

[引用格式] 刘国顺, 郭威, 兰彦军, 等. 基于 IDVD 的 AUV 实时运动规划算法 [J]. 水下无人系统学报, 2025, 33(3): 473-483.

基于 IDVD 的 AUV 实时运动规划算法

刘国顺, 郭威*, 兰彦军, 符一凡

(中国科学院深海科学与工程研究所, 海南 三亚, 572000)

摘要: 为了提升自主水下航行器(AUV)的智能化程度, 文中提出一种基于逆动力学虚拟域(IDVD)方法的实时运动规划算法, 确保其在未知障碍物环境中安全航行。考虑到 AUV 自身计算资源有限, 为保证较高的计算效率并实现实时规划, 算法采用分层结构框架: 首先通过路径规划算法搜索全局安全路径, 然后在前视声呐的感知范围内进行路径优化, 生成安全可行的轨迹, 针对欠驱动 AUV 的运动学约束, 全局路径搜索采用混合 A*算法来寻找安全路径, 并构建非线性优化问题以提高路径的平滑性和安全性; 利用 IDVD 方法获得可行的 AUV 速度和加速度轨迹, 并作为参考输入引导 AUV 航行。为了验证所提出的运动规划方法, 对“海缸二号”AUV 进行了仿真和实验, 结果表明文中方法能够在未知复杂环境中为 AUV 实现高效的在线轨迹规划。

关键词: 自主水下航行器; 运动规划; 逆动力学虚拟域; 轨迹优化

中图分类号: TJ630.34; U674.941

文献标识码: A

文章编号: 2096-3920(2025)03-0473-11

DOI: 10.11993/j.issn.2096-3920.2025-0033

A Real-time Motion Planning Algorithm for AUV based on IDVD Method

LIU Guoshun, GUO Wei*, LAN Yanjun, FU Yifan

(Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya 572000, China)

Abstract: To enhance the intelligence of autonomous undersea vehicles(AUVs), this paper proposed a real-time motion planning algorithm based on the inverse dynamic virtual domain(IDVD) method, ensuring safe navigation in unknown environments with obstacles. In view of the limited computational resources of AUVs, a hierarchical framework was adopted to guarantee high computational efficiency for real-time planning. First, a safe global path was generated using a path planning algorithm, followed by path optimization within the sensing range of the forward-looking sonar to produce safe and feasible trajectories. Specially, for the kinematic constraints of underactuated AUVs, the hybrid A* algorithm was employed for searching safe paths based on global path search. Subsequently, a nonlinear optimization problem was formulated to enhance path smoothness and safety. The IDVD method was then applied to derive feasible velocity and acceleration trajectories of AUVs, which served as reference inputs to guide the AUVs' navigation. Simulations and experiments on the “Stingray-II” AUV were conducted. The results validate that the proposed method is capable of efficient online trajectory planning for AUVs in unknown complex environments.

Keywords: autonomous undersea vehicle; motion planning; inverse dynamic virtual domain; trajectory optimization

收稿日期: 2025-02-26; 修回日期: 2025-03-24; 录用日期: 2025-04-17.

作者简介: 刘国顺(1991-), 男, 博士, 工程师, 主要研究方向为水下机器人智能控制.

* 通信作者简介: 郭威(1971-), 男, 研究员, 博士生导师, 主要研究方向为水下机器人总体及相关控制技术.

OPEN ACCESS

0 引言

自主水下航行器(autonomous undersea vehicle, AUV)已成为海洋工程领域不可或缺的重要工具,广泛应用于水下探测、管道安全检查以及海洋环境监测等任务。作为 AUV 执行任务的核心环节,运动规划直接决定了其路径的安全性与可行性,进而影响任务成败。因此,运动规划方法的先进程度在很大程度上反映了 AUV 的智能化水平。然而,海洋环境通常复杂多变,不仅存在未知的障碍物,还受到洋流等动态因素的影响。此外, AUV 的续航时间有限,前视声呐的传感范围也受到约束。这些挑战使得 AUV 在执行任务过程中必须能够实时进行在线动态规划或轨迹调整,以确保任务的安全性和效率。因此,开发高效、智能的运动规划算法对于提升 AUV 在复杂海洋环境中的自主性和可靠性至关重要。

运动规划算法主要包括全局路径和局部路径规划方法。基于图搜索的算法以及进化算法均可用于全局路径搜索。在图搜索算法中,如 A*或 Dijkstra 等,可快速搜索出全局最优路径,但由于此类方法依赖于地图空间划分,仅适用于低维度、小尺度地图。进化算法包括遗传算法^[1]、粒子群算法^[2]等,此类方法适用于复杂环境,但其计算效率低,且容易收敛到局部最优解。局部规划方法包括基于采样的方法和人工势场法等。基于采样的方法如快速搜索随机树(rapid exploration random tree, RRT)法通过在空间中快速撒点采样,从而在动态和高维空间中进行路径搜索,但无法保证路径平滑,其结果需要进一步优化。人工势场法将环境构造为吸力和斥力的势力场,障碍物产生斥力,目标点产生吸力,但该方法仅适用于环境简单场景,否则无法保证路径的合理性。

动态路径规划的主要挑战在于如何在有限的时间和计算资源下找到一条安全的路径。全局规划方法和局部规划方法均可用于动态路径规划,前提是必须具备较高的计算效率。基于网格^[3]和基于采样^[4]的路径规划方法因其高效的计算性能,常被应用于动态路径规划。这些方法能够在每个规划周期内,根据先前的路径迭代生成一条新的路径。然而,这些方法生成的参考路径往往难以

被 AUV 直接遵循,因为它们未充分考虑 AUV 的动态特性,如运动学和动力学约束。为了解决这一问题, Yakimenko^[5]提出了逆动力学虚拟域(inverse dynamic virtual domain, IDVD)方法,并将其应用于 AUV 的路径规划中。IDVD 方法的核心在于每次重新规划路径时会摒弃先前的轨迹,并基于最新的环境地图实时计算新的轨迹。这种方法能够充分考虑 AUV 的动态特性,从而生成更符合其实际运动能力的路径。然而, IDVD 方法生成的轨迹通常仅适用于短程机动,在复杂环境(如长距离、多障碍物场景)中可能难以满足规划需求。

为了提高路径规划的计算效率和可行性,可以将复杂的规划问题分解为多个连续的子规划问题,并采用分层求解的策略^[6]。在分层规划框架中,上层规划器负责使用诸如 A*或 RRT 等方法,生成一条粗略但安全的全局路径;下层规划器则利用最优控制方对路径进行精细化优化,以生成动态可行的轨迹。为了进一步降低计算负担,可以通过移除障碍区域或提取有效空间(如构建安全走廊^[7]或消除冗余障碍^[8])来简化规划环境。然而,尽管这些方法能够在一定程度上减少计算复杂度,但由于优化变量的数量保持不变,实现实时规划仍然面临较大挑战。为了进一步提升计算效率, Zhuang 等^[9]和 Mahmoud Zadeh 等^[10]在优化过程中引入了微分平坦性理论,显著减少了优化变量的数量,从而加速了规划过程;然而微分平坦性的引入会导致规划的速度具有奇异性。

基于以上分析,文中提出一种融合混合 A*和 IDVD 方法的分层运动规划方法,通过滚动窗口规划机制和 IDVD 方法实现 AUV 运动规划的实时计算,通过与其他方法对比和实物测试,证明文中提出方法的有效性。

1 算法概述

为了提高运动规划的计算效率,采用层级结构规划器实现未知环境下轨迹规划。运动规划器上层为全局路径规划,一般由混合 A*^[11]或 RRT^[12]方法搜索一条从起点到目标点的最短路径;规划器底层为局部轨迹优化,根据任务需求和约束条件构建非线性优化问题,通过迭代优化的方式获得局部最优轨迹。其完整的框架结构如图 1 所示。

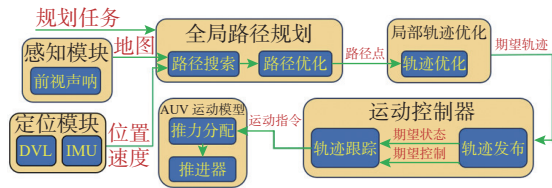


图 1 运动规划方法框架

Fig. 1 Framework of motion planning

运动规划的输入为规划任务、环境地图和里程计信息。其中, 规划任务为目标点的位姿; AUV 通过自身搭载的前视声呐获取周围障碍物信息, 构建增量式地图^[13]; 通过多普勒计程仪(Doppler velocity log, DVL)和惯性测量单元(inertial measurement unit, IMU)进行卡尔曼滤波, 实现航位推算, 获取 AUV 瞬时位置和速度^[14]。

文献 [15-17] 提出使用连接起点和目标点的直线路径作为全局路径, 但该路径存在与障碍物发生碰撞的风险。虽然通过局部轨迹优化可以使路径远离障碍物, 但轨迹优化是一类非线性优化问题, 优化结果依赖于初始路径。因此在全局路径规划阶段, 根据 AUV 非全向运动的特性, 选用混合 A* 算法搜索出与周围环境无碰撞的路径。由于搜索出的路径存在不平滑且距离障碍物较近的路段, 因此进一步通过数值优化方法保证所有航位点安全平滑, 同时优化后的路径能够减少局部轨迹优化的计算负担。

在局部轨迹优化阶段, 首先构造轨迹优化问题, 根据感知范围内的路径点确定优化问题的边界条件, 结合 AUV 的动力学约束, 局部感知范围内的障碍物约束构造最优控制问题。非线性的安全约束条件导致优化问题计算复杂, 为降低该优化问题的复杂程度, 将非线性约束条件转化为惩罚函数扩充到优化问题的目标函数中, 进而将轨迹优化问题转化为只包含线性等式和不等式约束的非线性优化问题。借助微分平坦特性, 采用 IDVD 方法实现局部轨迹优化的高效计算。

局部轨迹优化的输出轨迹作为运动控制器的前馈输入, 在其内部以时间戳方式获取瞬时期望状态和控制量, 通过底层的比例-积分-微分(proportional-integral-derivative, PID)控制器将期望状态量和控制输入转化为 AUV 的运动指令, 通过推力分配计算出各推进器的执行推力。

针对未知环境中的运动规划, 借鉴模型预测控制^[18](model predictive control, MPC)的思想, 在规划框架中引入滚动窗口规划机制, 在当前感知范围内进行运动规划。当 AUV 周围环境发生变化时, 该规划机制能够及时调整轨迹, 以保证 AUV 安全航行。动态规划机制如图 2 所示, 主要参数包括局部轨迹的规划范围 R_p 和执行范围 R_e 。其中: R_p 即为感知传感器(前视声呐)的测量范围; 而 R_e 的大小取决于 AUV 轨迹跟踪性能, 一般定为一个低于 R_p 的较小值; $X_i, i = 0, 1, \dots, \text{goal}$ 表示路径长度的采样点。每当 AUV 的轨迹执行长度超过 R_e 时, 运动规划器将触发轨迹重规划策略, 计算 AUV 当前位置到目标点的轨迹。需要注意的是, 轨迹规划的计算时间一定要明显低于轨迹执行范围所用的时间, 否则将会导致 AUV 在执行完当前轨迹时难以稳定跟踪新规划的轨迹。因此, AUV 的运动规划必须实现高效计算。为了提高运动规划的计算效率, 在全局路径规划阶段, 假定感知范围以外的区域为安全区域, 由于是未知环境的运动规划, 这种假设符合该场景的应用要求。

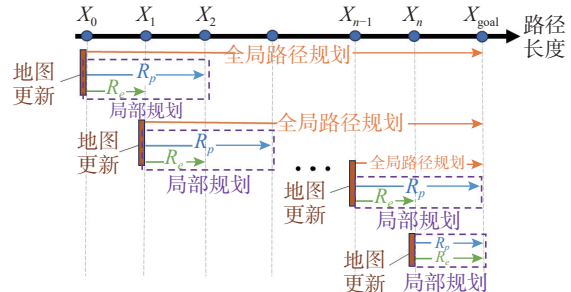


图 2 滚动窗口规划机制

Fig. 2 Receding horizon planning strategy

2 全局路径规划

2.1 混合 A* 算法

混合 A* 算法首次提出于无人驾驶的路径规划应用领域, 是一种基于图搜索的方法。与其他图搜索方法不同, 在进行空间扩展时, 混合 A* 考虑了无人驾驶汽车的运动学约束。由于搜索方法和 Dubins 曲线^[19]均考虑了车辆的运动学特性, 其路径更易于跟踪执行。其主要流程包括构造运动组元、运动组元空间扩展以及 Dubins 曲线扩展。

假定控制变量 $u(t)$ 恒定不变, 通过对运动学方程(式(1))前向积分, 获得符合运动学约束的路径。

$$\begin{cases} \dot{x}(t) = u(t)\cos(\varphi(t)) \\ \dot{y}(t) = u(t)\sin(\varphi(t)) \\ \dot{z}(t) = w(t) \\ \dot{\varphi}(t) = r(t) \end{cases} \quad (1)$$

式中: $u(t)$ 为航行速度; $\varphi(t)$ 为偏航角度; $w(t)$ 为垂向速度; $r(t)$ 为偏航角速度。

通过对控制变量 $\varphi(t)$ 离散化等分, 依次对时间 t 前向积分获得不同路径, 将所有路径组合形成运动组元, 文中将控制输入 $\varphi(t)$ 离散为三等份, 即 $[-\varphi_{\max}, 0, \varphi_{\max}]$, 生成运动组元如图 3(a) 所示, 以运动组元为扩展方式, 以任务起点为根节点进行空间扩展, 形成空间扩展搜索树, 见图 3(b)。

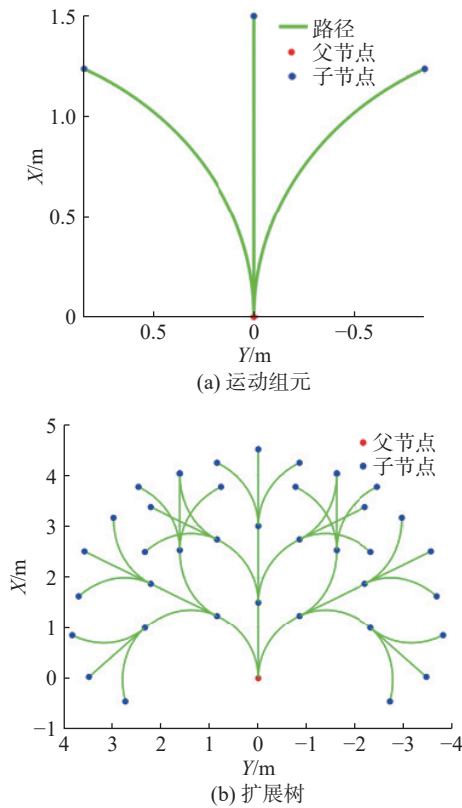


图 3 混合 A* 算法
Fig. 3 Hybrid A* algorithm

扩展过程中每个子节点被赋予不同的代价值, 记作 g 。每一个扩展节点到目标点的距离值记为 h , 称为启发式距离值, 采用 Dubins 曲线的距离长度进行计算。由此可以获得每个扩展节点在空间搜索树的总代价值 f , 即 $f=g+h$ 。每次扩展时优先对代价值最小的节点进行扩展, 直至搜索到目标点为止。

由于通过离散化的控制变量前向积分获得运动组元, 这意味着空间搜索树难以精确扩展到连续空间的坐标点。为解决这一问题, 引入 Dubins 曲线实现扩展节点到目标点的连接, 由此亦可提高路径规划的搜索效率。通过对 Dubins 曲线进行碰撞检测, 若与障碍物未发生碰撞, 则停止搜索, 此时安全路径由两部分组成, 分别是沿起点到该扩展节点的搜索树所包含的子节点和该扩展节点到目标点的 Dubins 曲线。

虽然 AUV 可在三维空间运动航行, 但文中仅考虑其在二维平面内的运动方式, 因此仅在水平方向内构建运动组元, 而轨迹的深度信息通过 Z 轴线性插值获得。首先将三维环境映射为二维地图, 并把地图离散为大小均匀的栅格以便于路径搜索。考虑到未知环境中的应用, 将感知区域以外的环境视为安全区域, 可以进一步降低环境地图的复杂性, 提高混合 A* 的搜索效率, 如图 4 所示。图中绿色图形表示障碍物, 包括感知范围内的已知障碍物和感知范围外的未知障碍物。由于路径规划仅考虑感知范围内的障碍物, 因此仅在感知范围内保证路径的安全性。

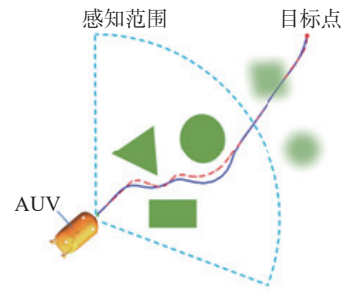


图 4 未知环境规划策略
Fig. 4 Planning strategy in unknown environment

2.2 路径优化

由于引入运动组元进行空间扩展, 导致路径存在不必要的转向, 形成不平滑路径, 此外 Dubins 曲线未考虑与障碍物的安全距离, 导致部分路径点距离障碍物较近, 存在碰撞的风险, 因此需要进一步对路径进行优化以保证更加平滑和安全, 同时减小局部轨迹优化的计算负担。

通过构建包含路径平滑和安全指标的非线性优化目标, 实现对路径点的优化。已知 1 组点集, 记为 $\bar{P}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$, $i = 1, \dots, n$, 作为优化变量, 并计

算出到每个航位点距离最近的障碍物坐标, 记为 $O_i = (x_i^o, y_i^o, z_i^o)$, $i = 1, \dots, n$, 为了优化航位点 $\bar{P}_i$, 可以构造出非线性优化目标函数, 即

$$\min J = \lambda_s J_s + \lambda_c J_c \quad (2)$$

式中: J_s, J_c 分别为平滑指标和安全指标; λ_s, λ_c 分别为对应的权重系数。 J_s 可以通过弹性带理论^[20]表示, 即

$$J_s = \sum_{i=2}^{n-1} \|(\bar{P}_{i+1} - \bar{P}_i) - (\bar{P}_i - \bar{P}_{i-1})\|_2^2 \quad (3)$$

其描述为相邻路径点突变程度的累计惩罚, 其值越小表示路径越平滑; 而安全指标可以表述为

$$J_c = \sum_{i=1}^n \begin{cases} (\|(\bar{P}_i - O_i)\|_2 - d_s)^2, & \|(\bar{P}_i - O_i)\|_2 \leq d_s \\ 0, & \text{其他} \end{cases} \quad (4)$$

可描述为路径点到其最近障碍物的距离中小于安全距离的累计惩罚, 其中 d_s 为安全距离阈值。

3 局部轨迹优化

轨迹优化的目标是在路径安全的前提下, 最小化航行时间和能量消耗的同时, 提供可行的速度和加速度, 以便 AUV 获得较好性能的航行轨迹。首先根据确定的边界条件, 构建局部轨迹优化问题; 为了实现高效计算, 采用微分平坦理论减少优化变量的数量; 通过惩罚函数方法降低优化问题的复杂度; 最后使用 IDVD 算法实现轨迹的时空联合优化。

3.1 轨迹优化问题模型

轨迹优化问题的数学模型可以表示如下。

优化问题的目标函数, 即最小化能量消耗和航行时间为

$$\min_{X, U, t_f} \int_0^{t_f} U(t)^T R U(t) dt + t_f \quad (5)$$

其中, X 和 U 分别为状态和控制向量; t_f 为航行时间。系统动力学约束为

$$\text{s.t. } \dot{X}(t) = f(X(t), U(t)) \quad (6)$$

为保证规划轨迹的安全性, 障碍物的避碰约束为

$$\|P(t) - O_i\|_2 \geq d_s, \quad i = 1, \dots, k \quad (7)$$

物理限制约束为

$$\|X(t)\| \leq \bar{X}, \|U(t)\| \leq \bar{U} \quad (8)$$

用于保证规划的速度和加速度大小均在 AUV 的可执行范围内, 优化问题的边界条件为

$$X(t_0) = X_0, X(t_f) = X_f \quad (9)$$

式中: $\int_{t_0}^{t_f} U(t)^T R U(t) dt$ 为能耗指标, 其中 t_f 为航行时间, $U(t) = [a_s(t), a_h(t), \alpha_y(t)]^T$ 为 AUV 的控制变量, R 为权重矩阵; AUV 的状态向量 $X(t) = [x(t), y(t), z(t), \varphi(t), u(t), w(t), r(t)]^T$; AUV 的空间位置 $P(t) = [x(t), y(t), z(t)]^T$; O_i 为局部感知内的障碍物坐标。

在每个规划周期内, 由航位推算得到 AUV 瞬时位置和速度作为起点的边界条件, 而并未明确目标点的边界条件。文中算法中, 目标点的位置坐标可从全局路径中选择, 一般选取感知边界附近的路径点作为局部轨迹规划的目标点。为了达到较好的运动性能, 使 AUV 以最大前向速度航行, 目标点的速度设置为

$$V(t_f) = u_{\max} \frac{P_{i+1} - P_{i-1}}{\|P_{i+1} - P_{i-1}\|_2} \quad (10)$$

式中: P_{i+1}, P_{i-1} 分别为目标坐标点前后相邻的路径点; $u_{\max}$ 为最大航向速度。目标点的加速度假设为零。

对于一个非线性系统, 如果能够找到一组系统输出, 使得所有状态变量和控制变量可由这组输出与其导数的代数组合表示, 那么称此非线性系统为微分平坦系统, 该系统输出称为平坦输出。将平坦微分理论引入到轨迹规划研究中, 以平坦输出变量代替状态变量和控制变量, 实现了轨迹优化问题中优化变量的降维, 极大地减小计算量。文中选取 AUV 的平坦输出为 $\sigma(t) = [x(t), y(t), z(t)]^T$, 即 AUV 在世界坐标系下的空间位置。其状态变量由平坦输出表示为

$$\begin{cases} \varphi = \arctan(\dot{y}/\dot{x}) \\ u = \sqrt{(\dot{x})^2 + (\dot{y})^2} \\ w = \dot{z} \\ r = (\ddot{y}\dot{x} - \dot{y}\ddot{x})/u^2 \end{cases} \quad (11)$$

同理, 计算前向速度 u 、垂向速度 w 以及偏航角速度 r 相对时间 t 的导数, 可得到 AUV 控制变量在微分平坦空间的表达式。因此, AUV 的控制变量和状态变量均可由平坦输出线性表达。

通过微分平坦特性, 将轨迹优化问题从状态空间 $X(t), U(t)$ 转移至平坦输出空间, 则轨迹优化问题

进一步表示为

$$\begin{cases} \min_{\sigma} \int_0^{t_f} \eta_u^T R \eta_u dt + t_f \\ \text{s.t. } \|\eta_x - O_i\|_2 \geq d_s \\ \|\eta_x\| \leq \bar{\eta}_x, \|\eta_u\| \leq \bar{\eta}_u \\ \eta_x(t_0) = X_0, \eta_x(t_f) = X_f \end{cases} \quad (12)$$

式中: $\eta_x = X(t)$; $\eta_u = U(t)$ 。

3.2 IDVD 方法

IDVD 方法首次由 Yakimenko^[5] 提出, 该方法成功实现了轨迹规划的实时应用, 属于最优控制问题中的直接法, 因此所生成的轨迹属于局部最优解, 且该方法适用于短时间和小尺度场景。IDVD 的计算效率已经在不同的领域得到验证, 如航天器对接^[21]、AUV 与水下基站接驳^[22] 和四旋翼飞行器运动规划^[23] 等。

由于微分平坦特性, 轨迹在平坦输出空间的各坐标轴上实现了相互解耦。鉴于轨迹在空间和时间维度上的连续性, 通常以多项式参数化的方式表示。为了保证数值稳定性, 采用贝塞尔曲线来表示轨迹。该曲线的数学表达式为

$$P(t) = \sum_{i=0}^n c_i b_i^n(t), t \in [0, 1] \quad (13)$$

$$b_i^n(t) = C_n^i t^i (1-t)^{(n-i)} \quad (14)$$

式中: c_i 为控制点; $b_i^n(t)$ 为伯恩斯坦基函数; n 为贝塞尔曲线阶数; t 为归一化时间。一般情况下, 为了满足优化问题的边界条件, 阶数 n 通常需要满足

$$n = d_0 + d_f + 1 \quad (15)$$

式中, d_0, d_f 分别表示起点和终点在各坐标轴的最高阶导数。由于各坐标轴的边界条件包含位置、速度和加速度, 因此设定 $n = 5$ 。

贝塞尔曲线具有以下特殊的性质, 可以方便地进行数值计算:

- 1) 贝塞尔曲线的导数仍是贝塞尔曲线, 其 1 阶曲线的控制点为 $c_i^{(1)} = n(c_{i+1} - c_i)$ 。
- 2) 贝塞尔曲线起始于第 1 个控制点, 终止于最后 1 个控制点, 但曲线并不经过其他控制点, 如图 5 所示。

以 X 轴为例, 起点和目标点的边界条件分别为 $B_0 = [x_0, \dot{x}_0, \ddot{x}_0]^T$, $B_f = [x_f, \dot{x}_f, \ddot{x}_f]^T$, 借助贝塞尔曲线特性, 可构造出线性矩阵方程为

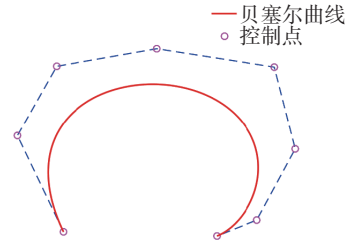


图 5 贝塞尔曲线

Fig. 5 Bezier curve

$$MC = B \quad (16)$$

其中

$$\begin{cases} M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -5 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 5 \\ 20 & -40 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & -40 & 20 \end{bmatrix} \\ C = [c_0, \dots, c_5]^T \\ B = [x_0, x_f, \dot{x}_0 t, \dot{x}_f t, \ddot{x}_0 t^2, \ddot{x}_f t^2]^T \end{cases} \quad (17)$$

式(16)中边界条件 B_0, B_f 已知, 而 t 为未知量, 通过调整 t 的取值, 可以改变贝塞尔曲线的形状。为了提高贝塞尔曲线的灵活性, 常见的策略是增加其阶数。考虑边界条件中起点和目标点的 3 阶导数值, 即 $\ddot{x}_0$ 和 $\ddot{x}_f$, 同时将贝塞尔曲线的阶数提高至 7 阶。由于速度是位置相对时间的导数, 当曲线的系数确定后, 可通过 1 阶微分解析计算其对应的速度曲线。因此在确定贝塞尔曲线的系数后, 其速度曲线也随之确定。为了在空间路径上调整速度值, 引入虚拟弧长概念, 将轨迹优化问题从时间域转换至虚拟弧长域, 从而实现位置和速度的解耦。虚拟弧长和时间的映射关系可以通过虚拟速度表示, 即

$$\lambda(\tau) = \frac{d\tau}{dt} \quad (18)$$

式中: 虚拟弧长 $\tau \in [\tau_0, \tau_f]$; $t \in [t_0, t_f]$ 。需要注意边界条件定义在时间域内, 因此需要将其通过映射关系从时间域转换到虚拟域。将虚拟弧长作为优化变量, 通过改变其大小, 进而更改空间路径对应的速度曲线。对于优化问题的状态和控制量 η , 其映射关系可以表示为

$$\begin{cases} \dot{\eta} = \lambda \eta' \\ \ddot{\eta} = (\eta'' \lambda + \eta' \lambda') \lambda \end{cases} \quad (19)$$

式中: $\dot{\eta}, \ddot{\eta}$ 表示变量对时间的导数; η', η'' 表示变量

对虚拟弧长的导数。虚拟速度与虚拟弧长成单调递增关系, 为不失一般性, 假设端点处的虚拟速度为

$$\begin{cases} \lambda(0) = \lambda(\tau_f) = 1 \\ \lambda'(0) = \lambda'(\tau_f) = 0 \end{cases} \quad (20)$$

贝塞尔曲线及其导数在虚拟域中的关系表达式为

$$\begin{cases} P(\bar{\tau}) = \sum_{i=0}^7 c_i b_i^7(\bar{\tau}) \\ P'(\bar{\tau}) = \sum_{i=0}^6 c_i^{(1)} b_i^6(\bar{\tau})/\tau_f \\ P''(\bar{\tau}) = \sum_{i=0}^5 c_i^{(2)} b_i^5(\bar{\tau})/\tau_f^2 \\ P'''(\bar{\tau}) = \sum_{i=0}^4 c_i^{(3)} b_i^4(\bar{\tau})/\tau_f^3 \end{cases} \quad (21)$$

式中, $\bar{\tau} = \tau/\tau_f \in [0, 1]$ 。根据边界条件, 式(21)可进一步转化为线性矩阵方程, 在此方程中, 只有边界条件的 3 阶导数值 $\ddot{x}_0$ 、 $\ddot{x}_f$ 和虚拟弧长 τ_f 为未知量。因此在整个轨迹优化过程中, 优化变量为 $[x_0''', x_f''', y_0''', y_f''', z_0''', z_f''', \tau_f]$, 即边界条件在每个坐标轴的 3 阶导数值和虚拟弧长。

3.3 轨迹生成方法

在虚拟域中, 首先将 τ 离散化, 划分为等间距的 $N-1$ 个子区间, 每个子区间弧长 $\Delta\tau = (\tau_f - \tau_0)/(N-1)$, 可以得到每个离散点到起点的长度值, 即

$$\tau_i = \tau_{i-1} + \Delta\tau, i = 2, \dots, N \quad (22)$$

进一步通过式(21)得到每个离散点在虚拟域中的位置坐标 $\mathbf{P}_i = [x_i, y_i, z_i]^T$ 和速度向量 $\mathbf{P}'_i = [x'_i, y'_i, z'_i]^T$, 则每相邻离散点之间的时间间隔为

$$\begin{cases} \Delta t_{i-1} = \frac{2\|\mathbf{P}_i - \mathbf{P}_{i-1}\|_2}{v_i + v_{i+1}} \\ v_i = \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2 + z_i^2} \end{cases} \quad (23)$$

在时间域构造非线性优化问题, 目标函数为最小化能耗和航行时间, 其表达式为

$$J_{\text{obj}} = \underbrace{w_u \sum_{j=1}^{N-1} (a_{sj}^2 + a_{hj}^2 + a_{yj}^2) \Delta t_j}_{\text{能耗指标}} + \underbrace{w_t \sum_{j=1}^{N-1} \Delta t_j}_{\text{航行时间}} \quad (24)$$

式中: 能耗指标由每个离散点的控制输入平方和组成, 其中 a_{sj} 、 a_{hj} 、 a_{yj} 分别为前向, 垂向和转向加速度; 航行时间由每个离散点之间的时间和组成; w_u 、 w_t 分别为权重系数。安全约束采用 Yukawa 势场函数^[24]表示惩罚函数, 其数学描述形式为

$$P_c = w_c \sum_{j=1}^N \alpha \exp(-\beta(d_i - d_s)) \quad (25)$$

式中: w_c 为权重系数; α 为缩放因子; β 为延迟率; d_i 为 AUV 当前位置坐标和最近障碍物的距离; d_s 为安全距离阈值。动力学约束保证轨迹规划的速度和加速度大小均在 AUV 执行范围之内, 确保 AUV 易于跟踪规划的轨迹。动力学约束主要包括速度约束和加速度约束, 即

$$P_d = w_d \sum_{j=1}^N \begin{cases} \|\eta_j - \bar{\eta}\|_2^2, & |\eta_i| \geq \bar{\eta} \\ 0, & \text{其他} \end{cases} \quad (26)$$

轨迹优化问题进一步转化为无约束非线性优化问题, 即

$$\min J = J_{\text{obj}} + P_c + P_d \quad (27)$$

4 仿真分析

在全局路径规划中, 路径优化采用拟牛顿法, 如 L-BFGS(limited-memory Broyden-Fletcher-Gold-Shanno)^[25], 并使用 NLOpt 开源数值求解器实现。通过开源计算平台 CasADi^[26], 并使用顺序二次规划(sequential quadratic programming, SQP)实现局部轨迹优化。根据“海缸二号”AUV 的运动性能, 提出的运动规划算法涉及到的参数设置如下: $u=0 \sim 1$ m/s; $w=-0.2 \sim 0.2$ m/s; $r=-0.4 \sim 0.4$ rad/s; $R_p=10$ m。其中, 轨迹执行范围根据轨迹跟踪的精度进行设置, 为了保证较小的跟踪误差, 设置 $R_e=2$ m。测试电脑 CPU 为双核英特尔处理器@2.9 GHz, 8 GB 内存。设置仿真地图的尺寸为 $60 \text{ m} \times 60 \text{ m} \times 30 \text{ m}$ 。设置 AUV 从起点 $\mathbf{P}_0$ 出发, 途经航位点 $\mathbf{P}_1$ 和 $\mathbf{P}_2$, 最终抵达目标点 $\mathbf{P}_f$ 。由于微分平坦理论导致速度存在奇异值, 因此起点和目标点的速度为 0.1 m/s, 加速度为 0。为了保证一定的建模精度和计算效率, 仿真环境中障碍物设置为凸立方体形状。由于轨迹优化属于多目标优化问题, 权重系数反映了不同代价成本的相对重要性, 且某些成本之间的数值取向存在矛盾。通过仿真验证, 确定了较为合理的权重系数。在全局路径优化阶段, 设置权重参数 $\lambda_s = 1.5, \lambda_c = 1.0$; 在局部轨迹优化阶段, 设置权重参数 $w_u = 5, w_t = 1.5, w_c = 3, w_d = 2$ 。设置环境中的洋流速度在 X 和 Y 坐标轴的速度分量为 0.1 m/s。首先进行了路径优化的仿真验证, 如图 6 所示, 其中红色虚线表示初始路径, 蓝色实线表示优化后路径。可以看出优化后的路径趋于平滑且远离障碍物。

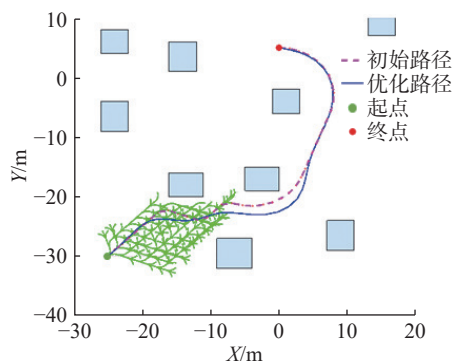


图 6 优化前后路径结果对比

Fig. 6 Comparison of the paths before and after optimization

为了证明文中方法的优势, 分别选用高斯伪谱法^[27](gauss pseudospectral method, GPM)和连续时间^[16](continuous time, CT)法进行了对比测试, 结果如图 7 所示。从图中可以观察到, 3 种方法生成的路径均表现为安全且平滑的特性。与 GPM 方法相比, 在一些区间, CT 法的路径存在波动。这种波动的原因在于 CT 法仅对空间进行优化, 而忽略了时间维度的优化。由于某些路段上的时间分配不合理, 在时间参数化多项式轨迹时可能导致路径与 GPM 稍有偏离。相比之下, 文中方法与 GPM 均采用时空联合方式进行轨迹优化, 因此它们的路径更为吻合。

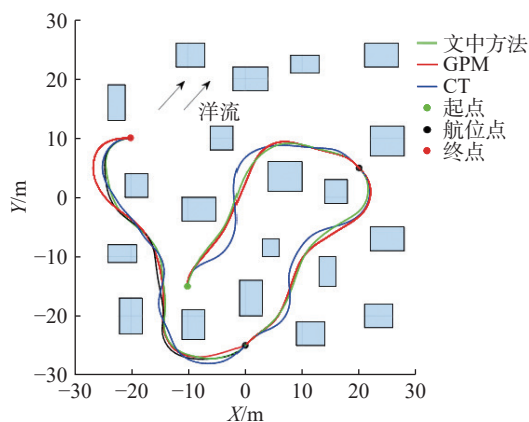


图 7 不同方法路径规划结果对比

Fig. 7 Path comparison with different path planning methods

图 8 展示了 3 种方法在前向速度、垂向速度和偏航角速度方面的对比结果。可以观察到, 所有方法生成的速度曲线均符合 AUV 的运动学限制, 从而确保 AUV 能够有效跟踪轨迹。在前向速度曲

线中, 文中方法与 GPM 相似, 速度曲线均包括加速、匀速和减速 3 个过程。值得注意的是, 在开始阶段, 文中方法的加速度略低于 GPM, 而 CT 法的速度曲线则表现出上下波动。这些结果的主要原因有两方面: 首先, GPM 作为全局轨迹优化方法能够保证计算结果的全局最优性, 而文中方法和 CT 法均采用局部轨迹优化方法, 由于缺乏全局地图信息, 计算结果无法保证全局最优; 其次, CT 法只进行空间轨迹优化, 由于每段路径上时间分配的不合理, 导致速度曲线存在上下波动, 而文中方法和 GPM 均采用时空联合轨迹优化方法, 因此二者的速度曲线形状更为相似。

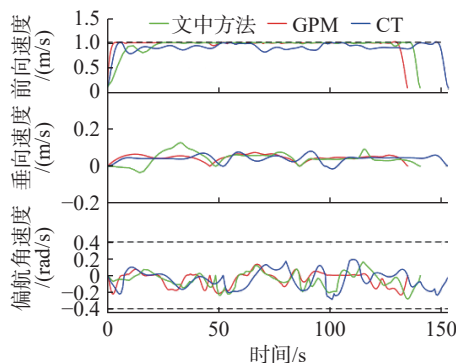


图 8 不同方法速度对比

Fig. 8 Velocity comparison with different methods

表 1 展示了 3 种方法在航行时间和计算时间方面的对比结果。在计算效率方面, 文中方法与 CT 法的计算效率相当, 比 GPM 快了 2 个数量级, 在 C++ 环境下测试的计算时间仅为 17 ms, 实现了实时轨迹规划, 而 GPM 由于存在大量的优化变量, 导致其计算时间相对较长。在航行时间方面, GPM 的航行时间最短, 为 134.9 s; CT 法由于仅对空间进行轨迹优化, 因此其航行时间最长, 为 153.2 s; 相比之下, 文中方法通过对轨迹进行时空联合优化, 航行时间为 140.6 s, 略慢于 GPM, 但相比 CT

表 1 不同方法计算性能对比

Table 1 Computational performance comparison with different methods

方法	航行时间/s	平均计算时间/s	
		Matlab	C++
文中方法	140.6	1.23	0.017
CT	153.2	1.18	0.020
GPM	134.9	72.38	6.320

法提高了 8.9%。这表明文中方法与 CT 法在计算效率上相当, 并且在轨迹质量上展现出接近 GPM 的水平。

此外, 为了评估轨迹优化阶段的收敛速度, 进一步对 3 种方法进行对比, 如图 9 所示。显而易见, 文中方法表现出了最快的迭代速度, 在进行 40 次迭代后迅速收敛至最优解。这主要归因于文中方法仅包含 7 个优化变量, 极大地提高了计算效率。相较之下, CT 法采用多项式参数化表示轨迹, 其优化变量为多项式系数, 数量相比文中方法多了数倍, 在进行 100 次迭代后达到收敛。而 GPM 则呈现出最慢的计算效率, 主要因为其优化变量数量最多, 每个离散点处的状态变量和控制变量均为优化变量, 在进行 220 次迭代后才实现了收敛。

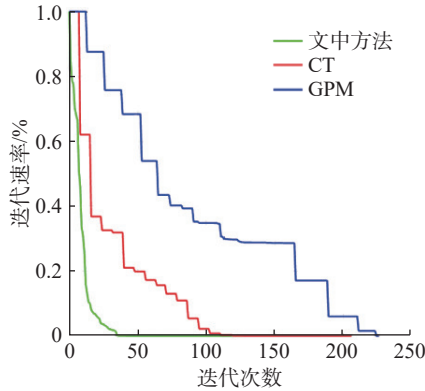


图 9 不同方法收敛速度对比

Fig. 9 Convergence speed comparison with different methods

5 水池测试

为了验证文中方法的有效性及其在工程实践中的可用性, 进行了水池实验, 以“海缸二号”AUV 为测试对象, 其尺寸为 1.8 m×1.2 m×0.8 m, 水池的尺寸为 20 m×10 m, 深度达 20 m。

测试过程采用地面端电脑通过光纤与 AUV 相连, 进而控制其运动。电脑配置与仿真测试中相同, 实验中采用机器人操作系统(robotics operation system, ROS)作为开发环境, 同时采用 C++11 进行开发。在水池实验中采用虚拟障碍物设置地图环境, 为安全起见设置 AUV 的最大前向速度为 0.3 m/s, 最大偏航角速度为 0.15 rad/s。根据 AUV 的尺寸, 设置 AUV 与障碍物的安全距离为 2 m, 即 $d_s = 2.0$ m, 其他参数设置和仿真测试相同。在水池分别测试

了 S 型和 8 字型轨迹以验证文中方法的有效性。

在 S 型轨迹测试场景中, 设置 AUV 从起点 P_0 出发, 抵达终点 P_f 。此外, 设置 2 个立方体作为虚拟障碍物, 尺寸为 3 m×5 m×20 m, 其中心的水平面坐标分别为 [3.5 m, 2.5 m] 和 [11.5 m, 7.5 m]。图 10 为期望路径和实际路径的对比, 由于 PID 运动控制器的计算误差和基于滚动窗口的规划机制, 导致期望路径在空间中不连续, 但在每个规划周期内, 期望路径能够保持平滑且安全, 以引导 AUV 驶向目标点。

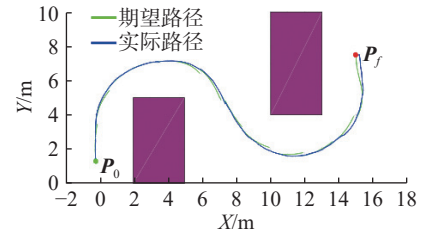


图 10 S 型路径

Fig. 10 S-shaped paths

图 11 展示了实验过程的不同时刻路径, 每个子图的左侧为 ROS RVIZ(robot visualization)可视化视角, 其显示了动态规划路径的过程。其中: 红色曲线表示规划轨迹, 在不同时刻均能够保证路径的安全性; 绿色和蓝色曲线分别表示已规划的路径和 AUV 的真实路径。

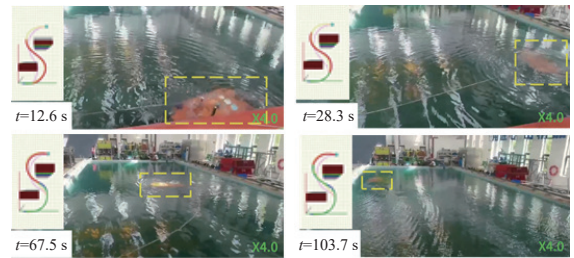


图 11 S 型轨迹瞬时路径

Fig. 11 Instantaneous paths of S-shaped trajectory

图 12 为期望的速度时间曲线, 曲线中的每个突变时刻表示轨迹的重规划过程, 最大前向速度为 0.265 m/s, 最大偏航角速度为 0.11 rad/s。在整个轨迹跟踪过程中, 这 3 条速度曲线的期望值均满足 AUV 的动力学限制, 表明 AUV 能够有效地跟踪规划轨迹。

在 8 字型轨迹测试场景中, 设置 AUV 从起点 P_0 出发, 分别途径 4 个航位点 $P_1 \sim P_4$, 最终抵达终点 P_f 。虚拟障碍物由 2 个直径为 4 m 的圆柱体组

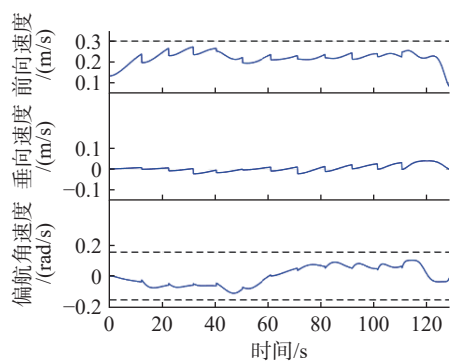


图 12 S 型轨迹速度曲线

Fig. 12 Velocity curves in S-shaped trajectory

成, 其水平坐标分别为 [4 m, 5 m] 和 [12 m, 5 m]。图 13 为期望路径和实际路径的对比, 与 S 型轨迹相似, 期望路径在空间表现出不连续特性。可以看出期望路径能够准确地经过航位点, 而由于 PID 控制器的计算误差, 真实路径并不能精准地经过航位点, 尤其在航位点 P_2 、 P_3 、 P_4 处出现较大的跟踪误差。这是因为当 AUV 接近这些航位点时, 推进器的旋转方向发生改变, 其推力转速曲线在零转速附近有较强的非线性特性, 导致出现较大的跟踪误差。

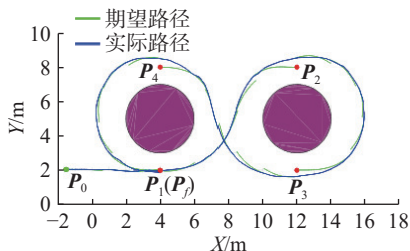


图 13 8 字型路径

Fig. 13 8-shaped paths

图 14 展示了实验过程的不同时刻路径。图 15 为 8 字型轨迹速度曲线, 其中最大前向速度为 0.27 m/s, 最大偏航角速度为 0.15 rad/s, 并且在每个航位点处的速度均符合设定的速度范围。

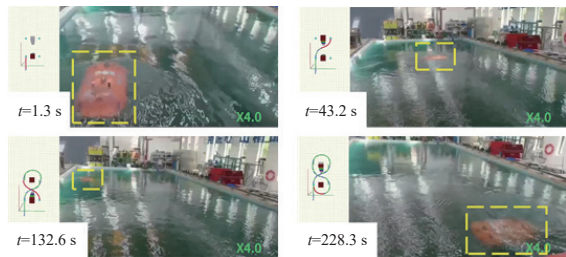


图 14 8 字型轨迹瞬时路径

Fig. 14 Instantaneous paths of 8-shaped trajectory

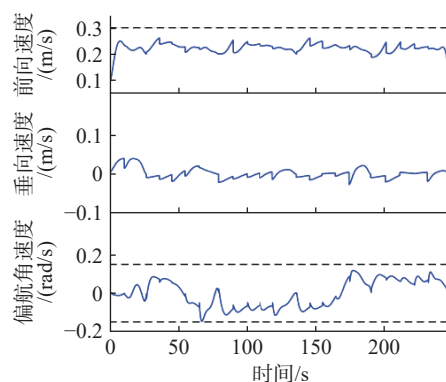


图 15 8 字型轨迹速度曲线

Fig. 15 Velocity curves in 8-shaped trajectory

以上不同场景的实验测试中, 每个规划周期内的计算时间如表 2 所示, 结果表明文中方法具有实时轨迹规划的能力。

表 2 规划周期内的计算时间

Table 2 Calculation time in planning cycle ms			
轨迹	最小时间	最大时间	平均时间
S型	24	47	37
8字型	22	45	36

6 结束语

针对 AUV 在未知环境下的自主航行问题, 提出了一种融合混合 A*算法与 IDVD 方法的分层运动规划框架。为了应对未知环境中的不确定性, 引入滚动窗口机制的规划策略, 进一步增强算法的鲁棒性和适应性。分别通过仿真测试和水池实验, 验证了文中方法在实时轨迹规划方面的性能, 能够有效生成安全、可行的轨迹, 达到了预期目标。

然而, 由于水池环境的局限性(如空间较小、缺乏真实海洋环境中的复杂动态因素), 实验结果尚不能完全反映算法在真实海洋环境中的性能。因此, 未来将进一步开展真实海洋环境下的运动规划实验, 以验证和优化算法的实际应用能力, 推动 AUV 在复杂海洋场景中的智能化与自主化发展。

参考文献:

- [1] GONG Y J, HUANG T, MA Y N, et al. MTrajPlanner: A multiple-trajectory planning algorithm for autonomous underwater vehicles[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(4): 3714-3727.
- [2] ZENG Z, SAMMUT K, LAMMAS A, et al. Efficient path re-planning for AUV operating in spatiotemporal

- currents[J]. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2015, 79(1): 135-153.
- [3] WILLNERS J S, GONZALEZ-ADELL D, HERNÁNDEZ J D, et al. Online 3-dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning[J]. *Sensors*, 2021, 21(4): 1-20.
 - [4] HERNÁNDEZ J D, VIDAL E, VALLICROSA G, et al. Online path planning for autonomous underwater vehicles in unknown environments[C]//2015 IEEE International Conference on Robotics and Automation. Seattle, USA: IEEE, 2015: 1152-1157.
 - [5] YAKIMENKO O A. Direct method for rapid prototyping of near-optimal aircraft trajectories[J]. *Journal of Guidance, Control, and Dynamics*, 2000, 23(5): 865-875.
 - [6] ZHOU B, GAO F, WANG L, et al. Robust and efficient quadrotor trajectory generation for fast autonomous flight[J]. *IEEE Robotics and Automation Letters*, 2019, 4(4): 3529-3536.
 - [7] LIU S, 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.
 - [8] MA D, HAO S, MA W, et al. An optimal control-based path planning method for unmanned surface vehicles in complex environments[J]. *Ocean Engineering*, 2022, 245: 110532.
 - [9] ZHUANG Y, SHARMA S, SUBUDHI B, et al. Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm[J]. *Ocean Engineering*, 2016, 127: 190-199.
 - [10] MAHMOUD ZADEH S, POWERS D M W, SAMMUT K, et al. A novel versatile architecture for autonomous underwater vehicle's motion planning and task assignment [J]. *Soft Computing*, 2018, 22(5): 1687-1710.
 - [11] RICHTER C, BRY A, ROY N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments[J]. *Springer Tracts in Advanced Robotics*, 2016, 114: 649-666.
 - [12] KARAMAN S, FRAZZOLI E. Incremental sampling-based algorithms for optimal motion planning[J]. *Robotics Science and Systems*, 2010, 104(2): 267-274.
 - [13] HERNÁNDEZ J D, MOLL M, VIDAL E, et al. Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments[C]//2016 IEEE International Conference on Intelligent Robots and Systems. Daejeon, Korea(South): IEEE, 2016: 1313-1320.
 - [14] CARRERAS M, HERNANDEZ J D, VIDAL E, et al. Sparus II AUV-A hovering vehicle for seabed inspection [J]. *IEEE Journal of Oceanic Engineering*, 2018, 43(2): 344-355.
 - [15] USENKO V, VON STUMBERG L, PANGERCIC A, et al. Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer[C]//2017 IEEE International Conference on Intelligent Robots and Systems. Vancouver, BC, Canada: IEEE, 2017: 215-222.
 - [16] OLEYNIKOVA H, BURRI M, TAYLOR Z, et al. Continuous-time trajectory optimization for online UAV replanning[C]//2016 IEEE International Conference on Intelligent Robots and Systems. Daejeon, Korea(South): IEEE, 2016: 5332-5339.
 - [17] ZHOU X, WANG Z, YE H, et al. EGO-Planner: An ESDF-free gradient-based local planner for quadrotors[J]. *IEEE Robotics and Automation Letters*, 2021, 6(2): 478-485.
 - [18] YE H, ZHOU X, WANG Z, et al. TGK-planner: An efficient topology guided kinodynamic planner for autonomous quadrotors[J]. *IEEE Robotics and Automation Letters*, 2021, 6(2): 494-501.
 - [19] SHKEL A M, LUMELSKY V. Classification of the Dubins set[J]. *Robotics and Autonomous Systems*, 2001, 34(4): 179-202.
 - [20] ZHU Z, SCHMERLING E, PAVONE M. A convex optimization approach to smooth trajectories for motion planning with car-like robots[C]//2015 Proceedings of the IEEE Conference on Decision and Control. Osaka, Japan: IEEE, 2015: 835-842.
 - [21] BOYARKO G, YAKIMENKO O, ROMANO M. Real-Time 6DoF guidance for spacecraft proximity maneuvering and close approach with a tumbling object[C]//2010 AIAA/AAS Astrodynamics Specialist Conference. Toronto, Ontario, Canada: AIAA, 2010: 7666.
 - [22] YAZDANI A M, SAMMUT K, YAKIMENKO O A, et al. IDVD-based trajectory generator for autonomous underwater docking operations[J]. *Robotics and Autonomous Systems*, 2017, 92: 12-29.
 - [23] COWLING I D, YAKIMENKO O A, WHIDBORNE J F, et al. Direct method based control system for an autonomous quadrotor[J]. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2010, 60(2): 285-316.
 - [24] ALTURBEH H, WHIDBORNE J F. Real-time obstacle collision avoidance for fixed wing aircraft using B-splines [C]//2014 UKACC International Conference on Control. Loughborough, UK: IEEE, 2014: 115-120.
 - [25] HEAD J D, ZERNER M C. A Broyden-Fletcher-Goldfarb-Shanno optimization procedure for molecular geometries[J]. *Chemical Physics Letters*, 1985, 122(3): 264-270.
 - [26] ANDERSSON J A E, GILLIS J, HORN G, et al. CasADi: A software framework for nonlinear optimization and optimal control[J]. *Mathematical Programming Computation*, 2019, 11(1): 1-36.
 - [27] GAN W, SU L, CHU Z. Trajectory planning of autonomous underwater vehicles based on Gauss pseudospectral method[J]. *Sensors*, 2023, 23(4): 57-71.

(责任编辑: 许妍)