



计算机科学

COMPUTER SCIENCE

基于空间运动约束的无人机碰撞回避规划

罗熊丰, 翟象平

引用本文

罗熊丰, 翟象平. [基于空间运动约束的无人机碰撞回避规划](#)[J]. 计算机科学, 2022, 49(9): 194-201.

LUO Xiong-feng, ZHAI Xiang-ping. [Collision Avoidance Planning for Unmanned Aerial Vehicles Based on Spatial Motion Constraints](#)[J]. Computer Science, 2022, 49(9): 194-201.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于双向蚁群算法的网络攻击路径发现方法](#)

Network Attack Path Discovery Method Based on Bidirectional Ant Colony Algorithm

计算机科学, 2022, 49(6A): 516-522. <https://doi.org/10.11896/jsjcx.210500072>

[基于终态神经网络的冗余机械臂重复运动规划](#)

Repeatable Motion Planning of Redundant Manipulators Based on Terminal Neural Networks

计算机科学, 2018, 45(12): 201-205. <https://doi.org/10.11896/j.jissn.1002-137X.2018.12.033>

[Sylvester 时变矩阵方程求解的终态神经网络算法](#)

Terminal Neural Network Algorithm for Solution of Time-varying Sylvester Matrix Equations

计算机科学, 2018, 45(10): 207-211. <https://doi.org/10.11896/j.jissn.1002-137X.2018.10.038>

[动态进化多目标优化中的串式记忆方法](#)

Bunchy Memory Method for Dynamic Evolutionary Multi-objective Optimization

计算机科学, 2016, 43(12): 241-247. <https://doi.org/10.11896/j.jissn.1002-137X.2016.12.044>

[移动环境中任务分析及任务建模方法](#)

Task Analysis and Task Modeling Method in Mobile Environment

计算机科学, 2014, 41(10): 210-215. <https://doi.org/10.11896/j.jissn.1002-137X.2014.10.045>

基于空间运动约束的无人机碰撞回避规划

罗熊丰 翟象平

南京航空航天大学计算机科学与技术学院 南京 211106

(luoxiongfengaugust@qq.com)

摘要 三维空间中,面对移动的障碍物,无人机如何进行运动规划是一个很有趣的研究方向。在动态环境下,传统的基于速度障碍物的算法主要针对二维机器人。用机器人的可达速度集减去导致碰撞的速度集,然后选取合适的速度实现避障动作。根据无人机和障碍物的当前位置和速度进行泛化的三维矢量运算,计算导致碰撞的速度集合。依据无人机制动器的最大速度和最大加速度,对其当前时刻可达到的速度空间进行约束。在这个相减集合中,依据场景的需求制定相应的策略并选取合适的速度,实现三维场景下的回避障碍物规划。针对被抽象为球状的无人机在三维空间下躲避球状障碍物行驶至终点的场景,在虚幻引擎中结合 C++ 编程和图形化编程实现了避障算法并进行验证。通过捕捉不同策略的运动轨迹,度量其耗时,有效地验证了所提算法能够完成无人机在三维空间下的动态障碍物回避任务。

关键词: 动态环境;碰撞回避;运动规划;速度障碍物;可达速度集

中图分类号 TP242

Collision Avoidance Planning for Unmanned Aerial Vehicles Based on Spatial Motion Constraints

LUO Xiong-feng and ZHAI Xiang-ping

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

Abstract In three-dimensional space, how to conduct motion planning when unmanned aerial vehicles (UAV) facing moving obstacles is an interesting research direction. In dynamic environments, traditional algorithms based on the speed obstacles mainly aim at two-dimensional robots, which realizes obstacle avoidance actions by selecting velocities from the reachable velocity set out of the collision velocity set. This paper generalizes algorithms based on the current positions and velocities of UAV and obstacles to calculate the set of velocities that will cause collisions. According to the maximum speed and maximum acceleration of UAV, the velocity space that can be reached at the current moment is restricted. Different strategies are formulated for the needs of various scenes and are selected in a specific method in this subtraction set, so as to avoid obstacles in a three-dimensional scene with specific requirements. Aiming at the scene where UAV abstracted as spherical travels to the destination by avoiding spherical obstacles in 3D space, this paper verifies the obstacle avoidance algorithm by combining C++ and blueprint programming. It captures the movement trajectory of different strategies and records the corresponding consumption time, which demonstrates that the proposed algorithm can effectively complete the dynamic obstacle avoidance task of UAV in three-dimensional space.

Keywords Dynamic environments, Collision avoidance, Motion planning, Velocity obstacle, Reachable velocity set

1 引言

动态环境中的运动规划问题,即根据无人机和障碍物的当前位置和速度,实时选择合适的速度实现无人机的避障动作,是计算机应用中的一个重要问题。该问题在许多现实领域中均有重要意义,如航天航空、自动驾驶、智能机器人等。该问题研究的智能可移动机器人是一门涉及计算机科学、人工智能、机械工程、电子工业技术以及自动化等多种学科的综合交叉学科,集合了这些学科最前沿的研究经验和成果^[1]。目前,汽车领域的无人驾驶技术已日趋成熟,许多商用智能

汽车都已搭载无人驾驶系统。但是,空中无人机的驾驶仍十分依赖人力。对动态环境下无人机避障行为进行研究,将十分有助于空中无人机自动驾驶的飞行轨迹的优化^[2]。由于空中飞行器的能耗往往比地面同量级驾驶器的能耗高,因此在这种情况下,选择更高效的避障逻辑和更精确的避障动作,对于提高无人机航行效率和节省能耗来说至关重要。

本文基于 1998 年 Fiorini 和 Shiller 提出的二维避障算法^[3],提出三维场景下基于现实约束的移动障碍物避让算法,然后在虚幻引擎上进行实现和验证。同时,对基于该避障核心算法的不同回避策略进行对比测试,对其测试结果进行

到稿日期:2021-07-12 返修日期:2022-03-04

基金项目:国家自然科学基金(61701231,61802181)

This work was supported by the National Natural Science Foundation of China(61701231,61802181).

通信作者:翟象平(blueicezhaxp@nuaa.edu.cn)

分析和总结。主要内容包括:

(1)分析可达避障速度空间理论,并将该理论泛化于三维场景。对于速度障碍物,基于无人机和障碍物的位置、速度和大小,计算导致碰撞的速度空间;对于可达速度空间,考虑无人机物理上的现实速度和加速度约束。

(2)设计基于以上理论的回避策略,即径直向目标策略和最大速度策略。

(3)在虚幻四引擎上实现算法。基于面向对象思想,结合C++和虚幻四特有蓝图的开发场景,验证对比不同策略下无人机避开障碍物抵达终点的时间和轨迹。

2 相关工作

2.1 国内外研究现状

动态环境下的运动规划问题是一个历史悠久的问題,也是解决机器人自动化、智能化的一项重要内容。机器人路径规划指在动态环境下,回避动态以及静态的障碍物,找出一条从起点到终点的可行路径,并对该路径过程中的速度进行控制^[4]。自20世纪以来,关于运动规划的很多经典算法不断涌现。

(1)在限制的速度内,已知障碍物的轨迹,将对应的时间点信息添加到机器人的配置中进行计算。Reif等通过在计算出的时间-空间位置图谱中搜索,解决了多边形机器人移动避开复数个多边形障碍物的平面移动问题^[5]。Erdmann等将机器人的时间-空间位置关系离散为连续时间间隔上的一系列空间位置切片^[6]。该方法从根本上解决了每个切片上的静态规划问题,并加入了可供备选的相邻路径解决方案。Fujimura等使用单元分解的思想来表示机器人的时间-空间位置关系,并引入了空单元以连接起点与终点^[7]。

(2)将运动规划问题分解为两个较小的子问题,即路径规划和速度规划。这一方法首先计算避开静态障碍物的可行路径,然后在这一路径上的不同位置选择不同的速度来避开移动障碍物。Kant等使用可见度图的方法计算了路径和速度的对应曲线^[8]。Lee等提出了一种针对两个运动中的机器人无碰撞运动的规划方式,并比较了延长反应时间和降低速度对运动时间的影响^[9]。Fraichard考虑了加速度的界限,并在状态-时间图谱中搜索来进行计算,得出能够产生最小时间轨迹的速度分布方案^[10]。Fraichard等提出了当避开静态障碍物的路径被移动障碍物阻塞时,可以从当前路径方案转到相邻的另一可行路径的算法^[11]。Fujimura考虑了机器人在动态变化的路径网络上移动的情况,该机器人移动路径网络上的节点可能会被移动的障碍物暂时遮挡^[12]。

(3)生成环境的可访问性图。Fujimura等将其定义为机器人以最大速度运动时会与障碍物发生碰撞的点的轨迹^[13]。这些点形成碰撞边界,可以链接在一起构建从起点到目标的路径。如果机器人的运动速度快于障碍物,则通过搜索该图而计算出的路径将是耗时最短的路径。环境的可访问性图的概念由Fujimura进行了扩展,适用于缓慢移动的机器人和短暂出现的障碍物(即在环境中出现不久后就消失的障碍物)的情况^[14]。动态环境中的在线运动规划主要强调推理和决策,而很少关注机器人动力学。Sanborn为在路面交通情景中

移动的机器人制定了移动避障规则^[15]。移动机器人通过预测路径上的空间在时间轴上的空闲状态来应对障碍物,障碍物将在不同时间点上穿越其路径点。这种方法考虑了环境的制约因素,但是忽略了智能机器人自身的物理限制。Tsoubouch等则增加了对机器人速度的限制,其中使用笛卡尔坐标系中的区域来表示机器人行动区域,并避免移动障碍物扫过这一区域^[16]。Hayward等提出了一个与联网情景下动态避开多个障碍物有关的问题,即通过计算出碰撞发生的时间来预测和避免未来可能发生的碰撞^[3]。

(4)Fiorini等使用障碍物和机器人的速度、位置信息直接进行计算,预测潜在的碰撞,确定在动态环境中移动的机器人的轨迹^[17]。此方法利用了速度障碍物(VO-velocity Obstacle)的概念,将动态环境映射到机器人速度空间中^[18-19]。速度障碍物是机器人速度的集合,该集合中的速度将在给定的时间范围导致机器人与障碍物发生碰撞。然后可以简单地通过选择速度障碍物之外的速度来计算得出回避操作。通过选择满足机器人制动器约束的速度,可以确保该避障行为在现实环境中是可行的。

(5)与机器学习相关的模糊逻辑算法^[20-23]。该算法是研究者通过观察现实中汽车驾驶员的习惯并进行研究得出的,即对每次采集到的周围物体位置、速度信息进行信息模糊化,再根据建立的知识库、规则库进行模糊推理,而后将得出的推理结果解模糊化,输出机器人的矢量速度变化。该方法正如驾驶员驾驶时并不会精准计算观察到的路面信息,而是根据提前建立的认知知识库做出判断。该算法由于不需要解析出避障行为,而是将这一过程经验化,因此在普通算法失效的复杂环境下表现很出色。但其也具有一定缺陷,即用于训练的经验不一定完备,会导致避障系统存在碰撞的风险。而在使用大量经验输入对知识库进行训练时,知识库内的推理规则会急速增多^[24]。

(6)基于运动信息建立动态碰撞区模型是一种较新的无人机避障算法^[25]。这种算法根据无人机和障碍机的位置、速度、入侵角等运动状态信息,使用几何方法计算,分别讨论了无人机在不采取回避动作时导致碰撞的区域,和采取最大转向动作时恰好躲避开障碍的界限区域,其分别对应无机碰撞区模型和最大机动碰撞区模型。由这两个模型提出不可规避区的概念,再由不可规避区定义了安全飞行包络,即无人机恰好能进行回避的分界线。

上面几种著名的算法均很出色,在特定的场景表现优异,但普遍仅适用于平面和二维的场景,且较少涉及基于现实约束的可行性研究。Fiorini等的算法考虑了机器人制动器的约束,即最大加速度,但仍忽视了机器人的最大速度限制即动力学的限制。而模糊逻辑算法属于机器学习范畴,该算法对于在三维空间内的躲避行为缺少训练样本,因此不适用于本文中三维空间内的避障运动规划研究。较新的无机动、最大机动碰撞区域模型算法讨论的是无人机面对单一侵入机的回避算法,未对同时出现多障碍物的情形进行处理。

2.2 三维仿真框架

本文通过虚幻四引擎上的开发工具和框架进行验证。

2.2.1 虚幻四简介

虚幻四引擎(Unreal Engine IV)是由 Epic 公司开发,并于 2016 年发布的游戏引擎。虚幻引擎是一套完整的集成工具,功能涵盖构建游戏、模拟和可视化,满足了用户关于图形化工具的绝大部分功能需求;也具备足够的灵活性,其功能的可扩展性和高度自定义的特性,允许用户自由添加新的自建插件模块,可以满足不同规模的开发团队的需求。全球的游戏开发商已经广泛使用虚幻四引擎,并制作出了许多享誉好评的游戏,例如由蓝洞公司使用虚幻四引擎开发的《绝地求生》(PUBG),游戏数据统计网站 SteamSpy 于 2017 年公布的数据显示其累计销量已达 1800 万份。同时,由虚幻四衍生的强大的插件工具和高效的制作流程,在整个游戏开发领域都产生了深远的影响。

在仿真模拟方面, López 等在 2017 年机器学习大会 (CoRL)上发布的 CARLA 就是一款虚幻四引擎支持的开源模拟器,用于支持汽车自动驾驶系统的开发、训练和检验^[26]。相比其他模拟测试环境,虚幻四提供了免费的源代码访问权限;丰富的 C++ API 接口对于本文算法的实现十分友好;高保真的图像和配备的控制功能,更利于演示算法的效果。

2.2.2 引擎框架

仿真开发的基础框架包含 GameSystem, Graphics 和 Gameplay 3 个功能模块,类图如图 1 所示。

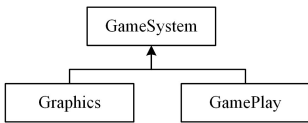


图 1 游戏开发基础类图

Fig. 1 Game development basic class diagram

其中,GameSystem 类负责游戏系统的控制,主要功能包括获取硬件信息和调用硬件等。Graphics 模块负责游戏画面的渲染,画面渲染的技术库包括 DirectX 各版本、OpenGL 等。Gameplay 模块是游戏的运行逻辑。对于使用游戏引擎进行开发来说,GameSystem 模块和 Graphics 模块一般均已被引擎实现并封装,用户只需在 Gameplay 模块下进行设计和开发。而在使用虚幻四进行开发时,也必须在虚幻四已有的 Gameplay 模块的框架下进行。其 Gameplay 的类图依据虚幻四官方文档,如图 2 所示。

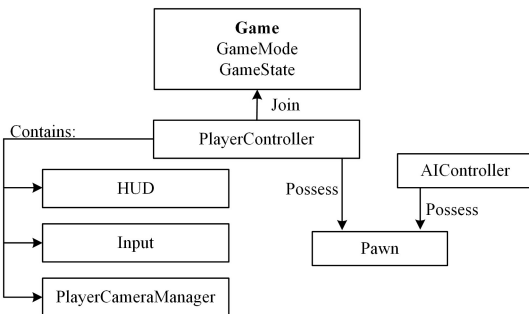


图 2 虚幻四的 Gameplay 类图

Fig. 2 Gameplay class diagram of Unreal IV

GameMode:游戏模式,这是一个游戏的玩法逻辑,如游戏规则和获胜条件。

GameState:记录游戏的状态,包括游戏得分、玩家状态等。

HUD:游戏中显示的二维信息,比如菜单、血条等。

Input:控制玩家输入的游戏交互指令。

PlayerCameraManager:游戏中的摄像机,即玩家的眼睛。摄像机捕捉到三维场景中的画面,再加上 HUD 就是玩家屏幕显示的内容。该模块负责管理摄像机的行为。

Pawn:Pawn 之下其实还有一个父类 Actor,Actor 是场景中的一切可被显示物体的基类,而 Pawn 则是这些 Actor 中可被控制器控制的物体基类,用以完成特定的任务。

PlayerController:玩家控制器,用于玩家交互,控制场景中的角色。交互的方式则是通过 HUD 和 PlayerCameraManager 的显示输出,以及 Input 的控制输入完成的。

AIController:控制场景中的非玩家 Pawn,相当于 AI。

以上就是虚幻四引擎框架中的主要类。在虚幻四引擎中的开发,就是在这个框架中对各个功能类进行实现、派生或功能扩展。

3 无人机回避策略

3.1 三维空间下的速度障碍物理论

无人机避开障碍物的核心问题是需要计算出哪些速度是当前时刻无人机在现实物理规则约束下可以达到,且不会导致和障碍物发生碰撞。这些速度的集合就是可达避障速度空间。可达避障速度空间原本是对二维速度空间的描述,在本文中我们将其泛化于三维空间。概括来讲,对于理论中涉及的二维图形、二维变量增加 z 轴上的第三维度,并替换为三维体积、三维变量,同时对这一理论计算过程中的定义和算法做出对应泛化。

在具体分析可达避障速度空间理论之前,需要先引入速度障碍物的概念。不失一般性,将无人机和障碍物统一用球体来表示。使用球体来表示无人机和障碍物并不是一个十分严格的设置,因为一般的三维物体均可以使用一个或多个球体进行表示,以此避免空间量化带来的误差^[27]。此外,无人机获取障碍物位置、速度信息以及计算回避速度必须是实时的。无人机设备上获取障碍物信息的传感器的敏感度和准确性则不在本文讨论范围中。因此,假设接近无人机一定范围内的障碍物的瞬时速度和位置是已知或是可被无人机实时测量的。

如图 3 所示,某一时刻的两个球形物体 A 和 B,其速度分别为 v_A 和 v_B 。球 A 代表无人机,球 B 代表障碍物。

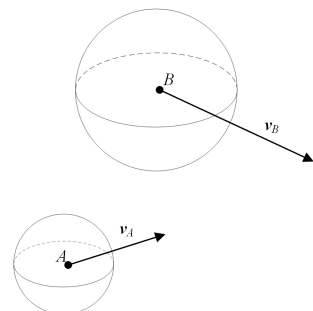


图 3 无人机 A 和障碍物 B 的运动方向示意图

Fig. 3 Diagram of motion direction of UAV A and obstacle B

为了计算 B 相对于 A 的速度障碍物,我们首先要将 B 的大小、速度信息映射到 A 上。方法是将球 A 半径减小至球心,如点 A 位于球 A 的球心位置,并将球 B 的半径扩大至球 A 的半径,可得出相对物体大小,如图 4 所示。

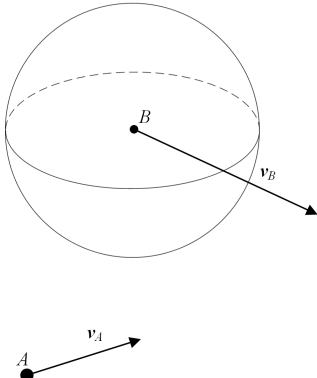


图 4 相对大小变换后的点 A 和球 B 示意图

Fig. 4 Diagram of point A and ball B after translating relative size

用 A 的速度减去 B 的速度,得到相对速度。每个运动对象的状态由其位置和附加到其中心的速度矢量表示。将碰撞锥(Collision Conic) CC_{AB} 定义为 A 与 B 之间的碰撞相对速度集合:

$$CC_{AB} = \{v_{AB} | \lambda_{AB} \cap \hat{B} \neq \emptyset\} \quad (1)$$

其中, v_{AB} 是 A 相对于 B 的相对速度,即 $v_{AB} = v_A - v_B$; λ_{AB} 是 v_{AB} 的延长线。

CC_{AB} 相对于障碍物而言,表现为点到圆形障碍物的切线夹住的扇形区域。而对于障碍物为球体的三维情景下的无人机而言, CC_{AB} 是一个圆锥体区域。该圆锥体是顶点在 A 的锥体区域,由从 A 到 B 的切面界定,如图 5 所示。任何位于锥体 CC_{AB} 之间的相对速度都会导致 A 和 B 之间的碰撞。显然,按照当前两者的速度, $\lambda_{AB} \cap B \neq \emptyset$,将发生碰撞。反之,只要障碍物 B 保持其当前形状和速度, CC_{AB} 以外的任何相对速度都可以保证无碰撞。

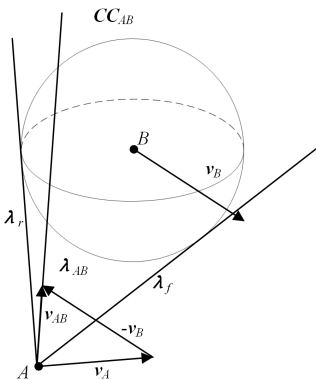


图 5 相对速度 v_{AB} 和碰撞锥 CC_{AB} 的示意图

Fig. 5 Diagram of relative velocity v_{AB} and collision cone CC_{AB}

一个碰撞锥确定于特定的某个无人机-障碍物对。若要考虑多个障碍,则需要在 A 的绝对速度上建立一个等效的可对比的表示方法。为此,将 B 的速度 v_B 加到 CC_{AB} 中的每个速度上,或者等效地通过将碰撞锥 CC_{AB} 沿 v_B 平移

来表示,如图 6 所示。

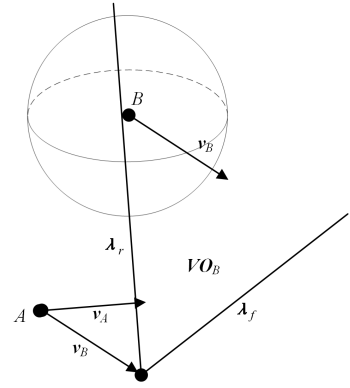


图 6 速度障碍物 VO_B 的示意图

Fig. 6 Diagram of speed obstacle VO_B

若障碍物是静止的,即 v_B 大小为 0,则碰撞锥 CC_{AB} 不发生平移。速度障碍物 VO 定义为:

$$VO = CC_{AB} \oplus v_B \quad (2)$$

其中, \oplus 是 Minkowski 向量和运算符。关于两个三维向量集合 A, B 的 Minkowski 向量和运算定义为:

$$A \oplus B = \{(x_1 + x_2, y_1 + y_2, z_1 + z_2) | (x_1, y_1, z_1) \in A, (x_2, y_2, z_2) \in B\} \quad (3)$$

对于 \oplus 运算,举例如下:

$$A = \{(1, 2, 3), (0, 0, -1)\} \quad (4)$$

$$B = \{(1, 1, 1), (-1, -1, -1)\} \quad (5)$$

$$A \oplus B = \{(2, 3, 4), (1, 1, 0), (0, 1, 2), (-1, -1, -2)\} \quad (6)$$

式(2)中的 v_B 可看作只含单个三维向量的集合。

VO 将 A 的绝对速度划分为避免速度和碰撞速度。在 VO 内的速度为导致碰撞的速度,在 VO 外的速度为回避速度。 v_A 选择在 VO 之外的速度可以避免与 B 发生碰撞。 VO 边界上的速度将导致 A 正好与 B 擦肩而过。为了避开多个障碍,我们要考虑多个速度障碍物的并集:

$$VO = \bigcup_{i=1}^m VO_{B_i} \quad (7)$$

其中, m 是障碍物的数量。回避速度由速度障碍之外的速度 v_A 组成,如图 7 所示。

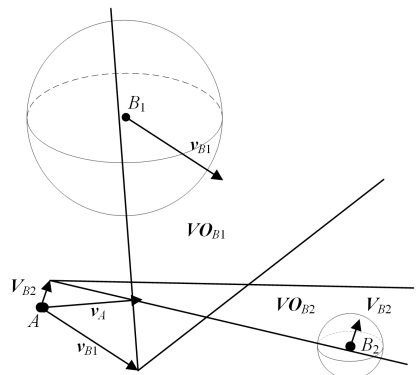


图 7 B_1 和 B_2 的速度障碍物示意图

Fig. 7 Diagram of speed obstacles for B_1 and B_2

再考虑多个无人机的情况:对于当前无人机来说,其他

无人机同样被视作障碍物。现存在无人机 A_1, A_2 和障碍物 B , 则对于无人机 A_1 来说, 障碍物 B 和无人机 A_2 均为需要避开的障碍物, 其速度障碍物示意图与图 7 一致。对于无人机 A_2 来说, 障碍物 B 和无人机 A_1 均为需要避开的障碍物, 如图 8 所示。

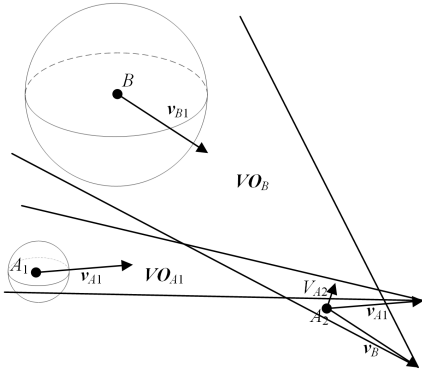


图 8 无人机 A_2 的速度障碍物 A_1 和 B 示意图

Fig. 8 Diagram of speed obstacles A_1 and B for UAV A_2

3.2 可达速度空间理论

可达速度空间理论通过无人机的制动器约束来计算在 t 时刻给定时间间隔 Δt 上无人机 A 可以达到的速度 (Reachable Velocity, RV), 其定义为:

$$RV(t+\Delta t) = \{v | v = v_A(t) \oplus \Delta t * a\} \quad (8)$$

其中, $v_A(t)$ 为 A 在 t 时刻的速度, a 为无人机的加速度。 $\|a\| \leq MA$, MA 是机器人的最大加速度大小。 $RV(t+\Delta t)$ 是由 A 的当前速度加上加速度乘时间来计算的, 如图 9 所示, 其中, $a * \Delta t$ 的范围是以当前速度 v_A 矢量的终点为圆心、 $a * \Delta t$ 的大小为半径的球体区域。

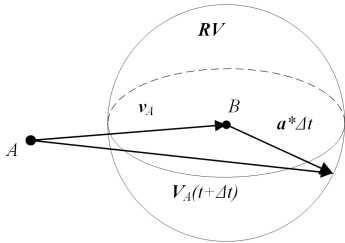


图 9 原可达速度空间 (RV) 示意图

Fig. 9 Diagram of original reachable velocity space (RV)

若每次都选取比 v_A 更大的速度, 则机器人速度会达到无限大, 明显与实际情况不符, 实际中机器人的加速是不可超过速度最大值的。因此修正过后的可达速度空间理论同时考虑了最大加速度约束和最大速度约束, 将 t 时刻在时间间隔 Δt 上可达到的速度 $RV(t+\Delta t)$ 的集合定义为:

$$RV(t+\Delta t) = \{v | v = v_A(t) \oplus \Delta t * a \& \& |v| \leq MV\} \quad (9)$$

相比式 (8), 式 (9) 增加了约束条件。 MV 是无人机的最大速度大小, v 的模长必须小于或等于无人机的最大速度大小。 $RV(t+\Delta t)$ 是由 A 的当前速度加上加速度乘时间来计算的, 且被约束在最大速度内。如图 10 所示, 最大速度的范围是以当前 A 点为圆心、 MV 为半径的球体区域, 因此 RV 的范围则是这两个球体相交的部分。

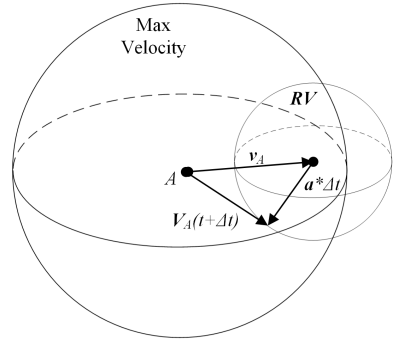


图 10 带约束的可达速度空间 (RV) 示意图

Fig. 10 Diagram of restricted reachable velocity space (RV)

3.3 可达到的回避速度

基于速度障碍物 (VO) 和带约束的可达速度集合 (RV), 可达到的回避速度集合 (Reachable Avoidance Velocity, RAV) 定义为可达速度集合与速度障碍物之间的差:

$$RAV(t+\Delta t) = RV(t+\Delta t) \ominus VO(t) \quad (10)$$

其中, \ominus 表示集合差的运算。两个三维向量集合 A 和 B 的集合差的运算定义为:

$$A \ominus B = \{(x, y, z) | (x, y, z) \in A \& \& (x, y, z) \notin B\} \quad (11)$$

对于 \ominus 运算, 举例如下:

$$A = \{(1, 2, 3), (1, 1, 1)\} \quad (12)$$

$$B = \{(1, 1, 1), (-1, -4, -3)\} \quad (13)$$

$$A \ominus B = \{(1, 2, 3)\} \quad (14)$$

选择 RAV 中的任意速度均可避开障碍物 B 。图 11 表示由两个不相交的闭合子集组成的集合 RAV , 即其中灰色区域为两个球体相交的部分 RV 减去与锥体 VO_B 重叠的部分。对于多个障碍物, RAV 可以包含多个不相交的子集。

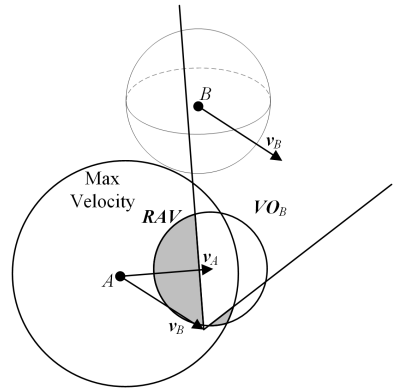


图 11 可达到的回避速度 (RAV) 示意图

Fig. 11 Diagram of reachable avoidance velocity (RAV)

综合上文, 可达避障速度空间的集合就是先获取当前时刻在现实条件约束下可达到的速度集合, 再减去会导致与障碍物碰撞的速度集合。利用此方法, 便得出无人机在当前时刻的附近障碍物的 RAV 集合。只要是在 RAV 中选取的速度, 都可以避开与障碍物的碰撞, 且都是现实中能达到的速度。

3.4 回避策略的选择

考虑现实因素, 动态环境下无人机无法在初始时刻获取

到路径附近的全部障碍物的速度和位置,因此设定无人机会获取到接近自己一定范围内的障碍物的位置和速度。无人机运动到终点的轨迹由一系列避开当前障碍物的操作产生。本文采用启发式算法,对每一个需要重新规划速度的时间节点进行独立计算,并更新无人机速度。

启发式的设计可以满足一系列优先目标,例如将无人机的生存作为首要目标,抵达目的地的时间长短作为次要目标等。对于所有包含在 **RAV** 集合中的速度,选择速度方向指向目标点的速度,可以确保无人机能够到达目的地。在目标的大致方向上选择最高可行速度,可能减少运动过程花费的时间。从 **RAV** 的适当子集中选择速度,可以确保所需的轨迹结构(从障碍物前部避开或后部避开)。需要注意的是,启发式算法不能保证任何时候都可以实现目标,启发式算法的目的是找到一个对于给定的需求而言局部最优的避障策略。

3.4.1 径直向目标策略

避开径直向目标策略(To Goal, TG),选择沿无人机-目标连线上的最高回避速度,使得无人机运动轨迹直线抵达目标。如图 12 所示,灰色区域表示 **RAV**,机器人在 **RAV** 区域中选取了与 Goal 目标连线的最大速度。具体策略为:沿直线抵达目标,遇见障碍物则减速,待障碍物经过再恢复速度继续前进。

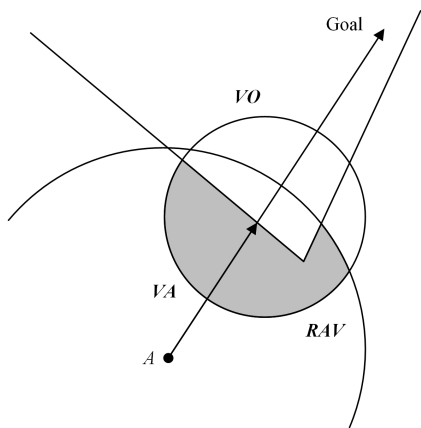


图 12 径直向目标(TG)策略示意图

Fig. 12 Diagram of TG strategy

在计算机中进行实现时,可以让速度沿着射线 $A \rightarrow \text{Goal}$ 进行递增。当速度在 **VO** 范围内时不予以赋值,不作为输出的最终速度;速度不在 **RV** 中时终止,最后一个速度就是目的速度,即是该无人机能够径直抵达目的地的最大可达避障速度。

3.4.2 最大速度策略

最大速度策略(Maximum Velocity, MV),选择从无人机到目标连线的限定角度 α 内的最大速度。无人机虽然没有直接瞄准目标,但是能够以较大的速度在可接受的偏差方向上高速移动。如图 13 所示,灰色区域表示 **RAV**,MV 策略选取了在射线 i, j 的夹角 α 内, **RAV** 中的最大速度 v_A ,且要求与射线 $A \rightarrow \text{Goal}$ 的夹角尽量小。

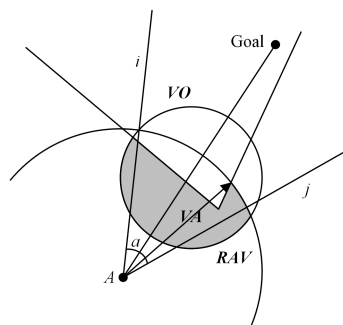


图 13 最大速度(MV)策略示意图

Fig. 13 Diagram of maximum velocity(MV)strategy

在计算机中进行实现时,可以遍历 **RAV** 集合中对于所有方向上的最大速度,同时保持其在角度 α 的范围内,优先选择与 $A \rightarrow \text{Goal}$ 连线角度更小即偏离目的地程度更小的速度。

以上两种策略实际上是分别对运动规划系统的两个方面即路径规划和速度规划做出了侧重处理。TG 策略着重考虑路径的确定性,对速度进行规划约束;MV 策略着重考虑保持运动过程的高速,侧重于对路径进行规划。这两种算法从设计上来说各有侧重,适用于不同的场景。

4 实验模拟

本节在虚幻四引擎上给予实现并验证,虚幻四引擎上采用 C++ 编程和蓝图编程相结合的开发方式。

4.1 实验数据

实验使用的距离单位均为虚幻四虚拟空间中的笛卡尔坐标系单位长度。使用如下数据捕捉轨迹并绘制无人机运动轨迹图。

目的地:坐标(2390,0,920);

无人机:坐标(-300,0,490),最大速度 10,最大加速度 3,半径 50,检测距离 400;

障碍物 1:坐标(450,-290,820),速度(0,5,-2.5),半径 50;

障碍物 2:坐标(1200,650,370),速度(3,-3,2),半径 50。

时间的测量用到多组不同的障碍物位置-速度对,见表 1。除此之外,还增设第二台无人机来验证多无人机情况下的可行性。第二台无人机的数据如下:坐标(340,1130,570),最大速度 12,最大加速度 4,其目的地(1200,1190,670)。

4.2 实验过程

对于以上数据设置,通过无人机运动轨迹和抵达目的地所用时间来衡量比较选择不同策略的优劣,具体方法如下。

(1)轨迹捕捉的方法:利用引擎自带的平行视角,分别对演示时的正上方和正右方视角每 10 帧进行一次截图,再使用 photoshop 工具将截图叠加合并起来,以显示场景中物体的运动轨迹和速度大小。图 14(a)中,红色球体轨迹表示无人机,黄色球体轨迹表示障碍物,绿色方块表示目的地。可通过轨迹上球体的疏密程度判断速度大小,越疏散的部分速度越大,越紧密甚至重叠的地方速度越小。

(2)计时方式:在无人机初始化时,获取当前时刻并存储。在无人机抵达目的地,即与图 14(a)中绿色方块撞击时,再次获取当前时刻,与初始化时存储的时刻相减便得到该路径所用的时间。

(3)判断抵达/坠毁的方式:无人机与障碍物、目的地均包含碰撞体组件,并编写相应触发事件函数。当机器人与障碍物发生碰撞时,机器人销毁(即从屏幕上消失)并在屏幕上打印提示文字“Collision”;当机器人与目的地相撞时,机器人消失并在屏幕上输出耗时,精确到 0.01 s。

4.3 结果展示

TG 策略的演示轨迹如图 14 所示。

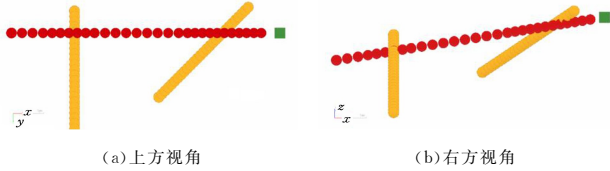


图 14 TG 策略下的运动轨迹

Fig. 14 Movement trajectory using TG strategy

MV 策略的演示轨迹如图 15 所示。

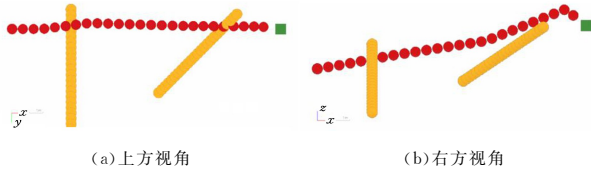


图 15 MV 策略下的运动轨迹

Fig. 15 Movement trajectory using MV strategy

TG 和 MV 策略在不同障碍物速度和位置时的耗时如表 1 所列。

表 1 TG 和 MV 策略的耗时

Table1 Time consumption of TG and MV strategies

组别	障碍物 1	障碍物 2	TG 耗时/s	MV 耗时/s
	位置速度	位置速度		
1	(450, -290, 820) (0, 5, -2.5)	(1200, 650, 370) (3, -3, 2)	5.22	4.61
2	(460, -10, 1100) (0, 0, -5)	(1200, 650, 370) (1.5, -4, 2)	6.41	4.53
3	(460, -390, 300) (0, 5, 5)	(1670, 650, 840) (-2, -4, 0)	4.76	4.49
4	(890, 690, 360) (-1, -7, 3)	(1730, 650, 840) (-2, -4, 0)	4.73	4.52
5	(80, 640, 530) (7, -5, 1)	(80, -640, 530) (7, 5, 1)	4.95	4.53

多无人机验证组中,两架无人机也顺利到达各自目的地。

4.4 结果分析

由图 14 可以看出,在 TG 策略下,无人机轨迹笔直且指向目的地,先以最大速度行驶,对两个障碍物均是在其靠近后先减速,等到障碍物远离后再加速,恢复到最大速度。

由图 15(a)可以看出,在 MV 策略下,机器人向左移动以回避第一个障碍物;由图 15(b)可以看出,机器人向上移动以回避第二个障碍物。由于轨迹全程稀疏,可得出机器人速度大小在全过程中均维持在一个数值较大的范围内。

由表 1 可看出, MV 策略下的耗时均少于 TG 策略下的耗时。结合轨迹和耗时可以得出, TG 策略的优点是轨迹稳定可预测,但平均速度没有 MV 策略快,且在路径上有静态障碍物时该策略无法工作,需要暂时调用其他策略; MV 策略的优点是平均速度快,但是运动轨迹曲折。因此这两者适用于不同的场景,例如在限定航道范围或运送大重量货物的场景下适合用 TG 策略,而追求速度的竞速场景则适合采用 MV 策略。

结束语 本文针对空中无人机面对的多个移动障碍物,研究了无人机动态避障的算法设计。根据无人机和障碍物的位置、速度信息,建立速度障碍物区域。所给出的速度障碍物区域用于预测当前无人机与障碍物是否会发生碰撞,以及根据该区域界定哪些速度可以回避碰撞。根据无人机的制动器最大加速度和最大速度的限制建立无人机在当前时刻的可达速度空间,用于约束机器人回避障碍物的速度,使其在现实约束下可达。用可达速度空间减去速度障碍物区域,得到无人机可达的避障速度空间。依据现实中可能出现的情境提出两种避障策略,在无人机的可达速度空间中选取不同情境下的最优避障速度。通过在虚幻四上进行实验模拟,验证了无人机避障理论及其实现的运动规划系统的正确性和不同避障策略针对不同需求的适用性。

后续工作中,将对速度障碍物的有效计算范围进行约束;对于探测多障碍物,依据其将发生碰撞的先后次序进行优先级排序,以实现更高效的运动系统;还将考虑环境因素,如风向等,对无人机避障行为的影响和对应校正。

参考文献

- [1] SUN D S, WANG Y. Robot control technology [M]. Beijing: Machinery Industry Press, 1997.
- [2] WANG Z, WEN M, DANG S, et al. Trajectory design and resource allocation for UAV energy minimization in a rotary-wing UAV-enabled WPCN[J]. AEJ-Alexandria Engineering Journal, 2020, 60(1): 1787-1796.
- [3] HAYWARD V, AUBRY S, FOISY A, et al. Efficient collision prediction among many moving obstacles[J]. International Journal of Robotics Research, 1995, 14(2): 129-143.
- [4] XI Y G, ZHANG C G. Rolling path planning of mobile robot in a kind of dynamic uncertain environment [J]. Acta Automatica Sinica, 2002, 28(2): 161-175.
- [5] REIF J, SHARIR M. Motion planning in the presence of moving obstacles [C] // Symposium on Foundations of Computer Science. 2008.
- [6] ERDMANN M, LOZANO-PEREZ T. On multiple moving objects[J]. Massachusetts Institute of Technology, 1986, 2(4): 477-521.
- [7] FUJIMURA K, SAMET H. A hierarchical strategy for path planning among moving obstacles[J]. IEEE Trans. Robotics Automation, 1989, 5(1): 61-69.
- [8] KANT K, ZUCKER S W. Toward Efficient Trajectory Planning: The Path-Velocity Decomposition[J]. International Journal of Robotics Research, 1986, 5(3): 72-89.
- [9] LEE B H, MEMBER, IEEE, et al. Collision-Free Motion Plan-

- ning of Two Robots[J]. IEEE Transactions on Systems Man & Cybernetics, 1987, 17(1): 21-32.
- [10] FRAICHARD T. Dynamic trajectory planning with dynamic constraints: A 'state-time space' approach[C]// IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE, 1993.
- [11] FRAICHARD T, LAUGIER C. Path-velocity decomposition revisited and applied to dynamic trajectory planning[C]// IEEE International Conference on Robotics & Automation. IEEE, 1993.
- [12] FUJIMURA K. Time-minimum routes in time-dependent networks[J]. IEEE Transactions on Robotics & Automation, 1995, 11(3): 343-351.
- [13] FUJIMURA K, SAMET H. Time-minimal paths among moving obstacles[C]// IEEE International Conference on Robotics & Automation. IEEE, 1989.
- [14] FUJIMURA K. Motion Planning Amid Transient Obstacles[J]. The International Journal of Robotics Research, 1994, 13(5): 395-407.
- [15] SANBORN J. A model of reaction for planning in dynamic environments[J]. Artificial Intelligence in Engineering, 1988, 3(2): 95-102.
- [16] TSUBOUCHI T, ARIMOTO S. Behavior of a mobile robot navigated by an "iterated forecast and planning" scheme in the presence of multiple moving obstacles[C]// IEEE International Conference on Robotics & Automation. IEEE, 1994.
- [17] FIORINI P, SHILLER Z. Motion planning in dynamic environments using the relative velocity paradigm[C]// International Conference on Robotics and Automation. IEEE, 1993.
- [18] FIORINI P, SHILLER Z. Robot Motion Planning in Dynamic Environments[M]. Springer, 1996.
- [19] FIORINI P, SHILLER Z. Motion Planning in Dynamic Environments Using Velocity Obstacles[J]. International Journal of Robotics Research, 1998, 17(7): 760-772.
- [20] LI L, YE T, TAN M, et al. Presentstate and future development of mobile robot technology research[J]. ROBOT, 2002, 24(5): 475-480.
- [21] ZHUANG X D, MENG Q C, YIN B, et al. A method of robot's path searching in dynamic environment based on fuzzy concept [J]. ROBOT, 2001, 23(5): 397-399, 458.
- [22] LI R F, LI W Z. Path Planning for Mobile Robot Based on Multi-sensor Information [J]. Mechatronics, 2002(4): 21-24.
- [23] YUNG N H C, YE C. An intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning [J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 1999, 29(2): 314-321.
- [24] WANG J, HUANG X H. Obstacle-avoidance control of the two-wheeled cart based on evolving neural network[J]. ROBOT, 1996, 18(5): 292-297.
- [25] LIU C, WANG H L, YAO P, et al. Modeling and analysis of dynamic collision region for UAV avoiding aerial intruders [J]. Journal of Beijing University of Aeronautics and Astronautics, 2015, 41(7): 1231-1238.
- [26] LÓPEZ A M, VILLALONGA G, SELLART L, et al. Training my car to see using virtual worlds-ScienceDirect[J]. Image and Vision Computing, 2017, 68: 102-118.
- [27] O'ROURKE J, BADLER N. Decomposition of three-dimensional objects into spheres [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979, 1: 295-305.



LUO Xiong-feng, born in 1998, is a member of China Computer Federation. His main research interests include UAV and Internet of things.



ZHAI Xiang-ping, born in 1984, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include the area of UAV, Internet of things, wireless networks, resource optimization, and spatial analysis.

(责任编辑:柯颖)