

改进型 Bi-RRT* 算法的水面无人艇路径规划

师颖慧¹, 张冰², 赵强²

(1. 江苏航空职业技术学院 无人机与智能产业学院, 江苏 镇江 212134;
2. 江苏科技大学 自动化学院, 江苏 镇江 212100)

摘要: 水面无人艇(USV)在进行路径规划时存在算法收敛速度慢、规划出的路径过于曲折的问题,无法满足其在路径规划时对于实时性、路径平滑性与可行性的要求。为此,在双向快速扩展随机树(Bi-RRT*)算法基础上,设计一种加入目标偏置策略并且引入智能样本插入的算法(PIB-RRT*)。用目标偏置策略代替RRT*中的四周均匀采样策略,从而提高算法的收敛速度;选择新节点时引入智能样本插入,使算法快速收敛到最优解;在节点生成后引入转向角约束,避免出现大角度转向,从而使规划出的路径对于无人艇更具有可行性。分别设计简单、复杂和特殊3种环境下的路径规划实验方案,对比PIB-RRT*算法、Bi-RRT*算法、RRT*算法在规划时间和路径长度两个指标下的规划效果。实验结果证明,PIB-RRT*算法在满足USV转角约束的同时能够大幅提高收敛速度,算法收敛速度为原来的1/10,路径长度减少了约10%,路径的平滑度也更高。

关键词: 无人艇;路径规划;Bi-RRT*算法;PIB-RRT*算法;转向角约束

DOI:10.11907/tjdx.251127

中图分类号:TP18

文献标识码:A

文章编号:1672-7800(2025)010-0134-07



扫描二维码阅读全文:

Improved Bi-RRT * Algorithm for Path Planning of Unmanned Surface Vehicles

SHI Yinghui¹, ZHANG Bing², ZHAO Qiang²

(1. School of Drones and Intelligent Industry, Jiangsu Aviation Vocational and Technical College, Zhenjiang 212134, China;
2. School of Automation, Jiangsu University of Science and Technology, Zhenjiang 212100, China)

Abstract: Unmanned surface vehicles (USVs) face problems in path planning, such as slow algorithm convergence and excessively tortuous paths, which cannot meet their requirements for real-time, smooth, and feasible path planning. Based on the Bidirectional Rapidly Expanding Random Trees (Bi-RRT *) algorithm, a PIB-RRT * algorithm is designed, which incorporates target bias strategy and intelligent sample insertion. Replace the uniform sampling strategy in RRT * with a target bias strategy to improve the convergence speed of the algorithm; Introducing intelligent sample insertion when selecting new nodes enables the algorithm to quickly converge to the optimal solution; Introducing steering angle constraints after node generation to avoid large angle turns, thus making the planned path more feasible for unmanned vessels. Design experimental plans for path planning in three different environments: simple, complex, and special. Compare the planning performance of PIB-RRT * algorithm, Bi-RRT * algorithm, and RRT * algorithm under two indicators: planning time and path length. The experimental results demonstrate that: The PIB-RRT * algorithm can greatly improve the convergence speed while satisfying the USV corner constraint. The convergence speed of the algorithm is 1/10 of the original, the path length is reduced by about 10%, and the smoothness of the path is also higher.

Key Words: unmanned surface vehicle; path planning; Bi-RRT * algorithm; PIB-RRT * algorithm; steering angle constraint

0 引言

水面无人艇(Unmanned Surface Vehicle, USV)是一种

可以在复杂及危险的海洋环境中开展长时间、大范围海洋作业的小型化、多用途、智能的无人海洋运载平台^[1]。水面无人艇属于机器人的一种,机器人路径规划主要包括3个方面:①最优性:确保能找到最优路径^[2];②可行性:机

收稿日期:2025-03-02

基金项目:镇江市重点研发计划项目(SH2023098);江苏航空职业技术学院院级课题资助项目(JATC24010113)

作者简介:师颖慧(1993-),女,江苏航空职业技术学院无人机与智能产业学院硕士研究生,研究方向为机器人路径规划技术;张冰(1967-),女,博士,江苏科技大学自动化学院教授,研究方向为船舶电子信息控制。本文通讯作者:师颖慧。

器人不会在计划的路径上与障碍物碰撞^[3];③完备性:如果存在路径,总是可以找到它^[4]。

虽然现在一些用于无人艇路径规划的算法已经比较成熟,如 Dijkstra 算法^[5]、遗传算法^[6]、A-star 算法^[7]、粒子群算法和原始蜣螂优化算法等。然而,在复杂的场景中,这些算法并不总能满足最优性、可行性或完备性的要求。Dijkstra 算法是典型的单源最短路径算法,用于计算一个节点到其他所有节点的最短路径^[8]。虽然其获得最短路径的成功率高,且鲁棒性好,但在大规模复杂路径拓补网络中,节点遍历和效率低下是其致命缺陷,所以不适合复杂的海面环境。A*算法是 Dijkstra 算法的优化算法,相比 Dijkstra 算法搜索效率有所提升,但是收敛速度慢、路径转折点多,且与障碍物距离过近,规划出的路线对于无人艇在海面行驶不具有可行性和安全性。粒子群优化算法(Particle Swarm Optimization, PSO)是一种智能优化算法,其优化方式非常简单,容易理解且不需要复杂的计算,也不需要过多的调整参数,鲁棒性好,但是算法后期收敛速度慢,容易陷入局部最优,使得规划出的路径不具有最优性^[9]。遗传算法具有良好的鲁棒性和全局寻优能力,但是计算速度较慢,局部搜索能力差,使得规划出的路径不具有实时性。原始的蜣螂优化算法(Dung Beetle Optimizer, DBO)收敛精度低,且易陷入局部最优^[10]。

USV 体积小、速度快,需要为其规划高质量路径,保证其安全到达目标点并完成相应的水上任务^[13]。然而,海上环境信息复杂多变,上述算法无法保证规划的安全性、实时性和最优性。快速探索随机树(Rapidly-exploring Random Trees, RRT)算法用于构建植根于起点的树,并通过在空间中随机采样来探索状态空间^[11-13]。一旦对目标位置进行了采样,探索过程则结束。该算法空间搜索能力强、避障性能优、实时性强,可满足 USV 航行的要求。

然而, RRT 算法也存在一些局限性。例如,使用 RRT 算法时通常容易陷入局部最优,而且所有基于 RRT 的方法几乎都是次优的。由于使用 RRT 算法规划出的路径不是最优,研究人员于 2011 年又提出了 RRT* 算法,其规划出的解接近最优解,在 RRT 基础上添加了随机几何映射和修剪优化理论^[14]。但该算法与 RRT 算法一样,在采样时由于采用随机搜索策略,导致路径搜索效率较低、路径平滑性与路径质量较差,规划出的路径对于 USV 来说不具有可行性。而且在复杂的规划空间内,其规划时间也大幅增加。

为了解决以上 RRT* 算法存在的问题,研究人员提出了很多改进方法,大致可分为两大类^[15]:

一类是对经典 RRT 算法在扩展方向、选择步长等方面进行改进。例如 Ayalew 等^[16]提出双向版的 RRT* 算法,也称为 B-RRT* 算法。算法由原来在起点的一棵树搜索变为分别在起点、终点的两棵树上同时搜索,加快了收敛速度,但在复杂环境下计算节点数量较多,规划效率低。Liu

等^[17]利用三角不等式原理在 RRT* 算法基础上采用重新选择父节点和路径重连策略,极大地减少了节点数量,但规划出的路径转角较大,USV 的控制难度大。

另一类是与其他路径规划算法融合的改进,例如结合 APF(Artificial Potential Field)算法^[18]。相较于经典的 RRT* 算法,改进后的算法控制了节点扩展的方向和步长,减少了参与运算的节点数量,在一定程度上提高了算法收敛速度,但是仍存在算法收敛速度慢,以及路径转角过大、不平滑的问题,无法满足 USV 海上航行对实时性和可行性的要求。本文针对以上基于 RRT* 算法的路径规划中仍未解决的问题,改进了 Bi-RRT*(Bidirectional Rapidly-exploring Random Trees)算法:①改进 Bi-RRT* 算法的随机点采样机制,采用目标偏置策略,使随机点有一定概率落在目标区域,提高算法的搜索效率;②选择新节点时引入智能样本插入算法,把所有可能的节点都融入计算中,提高算法收敛到最优解的速度;③根据 USV 的转向性能要求,增加转向角约束,使规划的算法能够与底层控制相结合,降低 USV 的控制难度。

1 算法介绍

1.1 RRT* 算法

RRT* 算法在 RRT 基础上添加了重选父节点和路径重连两个过程。通过路径重连的过程不断优化路径,最终使算法趋于最优。

RRT* 算法的基本原理如下^[19]:

(1)重选父节点。RRT* 算法在新产生的节点 q_{new} 附近以定义的半径范围(即 q_{new} 的邻域)寻找近邻点,作为替换 q_{new} 父节点的备选。近邻点的集合 $q_{near} = \{q_{near1}, q_{near2}, q_{near3}, \dots\}$ 。RRT* 算法在每次迭代中遍历集合 q_{near} 内的所有节点,对集合中的每个节点分别计算起始点 q_{init} 到 q_{near} 的距离与 q_{near} 到 q_{new} 的距离之和,将对应距离之和最小时的 q_{near} 作为新节点 q_{new} 的父节点。在图 1 中,节点 q_{new} 原来的父节点为 q_{near1} ,经过重选父节点后,在图 2 中,节点 q_{new} 的父节点变为 q_{near4} 。

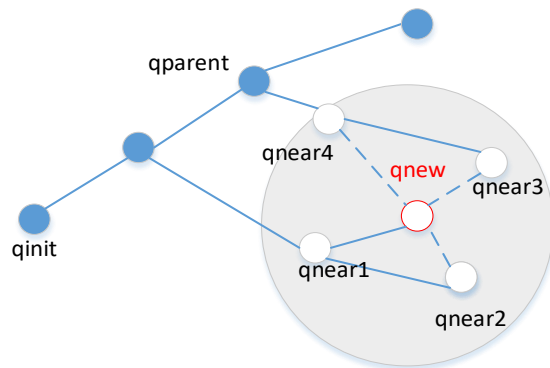


Fig. 1 Expansion process of RRT* nodes

图 1 RRT* 节点扩展过程

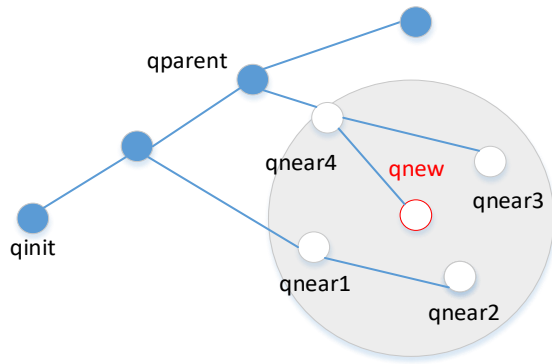


Fig. 2 RRT* reselecting parent node

图2 RRT*重选父节点

(2) 路径重连。集合 q_{near} 范围内其他节点将节点 q_{new} 作为父节点, 分别重新计算 q_{init} 到 q_{new} 与 q_{new} 到集合 q_{near} 内所有点的路径距离之和。若路径与原来相比变短, 则更新父节点。在采样的同时优化路径, 直到到达终点。在图 3 中, 在经历过重选父节点之后, q_{new} 的父节点为 q_{near4} , 此时把节点 q_{new} 作为父节点。在图 4 中, 重新计算距离后, 从 q_{init} 到 q_{new} 与 q_{new} 到 q_{near2} 的距离之和小于 q_{init} 到 q_{near2} 的距离, 所以将 q_{near2} 的父节点改为 q_{new} 。在 q_{near} 集合内, 其他点由于计算后距离值不变, 其父节点保持不变。由此可见, 重新布线可使随机树节点之间连接的距离尽量小。

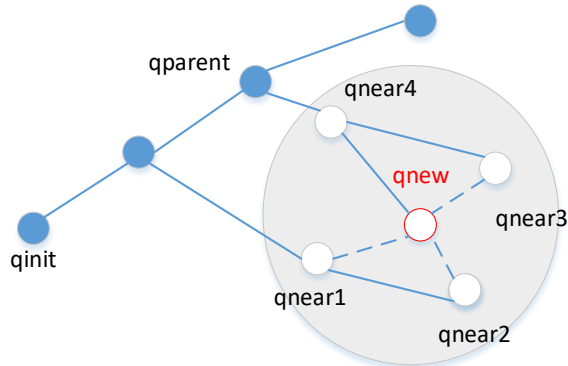


Fig. 3 RRT* traversing the parent node

图3 RRT*遍历父节点

1.2 Bi-RRT*算法

Bi-RRT*算法中包括两棵树, 分别记作 T_a 和 T_b 。Bi-RRT*在初始阶段以与原始 RRT*算法完全相同的方式工作, 即从配置空间 X_{free} 的采样开始, 之后对该随机样本 q_{new} 执行各种操作(与 RRT*相同)。将随机样本成功插入到树中后, 该算法计算从树 T_b 中 x_{new} 最近顶点 $x_{nearest}$ 的距离, 连接 x_{new} 、 $x_{nearest}$ 和 T_b , 然后执行两棵树的连接过程^[20]。

如果建立连接的尝试成功, 连接函数将返回连接起始点 x_{init} 和目标点 x_{goal} 的无碰撞路径 σ_{new} 。然后将该 σ_{new} 的成本与先前计算的路径 σ_{best} 进行比较。如果

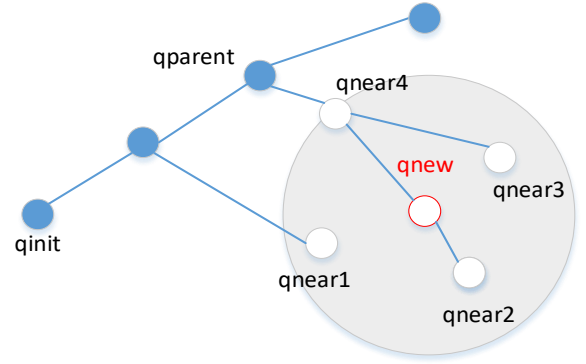


Fig. 4 RRT* path reconnection

图4 RRT*路径重连

σ_{new} 的成本小于 σ_{best} 的成本, 则 σ_{best} 被 σ_{new} 覆盖。最后, 迭代通过交换树结束, 在下一次迭代中在另一棵树上执行相同过程。

2 Bi-RRT*算法改进

2.1 随机点采样机制改进

针对 Bi-RRT*算法在路径扩展过程中因采用全局均匀采样策略使得算法采样效率低下的问题, 本文提出目标偏置采样方法。引入采样概率 λ ($0 \leq \lambda \leq 1$), 用 $rand$ 函数产生随机点。若 $rand < \lambda$, 则将采样点 $sample$ 设置为 $rand$, 即在全局空间自由区内随机产生采样; 若 $rand \geq \lambda$, 则将采样点设置为随机点乘以算法扩展的步长 σ , 保证了算法的概率完备性。

$$sample = \begin{cases} rand, & rand < \lambda \\ rand * \sigma & rand \geq \lambda \end{cases} \quad (1)$$

2.2 新节点智能插入

本文在 Bi-RRT*(双向扩展 RRT*算法)基础上引入新节点智能插入算法(Insert Bidirectional - Rapidly -exploring Random Trees, IB-RRT*)专用于复杂环境搜索。设来自树 T_a 和 T_b 的近顶点集分别由 q_{near}^a 、 q_{near}^b 表示, IB-RRT*算法逐步构建双向树。首先从无障碍配置空间 q_{free} 中选取一个随机样本 q_{rand} , 即 $q_{rand} \in q_{free}$ 。其次, 在以 q_{rand} 为中心、 r 为半径的球区域填充两棵树的一组近顶点 q_{near}^a 、 q_{near}^b , 该球区域记为近邻点集合 q_{near} 。应注意的是, 在形成的球区域中包括了来自两个树的近顶点。如图 5 所示, q_{near}^a 、 q_{near}^b 分别包含来自树 T_a 和 T_b 的 q_{rand} 近顶点。在两个近顶点集都为空的情况下, 这些集将用其各自树中最近的顶点填充, 即各自树上最靠近随机样本的顶点。

IB-RRT*算法主要包括 3 个步骤: 选择最佳的父节点、路径重连、双向树连接。

(1) 选择最佳的父节点。最初计算两棵树的最佳父顶点, 对于球区域每个近邻点, 分别计算从起始点 q_{init} 到近邻点的路径代价(距离)与近邻点到 q_{rand} 的路径代价, 并且将计算结果在列表中进行排序。在该过程中需要注意的是, 当起始点为 q_{init} 时, 在计算路径代价时计算来自球区域中

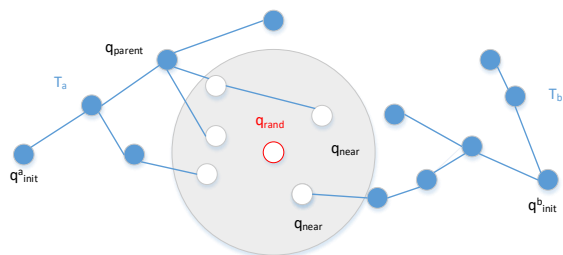


Fig. 5 Calculating a set of near nodes from two trees

图 5 从两棵树计算一组近节点

树 Ta 的近顶点, 此时树 Ta 中路径代价的最小值记为 q_{min}^a ; 当起始点为 q_{init}^a 时, 在计算路径代价时计算来自球区域中树 Tb 的近顶点, 此时树 Tb 中路径代价的最小值记为 q_{min}^b 。一旦列表按升序排列, 即出现了路径代价的最小值时, 将随机顶点 q_{rand} 插入到最佳选择树中。此时, 如果 $q_{min}^a < q_{min}^b$, 即 q_{rand} 距离 Ta 近, 则将随机点 q_{rand} 插入 Ta 中, 两颗树中路径的最小值 $x_{min} = q_{min}^a$; 如果 $q_{min}^a > q_{min}^b$, 即 q_{rand} 距离 Tb 近则 will 将随机点 q_{rand} 插入 Tb 中, 两颗树中路径的最小值 $x_{min} = q_{min}^b$ 。算法也会返回对应最佳选择树上最近的顶点, 该顶点有资格成为随机样本的父节点。如图 6 所示, Ta 为最佳选择树, 两颗树中路径的最小值 $x_{min} = q_{min}^a$ 。

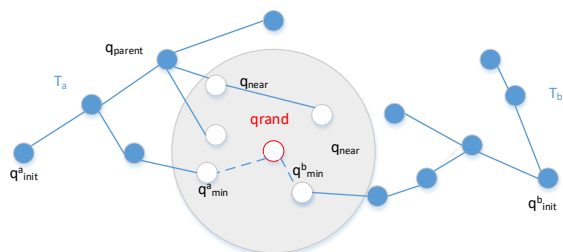


Fig. 6 Selecting the parent node

图 6 选择父节点

(2) 路径重连。选择最佳的父节点后, 随机顶点 q_{rand} 将插入到最佳选择树中, 即随机顶点 q_{rand} 成为了树中扩展的节点之一。接下来算法开始检查随机样本 q_{rand} 作为集合 q_{near} 中每个顶点的父节点的可行性。设顶点 $X \in q_{near}$, 如果通过随机样本 q_{rand} 到达邻近集 q_{near} 中特定顶点 X 的代价小于现有代价, 则 q_{rand} 成为该特定顶点 $X \in q_{near}$ 的父顶点, 保留最短的路径。如图 7 所示, 从 q_{init}^a 到 q_{rand} 再到 x_{min} 的路径代价与现有代价相同, 所以保留从 q_{rand} 到 x_{min} 的路径, 将 q_{rand} 到其他邻近点的路径舍弃。

(3) 双向树连接。当以 q_{rand} 为中心的区域球包含来自

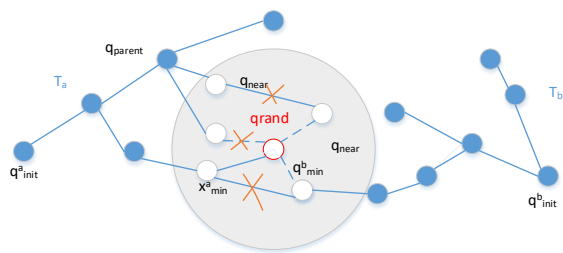


Fig. 7 Reselecting parent node

图 7 重选父节点

两个树 Ta 和 Tb 的近顶点时, 找到两棵树的最小路径代价 x_{min}^a , 算法则会连接两棵树, 否则连接不成功。因此, 与 Bi-RRT* 启发式连接不同, IB-RRT* 算法不贪婪, 因为连接仅在球区域内部进行, 如图 8 所示。最后, 树连接生成端到端全局路径, 如图 9 所示。

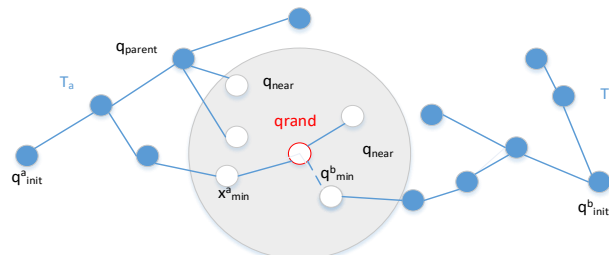


Fig. 8 Path reconnection

图 8 路径重连

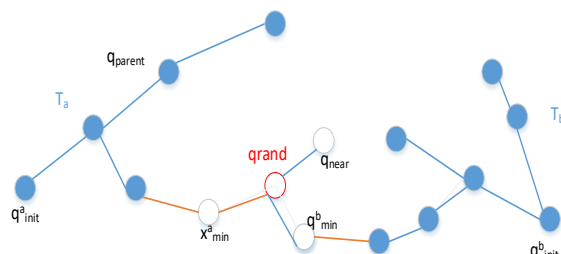


Fig. 9 Forming the final path

图 9 形成最终路径

2.3 转向角约束

USV 在海上航行时, 由于自身动力学和运动学的约束, 转向角限制在 $0^\circ \sim 40^\circ$ 。为更符合海上实际操纵情况, 本文将最大转向角设置为 38° 。在图 10 中, 以扩展树 Ta 为例, 设本次迭代过程中插入的新节点为 q_{new} , 计算上次迭代过程向量 $q_{min}^a q_{rand}$ 与向量 $q_{rand} q_{new}$ 的夹角 θ 。当 $\theta > 38^\circ$ 时, 表明新节点 q_{new} 不满足转向角约束, 舍弃 q_{new} ; 当 $\theta \leq 38^\circ$ 时, 表明扩展树延伸成功, 进入下一次迭代, 将 q_{new} 插入树中。

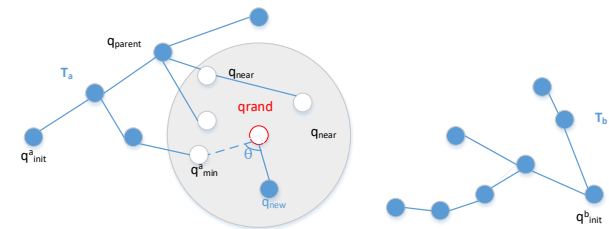


Fig. 10 Steering angle constraint

图 10 转向角约束

2.4 算法分析

本节将从采样方法、新节点的智能插入、转向角约束 3 个方面分析改进后的 PIB-RRT* 算法性能。

(1) 采样方法。PIB-RRT* 算法使用目标偏置采样方法代替 RRT* 中的全局均匀采样策略, 可以提高算法采样效率, 并且相较于 RRT* 一颗树的扩展, PIB-RRT* 算法是两棵树同时扩展, 可以加快算法运算速度。

(2) 新节点的智能插入。由于 RRT* 在搜索过程中生

成点附近的采样点更密集,因此如果采样点更接近树的生成点,其邻域中包含的邻近点密度更大。在算法进行路径重连时,检测随机样本 q_{rand} 到达邻近集 q_{near} 中特定顶点 x' 的成本是否低于现有成本。如果低于现有成本,则 q_{rand} 成为该特定顶点 $x' \in q_{\text{near}}$ 的父顶点。因此,PIB-RRT*将随机点插入邻近点密度更大的区域,使更多节点参与路径重连,尽可能使更多节点参与到每次迭代的重新布线过程中。这一步骤允许算法快速收敛到最优解。

(3)转向角约束。为了使规划的算法更容易控制USV,在改进算法中加入转向角约束,减少了转角时参与的节点数量,避免了绕行时左右频繁的大角度转舵,使规划出的路径更平滑,算法收敛速度更快。

3 实验结果与分析

为验证 PIB-RRT*算法的有效性,利用 MATLAB 分别进行简单、复杂和特殊3种水域环境下的仿真实验。在简单、复杂的仿真环境中,利用栅格法将环境地图分为 630×840 格,起始点设为(15,614),目标点设为(767,39);在特殊水域的仿真环境中,将环境地图分为 720×960 格,起始点设为(55,493),目标点设为(905,199)。比较 PIB-RRT*算法、Bi-RRT*算法和 RRT*算法的路径规划效果,采用规划时间、路径长度指标对算法进行评价。每组进行50次寻优实验,统计每个算法指标的平均值。

3.1 简单水域环境实验

从起始点到终点的路径上共设置大小和位置不同的8个障碍物(分别用不同半径的黑色圆形表示),障碍物位置沿从起始点到终点的直线方向呈对称排列,障碍物大小由起点到终点方向从小到大排列。在起始点和目标点之间设置障碍物能更好地验证算法的有效性,只有8个障碍物的分布凸显了简单的水下环境。3种算法的最终路径规划效果见图11,指标统计结果见表1。相较于其他两种算法规划出的路径,PIB-RRT*算法因为引入了目标偏置和转向角约束,所以在路径长度、路径平滑度和算法收敛速度上有了很大的改进。Bi-RRT*算法由于在全局空间内搜索扩展,所以规划效率不高。相较于 Bi-RRT*算法,PIB-RRT*算法规划时间减少了78%。RRT*算法由于是一棵树单向搜索,且在扩展过程中不考虑路径最优,所以规划出的路径较为曲折。相较于 RRT*算法,PIB-RRT*算法规划出的路径长度减少了27%。

3.2 复杂水域环境实验

在简单水域环境的基础上,从起始点到终点的路径上增加6个大小不一、随机排列的障碍物,共设置14个障碍物,障碍物数量的增多以及不规则分布增加了环境的复杂性。3种算法的最终路径规划效果见图12,指标统计结果见表2。PIB-RRT*算法相较于 Bi-RRT*算法,所用时间减少了43%;相较于 RRT*算法,所用时间减少了87%。通过

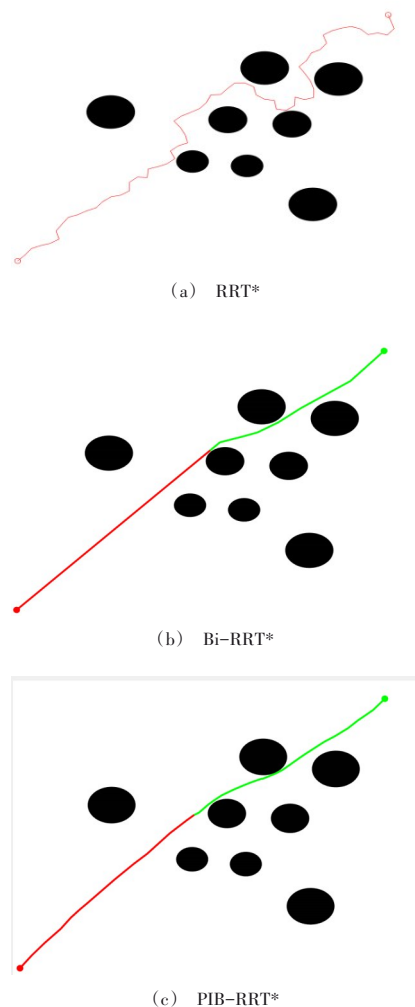


Fig. 11 Simulation in a simple environment

图11 在简单环境下的仿真

Table 1 Simulation experiment results of the algorithms in a simple environment

表1 算法在简单环境下的仿真实验结果		
算法	平均路径长度/m	平均运行时间/s
RRT*	1 191	465
Bi-RRT*	955	265
PIB-RRT*	940	61

图12可以看到,虽然 PIB-RRT*算法与 Bi-RRT*算法在路径长度上差别不大,但其规划出的路径平滑度较高,转弯处的角度小,而 RRT*算法的路径较为曲折。

3.3 特殊水域环境实验

设置真实的海面环境,3种算法的最终路径规划效果见图13,指标统计结果见表3。在真实的海面环境中,PIB-RRT*算法相较于其他两种算法所用时间更少,约为 RRT*算法的3%,Bi-RRT*算法的10.7%,并且规划出的路径长度最短。而且由于算法中加入了转角约束,所以拐角处的路径也较为平滑,更加符合 USV 海上航行对实时性和可行性的要求。

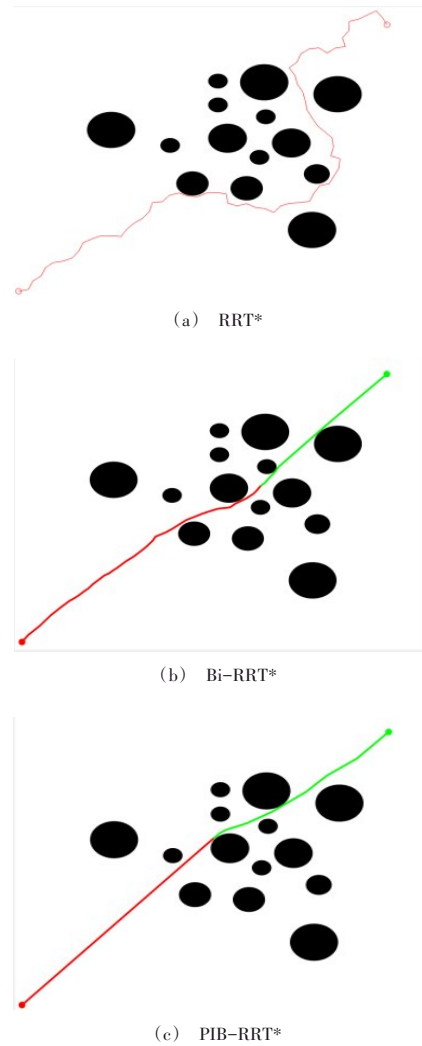


Fig. 12 Simulation in a complex environment
图 12 在复杂环境下的仿真

Table 2 Simulation experiment results of algorithms in a complex environment

表 2 算法在复杂环境下的仿真实验结果		
算法	平均路径长度/m	平均运行时间/s
RRT*	1 279	483
Bi-RRT*	956	257
PIB-RRT*	952	58

4 结语

本文改进了双向快速扩展随机树(Bi-RRT*)算法的随机点生成机制,在随机点生成过程中加入目标偏置策略,明确节点生成的方向性;加入节点(样本)的智能插入,进一步提高算法收敛到最优解的速度,以满足USV在航行时对实时性的要求;增加转向角约束,进一步平滑路径,使其能更好地满足USV的运动约束和动力约束。为验证改进算法PIB-RRT*在收敛速度、路径平滑方面的优越性,对PIB-RRT*算法、Bi-RRT*算法、RRT*算法分别在简单、复

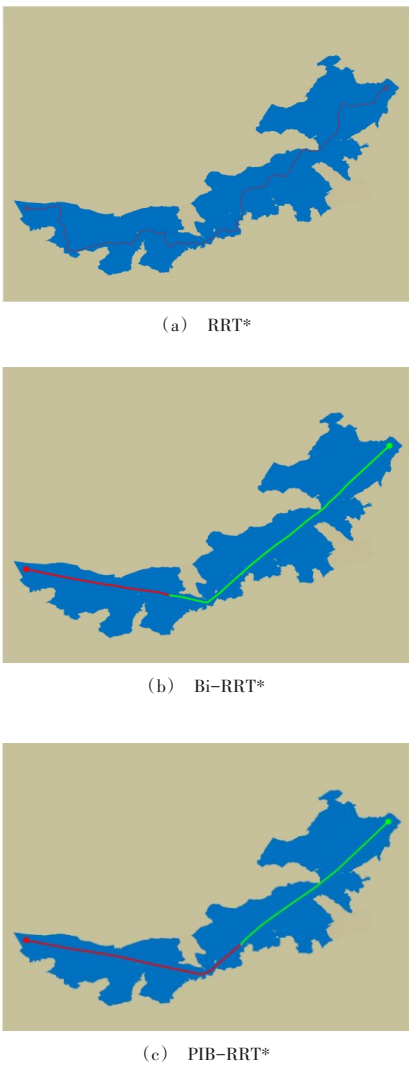


Fig. 13 Simulation in special water environment
图 13 在特殊水域环境下的仿真

Table 3 Simulation experiment results of algorithms in special water environment

表 3 算法在特殊水域环境下的仿真实验结果		
算法	平均路径长度/m	平均运行时间/s
RRT*	1 295	249
Bi-RRT*	998	79.8
PIB-RRT*	985	8.54

杂、特殊 3 种水域环境下进行仿真实验。结果显示,与改进前的算法相比,本文提出的目标偏置、智能节点插入和转向角约束可以提高收敛速度和路径质量,如提高路径平滑度、减少路径长度等。本文设计的算法可满足USV路径规划的需求,但未将波浪和海流等关键影响因素纳入其中。因此,如何在环境因素的干扰下进行USV路径规划将是下一步探讨的问题。

参考文献:

[1] ALEX R D , NOAH L , RICHARD J , et al. Long-term measurements of fish backscatter from Saildrone unmanned surface vehicles and comparison with observations from a noise-reduced research vessel [J]. ICES Journal

- of Marine Science, 2019, 76(7): 2459–2470.
- [2] HADI K A. Optimal control of the obstacle problem: optimality conditions [J]. IMA Journal of Mathematical Control & Information, 2006, 23(3): 325–334.
- [3] PLAKU E, BEKRIS K E, CHEN B Y, et al. Sampling-based roadmap of trees for parallel motion planning [J]. IEEE Transactions on Robotics, 2005, 21(4): 597–608.
- [4] JAILLET L, CORTÉS J, SIMÉON T. Sampling-based path planning on configuration-space costmaps [J]. IEEE Transactions on Robotics, 2010, 26(4): 635–646.
- [5] LIANG B, YANG X M. Research on a route planning method based on improved Dijkstra algorithm [J]. Informationization Research, 2020, 46(2): 13–16.
梁波, 杨新民. 一种基于改进型 Dijkstra 算法的路线规划方法研究 [J]. 信息化研究, 2020, 46(2): 13–16.
- [6] CHEN X. A review of research on path planning for mobile robots based on geneticalgorithms [J]. Science and Industry, 2023, 23(8): 274–278.
陈旭. 基于遗传算法的移动机器人路径规划研究综述 [J]. 科技和产业, 2023, 23(8): 274–278.
- [7] FU B, CHEN L, ZHOU Y, et al. An improved A* algorithm for the industrial robot path planning with high success rate and short length [J]. Robotics and Autonomous Systems, 2018, 106: 26–37.
- [8] ZHANG J, YU J, QU X, et al. Path planning for carrier aircraft based on geometry and Dijkstra's algorithm [C]//Proceedings of 2017 3rd IEEE International Conference on Control Science and Systems Engineering, 2017: 115–119.
- [9] DONALD B, LYNCH K, RUS D. Algorithmic and computational robotics [M]. Boca Raton: CRC Press, 2001.
- [10] JING H C, CAO Y M, GE C, et al. Three dimensional path planning for unmanned aerial vehicles based on hybrid strategy improved beetle optimization algorithm [J]. Modern Electronic Technology, 2024, 47(13): 144–152.
景会成, 曹育铭, 葛超, 等. 基于混合策略改进蜣螂优化算法的无人机三维路径规划 [J]. 现代电子技术, 2024, 47(13): 144–152.
- [11] CHRISTOF C, WACHSMUTH G. On second-order optimality conditions for optimal control problems governed by the obstacle problem [J]. Optimization, 2020, 70(10): 1–41.
- [12] ZHANG J, ZHANG G, PENG Z, et al. Position-based Dubins-RRT* path planning algorithm for autonomous surface vehicles [J]. Ocean Engineering, 2025, 324: 120702.
- [13] SHI G Y, ZHANG Y F, LIU J, et al. Path planning algorithm of unmanned surface vehicles based on Bi-RRT guided by artificial potential field [J]. Journal of Shanghai Maritime University, 2022, 43(4): 16–22.
史国友, 张一帆, 刘娇, 等. 基于人工势场法引导的 Bi-RRT 的水面无人艇路径规划算法 [J]. 上海海事大学学报, 2022, 43(4): 16–22.
- [14] HE J, XIA H P, LIU X Z. Research on improved methods based on RRT path planning algorithm [J]. Automotive Practical Technology, 2019 (22): 39–42.
何佳, 夏海鹏, 刘修知. 基于 RRT 路径规划算法的改进方法研究 [J]. 汽车实用技术, 2019(22): 39–42.
- [15] HE P. Research on collision avoidance path planning algorithm for complex waters ships [D]. Dalin: Dalian Maritime University, 2020.
何平. 复杂水域船舶避碰路径规划算法研究 [D]. 大连: 大连海事大学, 2020.
- [16] AYALEW W, MENEBO M, MERGA C, et al. Optimal path planning using bidirectional rapidly-exploring random tree star-dynamic window approach (BRRT*-DWA) with adaptive Monte Carlo localization (AM-CL) for mobile robot [J]. Engineering Research Express, 2024, 6(3): 035212.
- [17] LIU X F, SHEN Z J, LU P. Entry trajectory optimization by second-order cone programming [J]. Journal of Guidance, Control, and Dynamics, 2016, 39(2): 227–241.
- [18] YIN Q W. Robot arm path planning based on GB-APFB-RRT* algorithm [J]. Modular Machine Tool and Automation Processing Technology, 2023(10): 11–15.
尹庆文. 基于 GB-APFB-RRT*(*) 算法的机械臂路径规划 [J]. 组合机床与自动化加工技术, 2023(10): 11–15.
- [19] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning [J]. The International Journal of Robotics Research, 2011, 30(7): 846–894.
- [20] LE X L. Research and application of robot harvesting method for tomato skewers [D]. Guangzhou: South China University of Technology, 2021.
乐晓亮. 番茄串的机器人采收方法研究与应用 [D]. 广州: 华南理工大学, 2021.

(责任编辑: 黄健)