
        L ← leftfindobs(pcl)
        R ← rightfindobs(pcl)
        if L or R then
          TP ← getturnpoint( )
          S ← to TP
        end if
      end case
      case to TP
        ψ̇ = 0
        if dist(pos, TP) < d_thres then
          S ← turn
        end if
      end case
      case turn
        ψ̇ ← ψ̇_set
        L ← leftfindobs(pcl)
        R ← rightfindobs(pcl)
        if not L and not R then
          S ← obstacle free
        end if
      end case
    end switch
end for
```

飞行器与障碍物之间产生最小距离的情况如图3所示,此时心形算法恰好在心形区域的对称线上检测出了障碍物. 到达TP点后,按照上文中的策略,飞行器选择沿转向圆向距离目标点较近的一侧转向. 以向右侧转向为例,当飞行器前进至图3所示位置时,飞行器距离障碍物的距离最近为$R_{\rm margin}$. 从图3中容易得出,其他情况下,飞行器距离障碍物的距离均大于$R_{\rm margin}$. 心形算法使飞行器始终在图3中的绿色安全区域内行进,不会进入红色的安全半径裕量区域,从而保证飞行器的安全性.

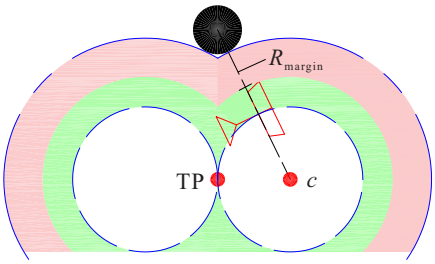


图 3　心形避障算法最小距离情况

1.5　U型障碍避障策略

当前方障碍物体积较小时,心形算法能够及时检测并躲避障碍物. 但是本文使用的双目摄像头视角与有效感知距离有限,在遇到U型障碍物时难以及时探测出障碍物的位置,找不到出口位置,最终困入U型障碍中. 飞行器如何躲避U型障碍的2个关键分别是如何知道已经处于U型障碍中,以及如何找到合理的路径离开U型障碍. 基于上节设计的心形避障算法,本节将设计一种针对U型障碍的避障策略.

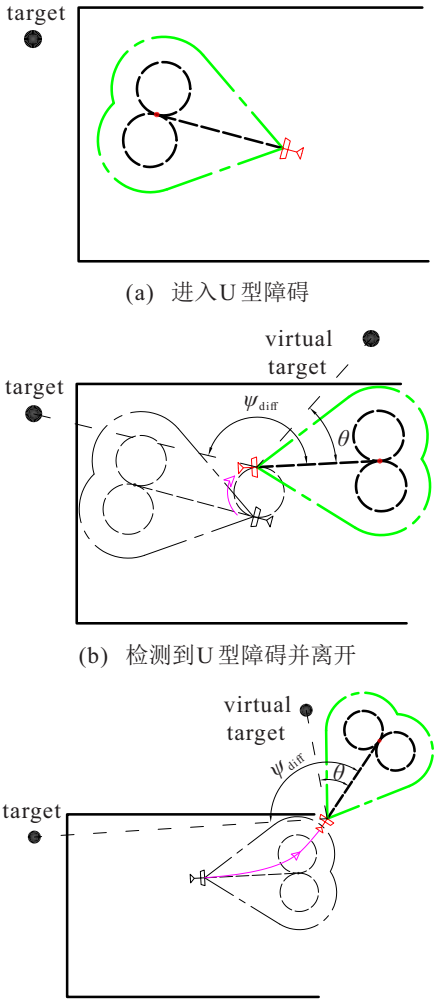
算法过程如图4所示. 图4(a)中,飞行器朝向目标点飞行,障碍探测范围如绿色点画线所示. 由于U型障碍没有进入障碍探测范围内,飞行器认为前方没有障碍,从而进入U型障碍内. 当算法探测到障碍物时,由于算法无法知道障碍物的形状,此时算法仍然不清楚已经进入了U型障碍. 依据心形算法策略,如图4(b)所示,飞行器将在没有探测到障碍物的一侧进行定半径转弯,直至前方的心形区域中没有探测到障碍物. 由于U型障碍的形状特性,在心形区域参数选择合适的情况下,无人机最终将会旋转到沿U型障碍的侧壁向外侧飞行. 令ψ_{diff}为当前扑翼飞行器飞行方向和飞行器当前位置与目标点连线之间的角度差,即当前航向角与期望航向角之间的偏差;ψ_{thres}为设定的判断阈值,在实验中取$\psi_{\text{thres}}=3\pi/4$. 此时,若$|\psi_{\text{diff}}|>\psi_{\text{thres}}$,则认为无人机已经进入U型障碍中,启动U型障碍的避障算法.

算法2　U型障碍避障算法.

输入: target目标位置;

输出: $\dot{\psi}$角速度.

1. for $t\in[1,T]$ do
2. 　pos $\leftarrow$ getpos( )
3. 　$\psi_{\text{diff}}\leftarrow$ getyawdiff(pos, S_{target})
4. 　if $\psi_{\text{diff}}<\psi_{\text{thres}}$ then
5. 　　$\dot{\psi}\leftarrow$ heart( )(algorithml)
6. 　else
7. 　　VT $\leftarrow$ getvirtualtarget(pos)
8. 　　$\dot{\psi}\leftarrow$ localplanner(pos, VT)
9. 　end if
10. end for



(a)　进入U型障碍

(b)　检测到U型障碍并离开

(c)　离开U型障碍

图 4　U型障碍避障策略

受到走出迷宫的一种策略的启发,本文使用类似的策略引导无人机快速离开U型障碍. 一种快速走出迷宫的算法就是一直沿着迷宫墙壁的一侧行进,直至走出迷宫,本节采用类似策略. 如图4(b)所示,当启动U型障碍避障算法后,算法会在当前时刻无人机位置靠近U型障碍内壁一侧θ角度处设定一个虚拟的引导目标,引导无人机沿着近侧障碍内壁前进. 虚拟目标的位置随无人机位置的变化而不断变化,但是相对于无人机的角度θ始终不变,本文实验取$\theta=\pi/4$. 设置虚拟目标后,虚拟目标将暂时代替真实目标,无人机将向虚拟目标方向前进. 以图4(b)为例,无人机左侧更接近U型障碍内壁,因此虚拟目标设置于扑翼飞行器的左前方θ角度处. 局部路径规划算法将引导无人机向虚拟目标方向前进. 在虚拟目标的引导下,无人机逐渐靠近左前方的U型障碍内壁,心形算法在心形区域左侧检测出障碍物,并向右侧进行转向避障,直至左侧没有检测出障碍物. 因此在局部路径规划算法与心形算法的交替作用下,无人机将不断尝试向左侧转弯,直至如图4(c)所示,当$|\psi_{\text{diff}}|\leqslant\psi_{\text{thres}}$

时,认为无人机已经离开U型障碍,U型障碍避障算法结束,虚拟目标结束作用,转到局部导航算法,无人机继续向真实目标点前进.

2 基于强化学习的局部路径规划算法

本节提出一种基于模仿学习的路径规划算法(imitate learning PPO2, IL-PPO2). IL-PPO2方法将上节提出的避障算法与局部路径规划算法相结合,解决了心形算法路径平滑度较低的问题. 同时,在连续动作空间与连续状态空间的问题背景下,传统的强化学习方法探索空间大、学习效率低. 因此,本节提出一种新的模仿学习方法,加快算法收敛.

2.1 心形算法存在的问题

心形避障算法更加贴合扑翼飞行器的飞行特性,能够提高避障算法的鲁棒性,但同时存在缺点. 由图2(c)和图2(d)可以看出,飞行器首先沿直线飞行至TP点,然后沿恒定转弯半径转弯,在TP点,飞行器的法向加速度并不连续,存在从0到$v_x^2/R_{\rm turn}$的骤变,导致飞行器的路径跟踪误差. 若直接将心形避障算法与局部路径规划算法相结合,则会导致路径规划的可靠性降低. 因此,本节设计一种基于强化学习的局部路径规划算法,与心形算法相结合,通过强化学习对两种策略的结合进行调优,以提高路径平滑度与可靠性.

2.2 模仿学习算法设计

强化学习的状态空间与动作空间巨大,尤其对于局部路径规划问题,如果单纯依靠模型在环境中搜索,则找到最优的多步决策策略十分困难. 而利用专家示教的数据进行模型预训练,可以显著缓解这一困难,该方法称为直接模仿学习[16]. 在模仿学习阶段,首先,获得n组专家示教轨迹对模型进行训练,作为强化学习的初始策略;然后,在与环境交互中学习更优的策略. 但是,直接强化学习存在明显的缺点. 专家系统采集的是没有碰撞的轨迹,因此在预训练过程中,模型训练最终得到的是对于局部导航策略的有偏估计. 当产生碰撞时,模型对于状态估计的可信度大大下降,在时序差分的策略中导致误差累计,难以收敛到最优的多步策略.

为了解决直接模仿学习在预训练阶段的有偏估计问题,本文避免将模型训练分为模仿学习与环境交互阶段,而是将2个阶段融合为1个阶段. 模型输出动作的策略为

$$a_t=\begin{cases}\text{expert}(s_t),\ \text{rand}<\varepsilon;\\ \pi_\theta(s_t),\ \text{rand}>\varepsilon.\end{cases}\tag{9}$$

其中:$\varepsilon=\max(0,\ \varepsilon_{\rm decay}^N),\varepsilon_{\rm decay}\in(0,1)$为衰减因子;$\text{rand}\in\text{random}(0,1)$,$N$为训练轮数. 开始训练时,专家系统为主导策略. 随着轮数增加,策略网络逐渐增加主导权直至完全主导动作选择. 本文选择PPO2[17]作为强化学习模型的学习策略. PPO2使用Actor-Critic网络结构,输出连续的动作空间,并且通过优势函数控制梯度优化的范围和方向,从而加快模型的收敛. 本文使用专家系统进行示教学习,称为IL-PPO2 (imitate learning PPO2)算法. PPO2的策略更新公式如下所示:

$$\begin{aligned}&J^{\theta_k}(\theta)\approx\\&\sum_{(s_t,a_t)}\min\Big(\frac{p_\theta(a_t\,|\,s_t)}{p_{\theta^k}(a_t\,|\,s_t)}A^{\theta^k}(s_t,a_t),\\&\text{clip}\Big(\frac{p_\theta(a_t\,|\,s_t)}{p_{\theta^k}(a_t\,|\,s_t)},1-\varepsilon,1+\varepsilon\Big)A^{\theta^k}(s_t,a_t)\Big).\end{aligned}\tag{10}$$

IL-PPO2使用专家系统进行辅助策略梯度下降,式(10)中$p_\theta(a_t\,|\,s_t)$变为$p(a_t\,|\,s_t)$,有

$$p(a_t\,|\,s_t)=\begin{cases}1,\ a_t=\text{expert}(s_t);\\ p_\theta(a_t\,|\,s_t),\ a_t=\pi_\theta(s_t).\end{cases}\tag{11}$$

则有

$$\begin{aligned}E_t(P(a\,|\,s))=&E_t(P(a\,|\,s)\,|\,a=\text{expert}(s))+\\&E_t(P_\theta(a\,|\,s)\,|\,a=\pi_\theta(s))=\\&P(\text{rand}<\varepsilon)+P_{\theta,t}(a\,|\,s)P(\text{rand}>\varepsilon).\end{aligned}\tag{12}$$

由式(12)可得

$$E_t(P(a\,|\,s))=P_\theta(a\,|\,s)+\varepsilon(1-P_\theta(a\,|\,s))\geqslant P_\theta(a\,|\,s).\tag{13}$$

对于PPO2策略,其算法的核心思想为在限制的$[1-\varepsilon,1+\varepsilon]$范围内,最大化策略更新公式. 通过式(13)可得,在IL-PPO2中,策略更新公式的更新项期望大于等于原始PPO2策略更新公式. 由此可知,相比于原始PPO2公式,所提出的模仿学习算法能够更快地进行策略更新,提高训练效率.

2.3 网络结构

目前,很多文献将视觉图像直接作为网络输入,构建端到端的强化学习网络,并展现了较为良好的学习效果. 对于扑翼飞行器而言,机身的振动会对图像的清晰程度造成很大影响,导致图片的有效信息减少. 同时,卷积层得到的信息量较大,不利于强化学习网络的收敛. 因此,本文将双目相机得到的深度图像进行分割,得到3×8的图像块,将每一块中的最小深度作为深度信息,组成1×24的深度信息向量,作为模型的感知输入.

最终,本文所设计的网络如图5所示. 输入分别为1×24的图像信息以及1×3的扑翼飞行器相对于目标点的位置信息$(d_{\rm hori}, d_{\rm vert}, \psi_{\rm diff})$,包括扑翼飞行器距离目标点的三维距离$d$在水平面上的投影距离$d_{\rm hori}$,$d$在地向的投影距离$d_{\rm vert}$,以及扑翼飞行器的航向角偏差$\psi_{\rm diff}$. 网络的动作输出向量为$(v_f, v_u, \dot{\psi})$,分别对应扑翼飞行器的前向速度、地向速度以及航向角速度.

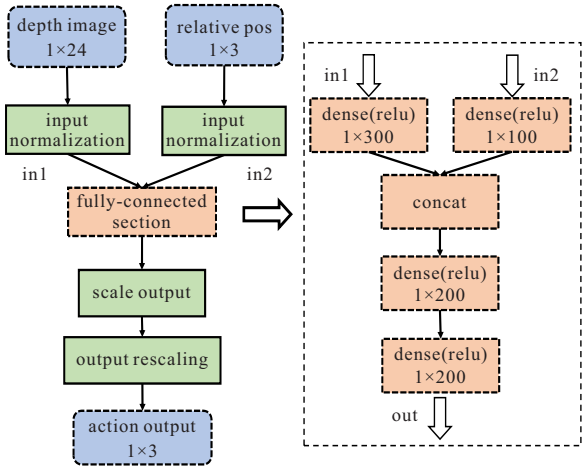


图 5　IL-PPO2网络结构

2.4　奖励函数设计

奖励函数会在很大程度上影响模型的训练效果. 为了提高训练效果,设置的奖励函数为稠密奖励形式,包括过程奖励和到达目标奖励.

过程奖励分为位置奖励和速度奖励. 位置奖励为

$$R_{\rm pos}=2^{\frac{1}{d_{\rm hori}}}+\pi-|\psi_{\rm diff}|+2^{\frac{1}{d_{\rm vert}}}. \tag{14}$$

其中: $d_{\rm hori}$为水平方向距离目标点的距离,$\psi_{\rm diff}$为距离目标点的航向角偏差,$d_{\rm vert}$为地向距离目标点的距离. 使用指数形式的奖励可以在无人机靠近目标时提高算法稳定性,防止无人机在目标点附近转圈. 速度奖励为

$$R_{\rm vel}=f_{\rm velx}\times w_{\rm velx}+f_{\rm velz}\times w_{\rm velz}+f_{\omega}\times w_{\omega}. \tag{15}$$

其中: $f_{\rm velx}$、$f_{\rm velz}$、f_{ω}分别为前向速度、地向速度和航向角速度的奖励符号; $w_{\rm velx}$、$w_{\rm velz}$、w_{ω}为对应的奖励权重. 对于前向速度和地向速度,$f_*=\mathrm{sign}(v_*\times d_*)$; 对于航向角速度,$f_*=\mathrm{sign}(\omega\times\psi_{\rm diff})$.

目标奖励为扑翼飞行器到达目标奖励,有

$$R_{\rm target}=\begin{cases}R_{\rm reach}, \text{ success};\\ R_{\rm collision}, \text{ collision}.\end{cases} \tag{16}$$

为了鼓励飞行器尽快到达目标点,添加时间奖励项

$$R_{\rm time}=-0.1.$$

则总奖励函数为

$$R=\begin{cases}R_{\rm target}, \text{ finish};\\ R_{\rm pos}+R_{\rm vel}+R_{\rm time}, \text{ otherwise}.\end{cases} \tag{17}$$

3　实验验证

本节对以上提出的算法进行仿真验证. 仿真平台采用Ubuntu 18.04,软件架构采用ROS melodic,仿真环境采用gazebo,以PX4开源栈[18]作为底层飞控的软件在环仿真(software in the loop, SITL).

3.1　心形算法性能对比

本节对心形算法、水滴算法以及DWA三种避障算法的避障性能进行测试,环境如图6所示. 地图尺寸为30 m × 30 m,其中随机分布着柱形障碍. 对于每种算法设置100个随机目标点,记录每种算法到达100个随机目标点过程中的运行步数以及与障碍物的碰撞次数. 仿真结果如表1所示.

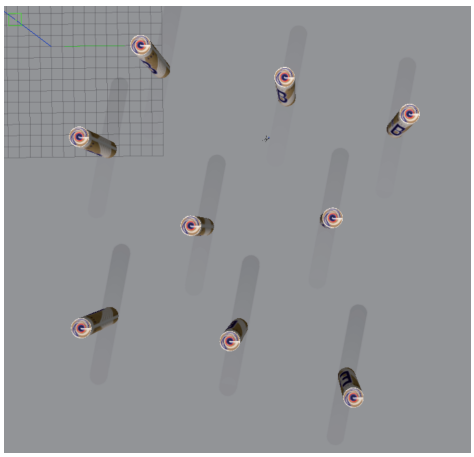


图 6　心形避障算法评估环境

表 1　不同算法避障性能比较

策略	平均运行步数	成功率/%
心形算法(本文算法)	333	100
水滴算法	365	100
DWA	299	81

结果表明,相比于水滴算法,本文算法在同样保持了100 %成功率的情况下,将平均运行时间提升了8.77 %. 这主要是由于水滴算法的相机基线存在安装偏置角,只能向一侧进行避障,而本文的心形算法解决了这一问题,从而可以选择更加合理的避障方式. DWA算法虽然平均运行步数较短,但是其成功率远低于心形算法. 这主要是由于相机的视角有限,在无人机需要大幅转向($|\psi_{\rm diff}|>90^{\circ}$)时,难以察觉到视野范围之外且距离较近的障碍物,从而发生碰撞. 心形算法在牺牲了一定运行速度的情况下,保证了运行成功率. 由此可见,心形算法更适合引导扑翼飞行器进行避障.

3.2 U型障碍避障算法验证

本节对所提出的U型障碍避障算法进行验证. 设置地图与飞行轨迹如图7所示.

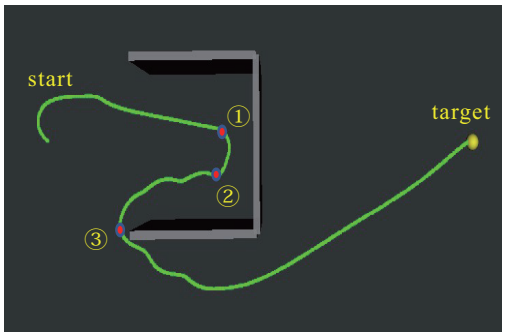


图 7 U型障碍避障策略飞行轨迹

飞行器从起始点向目标点飞行. 在1号点, 飞行器检测到前方存在障碍物, 启动心形算法, 开始进行转向. 在2号点, 飞行器与目标点之间的航向角偏差 $|\psi_{\rm diff}| > \psi_{\rm thres}$, 触发U型障碍避障策略. 在虚拟目标的引导下, 在2号点与3号点之间, 飞行器沿障碍内壁前进, 并不断尝试绕过障碍, 因此出现航向角的波动. 在3号点, $|\psi_{\rm diff}| < \psi_{\rm thres}$, 算法判断飞行器已经绕出U型障碍, U型障碍避障策略结束, 飞行器转换回局部导航策略, 直至到达目标点. 从上述分析可以看出, 所提出的U型障碍避障策略能够成功引导飞行器避开U型障碍.

3.3 局部导航算法性能比较

本节对基于强化学习的扑翼飞行器局部导航算法在训练时间、导航性能等方面进行评估. 网络搭建与训练采用 tensorflow 1.14 平台, 采用 Nvidia GTX 1 050 Ti 显卡加速训练. 网络训练参数如表2所示.

表 2 强化学习训练超参数

参数	数值
批量大小 (batch size)	32
每轮最大步数	1 000
Actor 网络学习率	1e-7
critic 网络学习率	2e-7
专家示教折扣率 ($\gamma_{\rm imit}$)	0.998
时序衰减因子 (γ)	0.95

3.3.1 专家示教系统搭建

采用心形算法与DWA相结合的方式作为专家示教系统. 为了保证专家系统与强化学习策略较高的契合度, DWA算法同样将采集到的图像分割成3 × 8共24个图像子块. 对于每个子块, 根据检测到的与障碍物之间的距离以及与目标点之间的距离计算评价值, 选取评价值最高的子块所在的方向作为飞行器的前进方向. 心形算法在距离障碍物较近时具有优势, 能够保证无人机不会与障碍物发生碰撞. DWA在距离障碍物较远时具有优势, 能够在评价函数的辅助下选择出一条局部最优轨迹. 专家系统将这两种方法相结合, 在无人机探测到最近的障碍物距离 $d_{\rm min} > d_{\rm thres}$ 时, 采用DWA; 在 $d_{\rm min} < d_{\rm thres}$ 时, 采用心形算法避障.

3.3.2 训练结果对比

本节将所提出算法与TD3fD避障算法[14]进行比较. TD3fD算法在直接模仿学习算法TD3的基础上进行改进, 在损失函数中加入逐渐衰减的模仿学习损失项, 引导模型从模仿学习阶段向环境交互阶段过渡, 在一定程度上提高了算法效果. 所提出的IL-PPO2策略将按照上文提出的模仿学习策略进行学习800轮; 根据文献[14]中提出的训练策略, TD3fD算法将先在模仿学习阶段训练200轮, 然后在交互环境中训练600轮. 2个策略网络参数均随机初始化.

两种策略的训练过程奖励值如图8所示. 由图8可见, IL-PPO2算法在训练时间与收敛状态方面均优于TD3fD算法. 对于IL-PPO2算法, 专家系统的指导权重逐渐下降, 算法能够以平稳的状态从专家系统过渡到模型引导, 每轮的平均奖励值逐渐上升; 对于TD3fD算法, 进入环境交互阶段后, 专家系统的指导权重逐渐减小, 奖励值产生了较大波动, 在800轮时, 模型仍处于波动上升的状态, 需要更长的训练时间才能达到更好的状态.

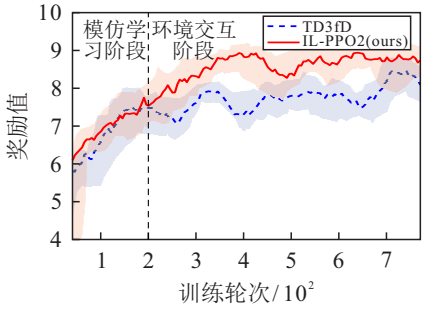
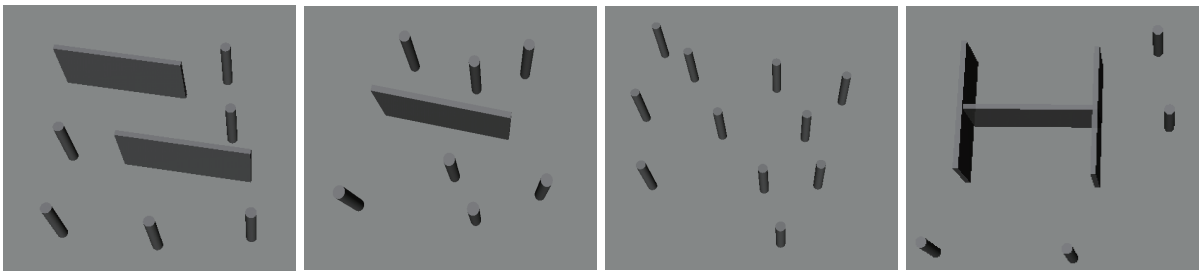


图 8 IL-PPO2训练奖励值

3.3.3 网络性能对比

本节对IL-PPO2、TD3fD以及DWA (专家系统)3种策略的性能进行对比, 用于评估的4个环境如图9所示. 其中: world 1为训练环境, world 2、world 3以及world 4为测试环境. 每个环境下设置100个随机出现的目标点, 记录3种算法达到目标点的平均运行步数与成功率. IL-PPO2的部分运行轨迹如图10所示. 可以看出, 该算法能够学会在三维环境下躲避障碍, 到达目标点.

3种策略的性能对比结果如表3所示, 平均角速度的对比结果如图11所示. 由表3可见, 在world 1 ~ world 3中, 在与心形算法相结合下, IL-PPO2保持



(a) world 1　(b) world 2　(c) world 3　(d) world 4

图 9　IL-PPO2评估环境

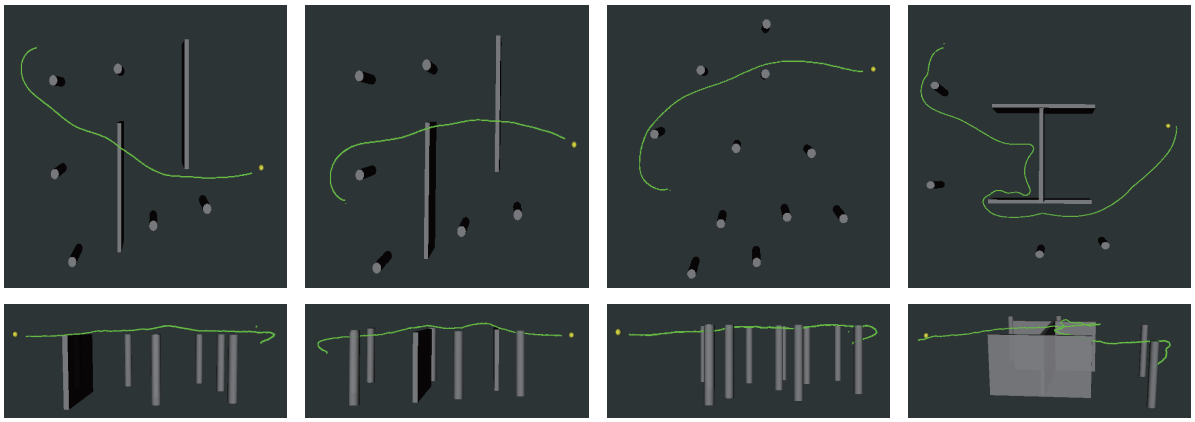


图 10　IL-PPO2局部路径规划轨迹

了100%的成功率,成功率与运行步数均远好于TD3fD,略逊于DWA. world 4测试了U型障碍避障算法的性能. IL-PPO2仍然保持了很高的成功率,未能到达目标点主要是由于以下两方面的原因:首先,U型障碍避障算法导致路径长度增加,超过所限制的最大步数;另外,在躲避U型障碍过程中存在大幅度转向的过程,无人机感知精度与控制精度降低,导致与障碍物发生碰撞. DWA算法主要通过爬升躲避U型障碍,飞行器爬升率的限制导致其成功率降低. TD3fD的平均步长最少,主要是由于其未能成功避开U型障碍,部分长路径的任务未能到达目标,导致其成功率与奖励值较低.

表 3　策略结果对比

环境	策略	平均步数	平均奖励	成功率/%
world 1 (train)	IL-PPO2	767	9.11	**100**
	DWA	**717**	**9.81**	84.6
	TD3fD	837	8.50	37.6
world 2 (test)	IL-PPO2	759	9.14	**100**
	DWA	**736**	**10.21**	86.8
	TD3fD	836	8.56	52.6
world 3 (test)	IL-PPO2	740	10.02	**100**
	DWA	**716**	**10.54**	**100**
	TD3fD	868	8.60	82.9
world 4 (test)	IL-PPO2	798	9.531	**97.3**
	DWA	756	**10.88**	74.1
	TD3fD	**750**	8.64	52.3

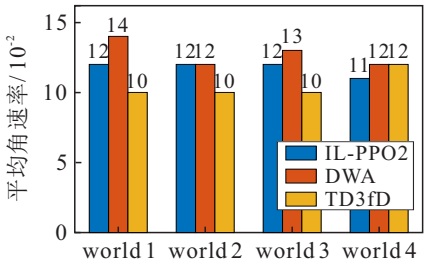


图 11　平均角速率对比

由图11可见,在4个环境下,IL-PPO2所需要的角速度均小于DWA,表明强化学习模型能够解决心形算法加速度不连续的问题. 而TD3fD算法角速度较小,主要是因为在一些情况下,该算法未能学会通过转向避开障碍物.

4　结　论

本文提出一种在未知三维环境中扑翼飞行器的避障与导航策略,针对扑翼飞行器的特性,提出了一种高鲁棒性的心形避障策略. 同时,将心形避障算法与局部路径规划算法相结合,提出一种专家系统辅助的强化学习局部路径规划算法,提高了路径规划的路径平滑度与成功率. 仿真实验表明,所提出算法在保证成功率的情况下,性能与传统路径规划算法类似,超过所选择的强化学习算法,且算法所需角速度低于传统算法,表明所提出的强化学习算法提高了路径平滑度.

后续的工作中,在更加复杂的仿真环境中训练测

试模型的性能,并在扑翼飞行器样机上进行实验,以验证所提出算法的性能.

参考文献(References)

[1] Hassanalian M, Abdelkefi A. Classifications, applications, and design challenges of drones: A review[J]. Progress in Aerospace Sciences, 2017, 91: 99-131.

[2] Jafferis N T, Helbling E F, Karpelson M, et al. Untethered flight of an insect-sized flapping-wing microscale aerial vehicle[J]. Nature, 2019, 570(7762): 491-495.

[3] He W, Meng T T, He X Y, et al. Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances[J]. IEEE Transactions on Cybernetics, 2019, 49(4): 1524-1535.

[4] Brock O, Khatib O. High-speed navigation using the global dynamic window approach[C]. Proceedings 1999 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 1999: 341-346.

[5] Vanneste S, Bellekens B, Weyn M. 3DVFH+: Real-time three-dimensional obstacle avoidance using an octomap[C]. International Conference on Mechatronics, Robotics, and System Engineering. York, 2014: 91-102.

[6] Tijmons S, de Croon G C H E, Remes B D W, et al. Obstacle avoidance strategy using onboard stereo vision on a flapping wing MAV[J]. IEEE Transactions on Robotics, 2017, 33(4): 858-874.

[7] Oleynikova H, Honegger D, Pollefeys M. Reactive avoidance using embedded stereo vision for MAV flight[C]. IEEE International Conference on Robotics and Automation (ICRA). Piscataway: IEEE, 2015: 50-56.

[8] Liu S K, Watterson M, Mohta K, et al. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments[J]. IEEE Robotics and Automation Letters, 2017, 2(3): 1688-1695.

[9] Gao F, Shen S J. Online quadrotor trajectory generation and autonomous navigation on point clouds[C]. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). Piscataway: IEEE, 2016: 139-146.

[10] Burri M, Oleynikova H, Achtelik M W, et al. Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway: IEEE, 2015: 1872-1878.

[11] Lin J, Wang L Q, Gao F, et al. Flying through a narrow gap using neural network: An end-to-end planning and control approach[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Piscataway: IEEE, 2019: 3526-3533.

[12] Wang C, Wang J, Shen Y, et al. Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2019, 68(3): 2124-2136.

[13] Shin S Y, Kang Y W, Kim Y G. Reward-driven U-Net training for obstacle avoidance drone[J]. Expert Systems With Applications, 2020, 143: 113064.

[14] He L, Aouf N, Whidborne J F, et al. Deep reinforcement learning based local planner for UAV obstacle avoidance using demonstration data[J/OL]. 2020, arXiv: 2008.02521.

[15] Pfeiffer M, Schaeuble M, Nieto J, et al. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots[C]. IEEE International Conference on Robotics and Automation (ICRA). Piscataway: IEEE, 2017: 1527-1533.

[16] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 390-392.
(Zhou Z H. Machine learning[M]. Beijing: Tsinghua University Press, 2016: 390-392.)

[17] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J/OL]. 2017, arXiv: 1707.06347.

[18] Meier L, Honegger D, Pollefeys M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms[C]. IEEE international conference on robotics and automation (ICRA). Piscataway: IEEE, 2015: 6235-6240.

作者简介

王思鹏(1997−), 男, 硕士生, 从事飞行器导航与路径规划的研究, E-mail: wangsipeng@zju.edu.cn;

杜昌平(1978−), 男, 副教授, 博士, 从事飞行器导航制导与控制等研究, E-mail: duchangping@zju.edu.cn;

郑耀(1963−), 男, 教授, 博士, 从事飞行器设计与推进等研究, E-mail: yao.zheng@zju.edu.cn.

(责任编辑: 魏 冰)