[引用格式]杨理驰,史圣哲,杨静雷,等. 基于改进 A* 算法的无人船自动靠泊路径规划[J]. 集美大学学报(自然科学版),2026,31(5):552-565. DOI:10.19715/j.jmuzr.2026.05.05.

基于改进 A* 算法的无人船自动靠泊路径规划

杨理驰[1]，史圣哲[2]，杨静雷[1]，俞万能[1]，廖卫强[1]

（1. 集美大学轮机工程学院，福建 厦门 361021；2. 中航工业集团特种飞行器研究所，湖北 荆门 448035）

[摘要] 针对传统 A* 算法在无人船自动靠泊路径规划中存在的冗余节点多、路径转折突兀以及终点航向难以满足靠泊姿态要求等问题，提出一种基于改进 A* 算法的靠泊路径规划方法。该方法在全局规划阶段引入 JPS 跳点机制与优化启发函数，以减少冗余节点扩展并提升搜索效率；在路径优化阶段结合贝塞尔曲线平滑处理与自适应末端控制点调整，实现靠泊末端航向的精确约束。仿真结果显示，在满足避障安全距离的前提下，与传统启发函数相比，新方法扩展节点数减少约35%，搜索时间缩短约18.5%，路径平滑度提升59.1%，靠泊终点航向角偏差由53.9°显著降低至5°以内，有效满足了无人船精确靠泊的姿态要求。实验结果验证了该方法在复杂港口环境中的适用性与鲁棒性。

[关键词] 无人船；自动靠泊；改进 A* 算法；姿态约束；路径平滑

[中图分类号] U 675.7；S 972.8 [文章编号] 1007－7405（2026）05－0552－14

Autonomous Berthing Path Planning of Unmanned Vessels Based on an Enhanced A* Algorithm

YANG Lichi[1], SHI Shengzhe[2], YANG Jinglei[1], YU Wanneng[1], LIAO Weiqiang[1]

(1. School of Marine Engineering, Jimei University, Xiamen 361021, China;
2. AVIC Special Vehicle Research Institute, Jinmen 448035, China)

Abstract: To address the issues of redundant nodes, abrupt path turns, and the difficulty of meeting berthing orientation requirements in traditional A* algorithms for unmanned surface vehicle automatic berthing, this paper proposes an improved A* path planning method. At the global planning stage, the jump point search mechanism and an optimized heuristic function are introduced to reduce redundant node expansion and enhance search efficiency. At the path optimization stage, Bézier curve smoothing is combined with adaptive terminal control point adjustment to achieve precise constraints on the final berthing heading. Simulation results demonstrate that, while maintaining a safe obstacle avoidance distance, the proposed method reduces the number of expanded nodes by at least 35% and search time by approximately 18.5% compared to the traditional heuristic. Furthermore, path smoothness is improved by 59.1%, and the final berthing heading angle deviation is significantly reduced from 53.9° to within 5°, effectively meeting the attitude requirements for precise USV berthing. Experimental results

[收稿日期] 2025－11－29
[基金项目] 航空科学基金项目（20240023132001）
[通信作者] 史圣哲，高级工程师，主要从事水动力性能、测量不确定度分析等方向研究，E-mail:shishengzhe05011232@126.com；杨静雷，博士，副教授，主要从事高性能船舶总体设计与智能控制技术方向研究。

verify the applicability and robustness of the proposed method in complex port environments.

Keywords: unmanned vessel; automatic berthing; improved A* algorithm; attitude constraints; path smoothing

0　引言

船舶在靠泊过程中，受操作复杂、人为失误、环境干扰等因素的影响，容易发生事故。为降低靠泊事故的发生率，无人船自动靠泊技术应运而生。无人船路径规划算法主要有 Dijkstra 算法[1]、A* 算法[2]、蚁群算法[3]和快速扩展随机树（RRT）[4]等。其中，A* 算法的高效性、灵活性和广泛性[5-6]使其成为路径规划算法中的首选。然而，采用传统 A* 算法规划无人船靠泊路径，会出现冗余节点多、路径转折突兀不够平滑、终点航向难以精确匹配码头等问题，这制约了无人船靠泊路径的航行性能和实用性。

近年来，无人船路径规划研究取得了突破性进展。殷宏亮等[7]在传统 A* 算法上引入了碰撞约束机制，结合目标导向和跳跃点技术，优化了 A* 算法，缩短了路径长度。朱泺谕等[8]提出了一种改进的 A* 算法，通过设计奖惩机制优化路径选择，增加路径的平滑性。杨兰英等[9]结合 A* 算法的优势，提出了一种自适应寻路蚁群算法，使用传统 A* 算法规划的路线作为蚁群算法的初始迭代数据，解决蚁群算法初次收敛速度慢的问题。马自勇等[10]整合改进 A* 算法和动态窗口法提出新型路径规划算法，用自适应参数因子和安全半径路径优劣评价函数优化全局规划中的冗余节点。方祝平等[11]结合改进 A* 算法和速度障碍物法提出实时动态避障算法，引入基于 ESDF 地图的自适应节点拓展步长策略，有效减少了路径搜索的冗余节点。

尽管上述研究为无人船靠泊路径规划提供了有益参考，但现有自动靠泊路径规划方法多为由远及近的全局路径规划，其规划尺度通常为数公里却未考虑路径终点的朝向，且仍存在路径转折过多、搜索效率低及平滑度不足等问题。为此，本文针对这些问题，对传统 A* 算法进行了改进，以提高无人船自动靠泊路径规划的安全性、平滑性和靠泊精度，为无人船自动靠泊提供更加高效可靠的路径规划方法。

1　算法改进

本改进算法的框架如图 1 所示。

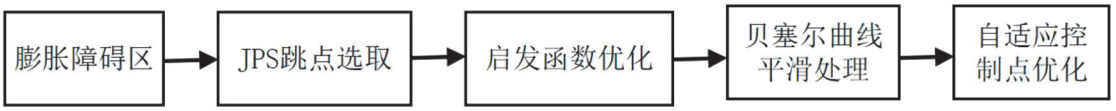


图 1　算法框架图

Fig.1　Algorithm framework diagram

首先，本算法对膨胀障碍区进行处理；接着，对传统 A* 算法进行了改进，通过引入 JPS 跳点选取机制[12]显著提升了路径搜索的效率，并通过优化启发函数确保规划出的路径能够满足靠泊的具体需求；然后，利用贝塞尔曲线[13]对生成的路径进行平滑处理以方便船舶的跟踪，并通过自适应优化贝塞尔曲线的控制点使路径更加精确地符合靠泊路

径的要求。

1.1 障碍区膨胀化

为确保无人船靠泊路径距离岸基等障碍物有足够的安全距离，需要对禁行区进行膨胀处理[14]。首先，获取原始栅格地图，遍历栅格图的每个节点，对障碍物区域进行膨胀。然后，以该节点为中心，将周围安全距离范围内的点标记为障碍，将扩展后的栅格地图应用于仿真环境中。安全距离 $D_S = 1.5$ 的扩展地图如图 2 所示。

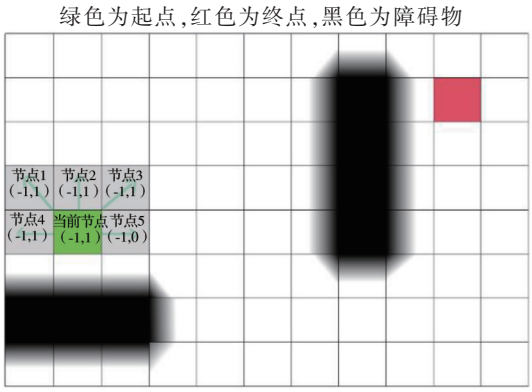


图 2 扩展障碍区栅格图

Fig.2 Grid map of expanded obstacle region

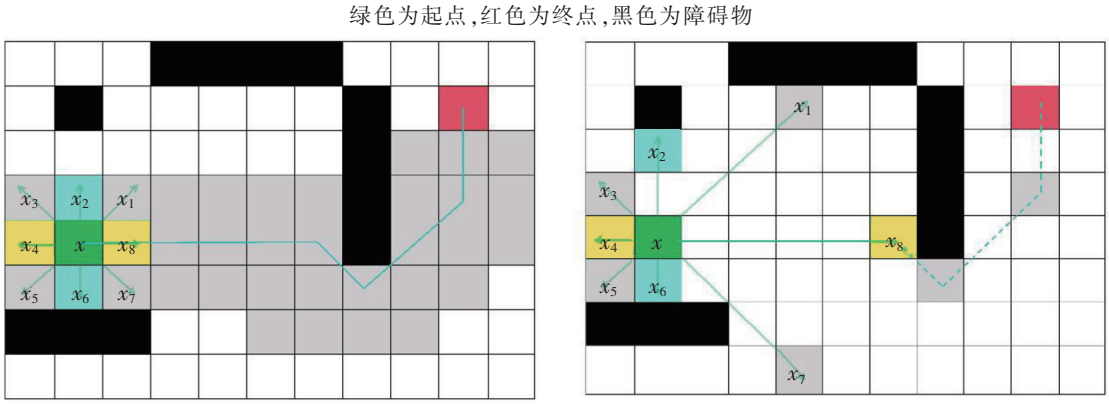
1.2 A* 算法中跳跃点选取

引入 JPS[12] 作为 A* 算法的优化寻路算法，通过跳过不必要的节点扩展，提高路径规划的效率。其核心思想是：在网格上，当某一方向上的连续空白节点可以直接到达某个目标节点时，可以跳过这些中间节点，直接跳到更远的关键节点，从而减少搜索空间和计算量。JPS 寻路算法通过以下公式定义跳跃规则：

$$J(x,y,p_x,p_y) = \begin{cases} (x,y) & \text{无障碍物且无可行跳跃点，} \\ \text{None} & \text{当前方向不可行，} \\ J(x+p_x,y+p_y,p_x,p_y) & \text{在当前方向跳跃。} \end{cases}$$

其中：$J(x,y,p_x,p_y)$ 表示从当前节点 (x,y) 按照方向 (p_x,p_y) 跳跃到下一个有效节点的操作。如果当前位置存在障碍物或不可行的方向，则跳跃失败。

本文采用这种智能跳跃的路径搜索方法。通过在搜索过程中跳过一些不必要的节点直接到下一个关键节点，避免了对每一个节点进行逐一扩展的计算。扩展子节点跳跃的 A* 算法规划路径如图 3 所示。



a) 传统 A* 算法的节点扩散 b) 本改进 A* 算法的节点扩散

图 3 基于扩展子节点跳跃的 A* 算法规划路径

Fig.3 Planned path of A* algorithm with expanded node-jumping strategy

1.3 启发函数优化

启发函数 $h(n)$ 有三种常见的计算方式：曼哈顿距离[15]、欧几里得距离[16]、切比雪

夫距离[17]。使用曼哈顿距离或欧几里得距离通常可以得到较为理想的路径，但在考虑实际运动方式时，可能存在不足。为了更好地适应不同方向的路径规划需求，本文将三种计算方式进行结合，提出一种改进的启发函数，旨在提高路径规划的准确性和效率。具体的改进公式为：

$$h(n) = |x_1 - x_2| + |y_1 - y_2| + (\sqrt{2} - 2) \cdot \min(|x_1 - x_2|, |y_1 - y_2|)。$$

其中：(x_1, y_1)、(x_2, y_2) 分别为无人船移动前、后的位置。

该启发函数在曼哈顿距离的基础上引入对角线修正项，其本质是对切比雪夫距离[17]在连续空间中的加权表达。公式中前两项 $|x_1 - x_2| + |y_1 - y_2|$ 表示无人船在栅格环境下沿主方向移动所需的最小累计航行距离，而修正项 $(\sqrt{2} - 2) \cdot \min(|x_1 - x_2|, |y_1 - y_2|)$ 则用于补偿对角方向移动相较于两次正交移动的实际距离差。该修正使启发函数在存在对角可行路径时，能够更准确地估计剩余航程，避免因路径代价估计偏大而产生不必要的节点扩展。

该启发函数反映了无人船在路径规划过程中优先通过对角方向运动同步减小横向与纵向位移误差，其剩余位移通过主方向运动逐步消除的路径生成特性。该特性符合无人船在自动靠泊阶段尽量减少航向频繁切换、保持航向变化连续和平缓的运动规律。因此，在保证启发函数可接受性与一致性的前提下，该启发函数形式更有利于生成转折次数较少且方向变化平滑的靠泊路径，为后续基于船舶动力学模型的轨迹跟踪与控制提供更加合理的参考轨迹。

1.4　贝塞尔曲线平滑处理

本文引入贝塞尔曲线对 A* 算法生成的路径进行平滑优化。首先，利用 A* 算法在栅格地图中搜索无人船从起点到目标点的最短路径，生成由离散节点组成的初始路径；然后，提取 A* 路径中的所有关键节点作为控制点，计算贝塞尔曲线，生成连续平滑的路径；最后，对生成的贝塞尔路径进行碰撞检测和优化，遍历贝塞尔路径上的每一个点，检查是否有在标定的障碍物区域，若有则采用局部重构的优化策略，对发生碰撞的路径段回退到原 A* 路径，必要时可适当增大障碍区膨胀半径以满足安全距离要求，确保其避开障碍物并满足靠泊精度要求。通过上述步骤，实现从粗粒度路径搜索到高精度路径平滑的完整规划过程。

改进算法实现的主要伪代码展示为如下的算法 1 和算法 2。

1）算法 1

```
Input: path, astar_ start, astar_ target, grid_ map
Output: local_ targets (jump points)
1. local_ targets ← [ ]
2. path ← remove_ first_ and_ last (path)
3. directions ← calculate_ directions (path)
4. angle_ changes ← calculate_ angle_ changes (directions)
5. i ← 1
6. for each (point, angle_ change) in zip (path, angle_ changes) do
7. if angle_ change = 0 then
```

```
8. continue
9. else if point aligned with astar_ start then
10.            add_ to_ local_ targets (point)
11. else if point aligned with astar_ target then
12. continue
13. else
14.            # JPS
15. if has_ forced_ neighbor (point, grid_ map) then
16.                add_ to_ local_ targets (point)
17. else if is_ diagonal_ turn (point, directions) then
18.                add_ to_ local_ targets (point)
19. else if i is even then
20.                add_ to_ local_ targets (point)
21.                i ← i + 1
22. else
23.                i ← i + 1
24. end if
25. end if
26. end for
27. local_ targets ← [astar_ start] + local_ targets + [astar_ target]
28. return local_ targets
```

2）算法 2

```
Input: path_ points (from JPS - A*), step_ size
Output: smoothed_ path (bx, by)
1. control_ points ← path_ points
2. n ← length (control_ points) - 1
3. smoothed_ path ← [ ]
4. binomial_ coeffs ← calculate_ binomial_ coefficients (n)
5. for t from 0 to 1 step step_ size do
6.        x_ sum, y_ sum ← 0, 0
7.       for i from 0 to n do
8.            B ← binomial_ coeffs [i] * (t^i) * ( (1 - t) ^ (n - i))
9.            x_ sum ← x_ sum + control_ points [i]. x * B
10.            y_ sum ← y_ sum + control_ points [i]. y * B
11.       end for
12.       smoothed_ path. append ( (x_ sum, y_ sum))
13. end for
14. if (1, 1) not in smoothed_ path then
15.        smoothed_ path. append (control_ points [ -1])    # 添加终点
16. end if
17. return smoothed_ path
```

1.5 自适应靠泊末端控制点

基于 A* 算法的贝塞尔曲线优化方法虽然有效，但在实际应用中发现，当船舶到达泊位时，其航向角并未完全与码头平行，存在一些偏差。因此，针对贝塞尔曲线靠近末端的部分控制点，需要进行自适应优化处理。这里设置靠泊终点坐标为 $P_n = (x_n, y_n)$，码头的方向向量为 $\boldsymbol{T} = (\boldsymbol{T}_x, \boldsymbol{T}_y)$，贝塞尔曲线的控制点为 $C_0, C_1, \cdots, C_i, \cdots, C_{n-1}, C_n$。为

了更灵活地适应不同码头的靠泊需求，通过选取第 i 个控制点 C_i 来进行自适应调整，使贝塞尔曲线在不同的靠泊环境下更加灵活和高效。其中，第 i 个控制点选取为靠近终点且对贝塞尔曲线末端切线方向起决定作用的关键控制点，其位置由终点向前回溯确定。贝塞尔曲线末尾的切线向量 $\boldsymbol{T}_{\text{end}}$ 是由 C_i 和 P_n 之间的差值决定的。切线方向可以表示为

$$\boldsymbol{T}_{\text{end}} = n(P_n - C_i)。\tag{1}$$

其中：n 为贝塞尔曲线的阶数。为了使贝塞尔曲线的末尾方向与码头的方向平行，需调整 C_i 使得切线向量 $\boldsymbol{T}_{\text{end}}$ 和目标方向 $\boldsymbol{T}$ 之间的夹角趋近于 0，即 $\boldsymbol{T}_{\text{end}} \| \boldsymbol{T}$。

调整后控制点 C'_i 的计算公式为

$$C'_i = P_n - \lambda \frac{P_n - C_{i-1}}{\|P_n - C_{i-1}\|} \cdot \|P_n - C_{i-1}\|。\tag{2}$$

其中：C_{i-1} 是贝塞尔曲线中 C_i 之前的控制点；λ 是缩放因子，控制调整的幅度，其值取决于实际场景；$\frac{P_n - C_{i-1}}{\|P_n - C_{i-1}\|}$ 为单位向量，表示终点 P_n 到控制点 C_{i-1} 的方向。

还需要对贝塞尔曲线 C_i 后的控制点进行逐步修正，确保曲线的末尾部分与码头方向平行。对于 $C_{i+1}, C_{i+2}, \cdots, C_{n-1}$，需要确保它们与调整后的控制点保持适当的关系。用 C'_j 表示修正后的控制点，其中 $j = i, i+1, \cdots, n-1$。对每个修正后的控制点 C'_j 有

$$C'_j = C_j + \alpha(C'_i - C_j)。\tag{3}$$

即，将 C_j 沿着线段 C_j 到 C'_i 方向移动 α 倍的距离。其物理上可直观解释为：当末端关键控制点 C_i 被调整为 C'_i 以满足终端切线对齐后，为避免末端仅在单点发生突变，需要将末端段的控制点进行协同调整；通过对后续控制点施加逐步收敛的牵引，使控制多边形末端段的方向与尺度呈现一致性，从而使曲线末端切线方向更稳定、曲率变化更连续。

α 为末端约束影响的传播率，决定了修正的幅度，其值通常在 [0,1] 之间变化。对控制点 C_j 施加一个指向 C'_i 的吸引或牵引项，使末端控制点簇在几何上形成一致的方向趋势，从而保证曲线末端段的切线方向与曲率过渡更连续。$\alpha = 1$ 表示不传播，除 C_i 外其余控制点保持原状，易产生末端局部形状不协调；$\alpha = 0$ 表示完全对齐，所有相关控制点强制向 C'_i 收缩，可能导致整体末端形态被过度拉直甚至扭曲；$0 < \alpha < 1$ 表示在保留原有曲线整体趋势的同时，实现末端方向约束的平滑过渡。因此，α 值选取为终点航向对齐—末端平滑性—整体形状保持之间的折中位置。一般 α 取值范围为 0.3 ~ 0.7 即可在大多数靠泊场景中取得较稳健的效果。

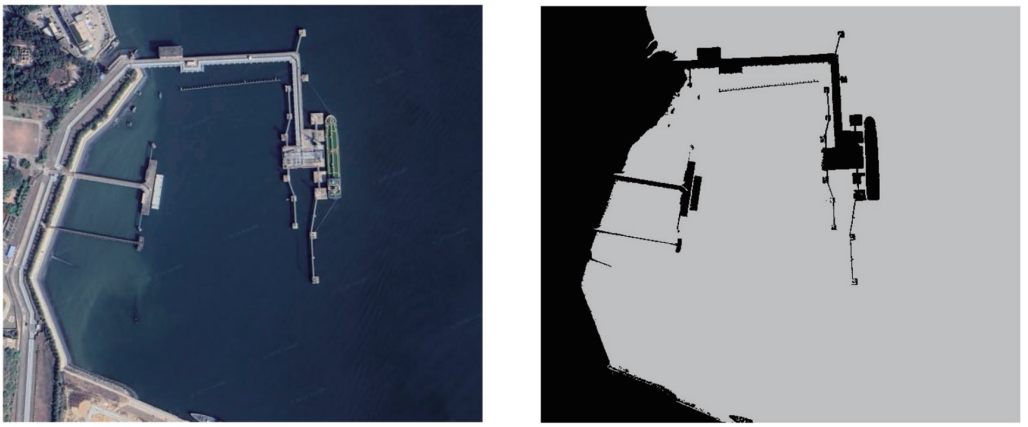
通过以上步骤，所有修正后的控制点 $C_0, C_1, \cdots, C_{i-1}, C'_i, C'_{i+1}, \cdots, C'_{n-1}, P_n$，可使贝塞尔曲线的末尾切线与码头方向平行，并且路径在靠泊时更加平滑、稳定。

2　仿真实验验证

2.1　仿真场景建模

选择厦门港 A 区作为仿真场景，选定的地图区域长 700 m、宽 700 m。在 Python 中将地图上厦门港 A 区的实际卫星云图转换为二维灰度图。在数学建模中，参照文献[18]，通过定义二维空间域上的离散二元矩阵实现环境表征。设置栅格化地图的映射维

度为 700 × 700 的矩阵 $\boldsymbol{M}$，即 $\boldsymbol{M} \in \{0,1\}_{700\times700}$。其中矩阵元素 $M_{ij} = 0$ 表示可航行区域，$M_{ij} = 1$ 表示障碍物占据区域。为确保船舶靠泊安全，若某栅格单元被障碍物部分覆盖，则 M_{ij}统一赋值为 1。本文地图范围为（24°27. 22′N，118°02. 24′W）到（24°27. 00′N，118°02. 42′W）。创建的 700 × 700 栅格地图如图 4 所示。



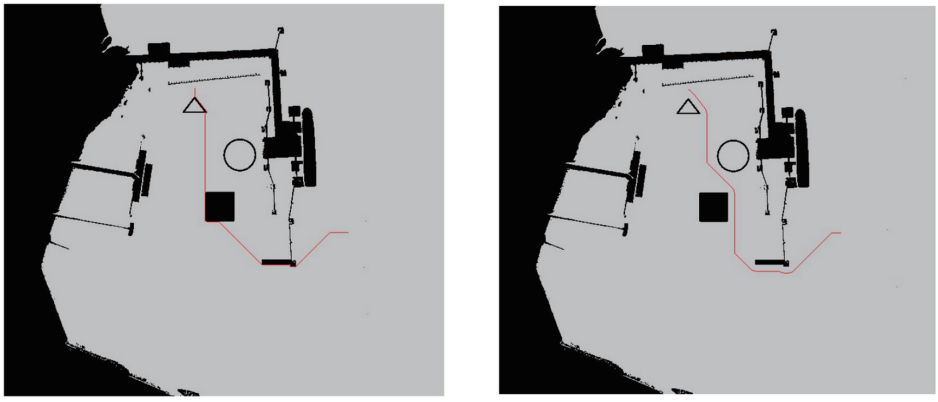
a) 卫星云图　　b) 二值化后的栅格图

图 4　厦门港 A 区的卫星云图和栅格地图

Fig.4　Satellite cloud image and grid map of Xiamen Port Area A

2. 2　障碍物膨胀的仿真结果与分析

为了验证对障碍物保持足够安全距离的有效性，本文在原始地图上手动添加若干障碍物。靠泊起点与泊位终点的分布情况如图 2 仿真地图中起点与目标点分布所示。图 5a 和图 5b 所示的分别是未经膨胀障碍物处理的 A* 路径和经 $D_S = 10$ 膨胀障碍物处理后的 A* 路径。由图 5b 可以看出，经过膨胀障碍物处理后，路径与障碍物之间保持足够的安全距离，避免船舶在靠泊过程中与障碍物发生碰撞的风险。相比之下，图 5a 中的路径虽然更短，但由于未考虑障碍物的膨胀，路径过于靠近障碍物，存在潜在的安全隐患。



a) 未膨胀障碍物路径　　b) D_s=10 膨胀障碍物路径

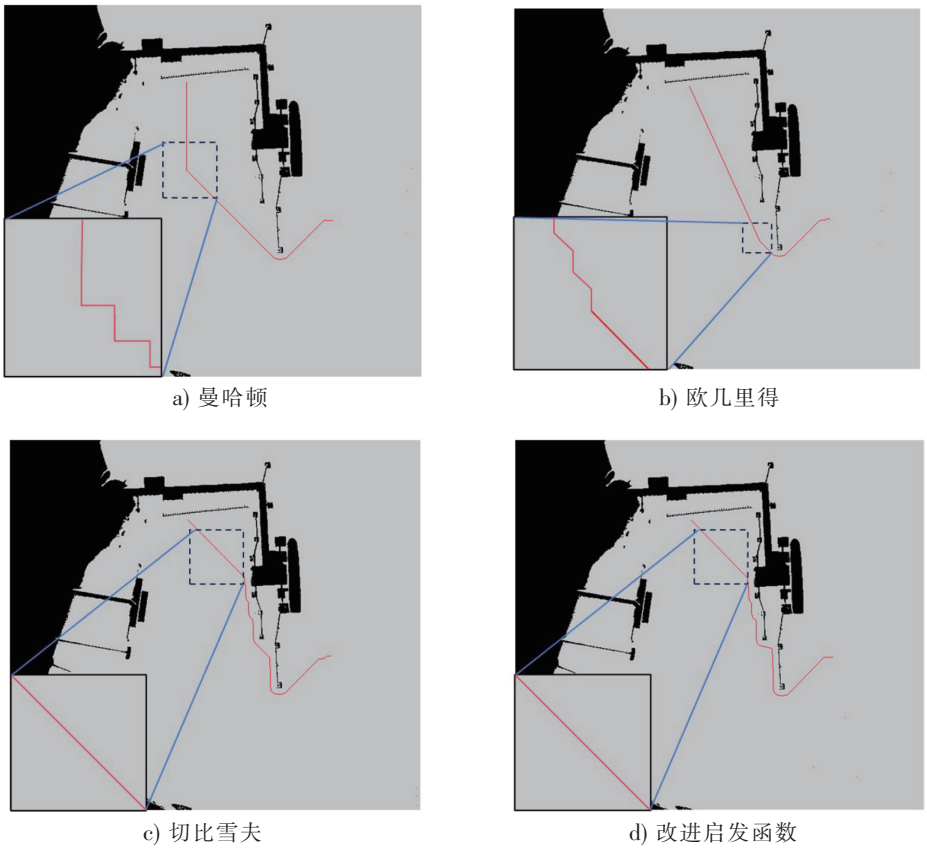
图 5　靠泊路径

Fig.5　Berthing path

2. 3　改进 A* 算法的仿真结果与分析

在仿真地图中，引入 JPS 跳点机制并结合改进的启发函数探索路径。用本改进算法与曼哈顿、欧几里得、切比雪夫三种传统启发函数算法得出的靠泊路径如图 6 所示。本

改进算法的靠泊路径（见图 6d）比传统切比雪夫启发函数算法的靠泊路径（见图 6c），在路径平滑性方面有所提升并且更贴合船舶运动学特性。



a) 曼哈顿　　b) 欧几里得

c) 切比雪夫　　d) 改进启发函数

图 6　传统启发函数与本改进算法对比

Fig.6　Comparison between traditional heuristic function and this improved algorithm

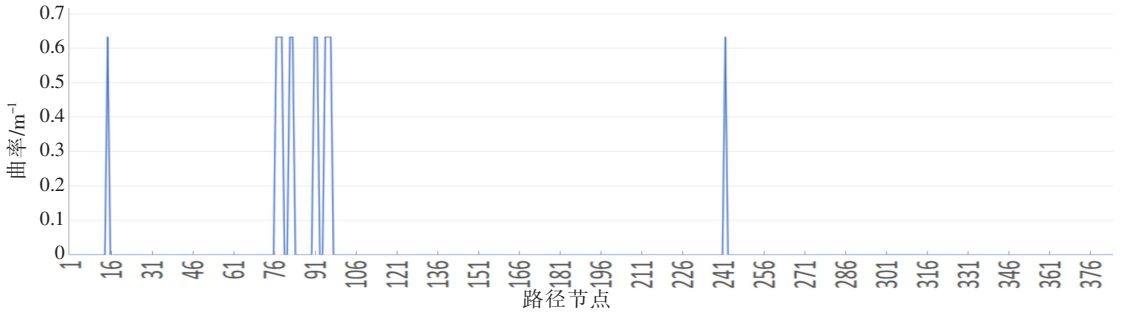
四种算法的路径曲率如图 7 所示。从图 7d 中可以看出，相较于其他启发函数，本改进算法显著减少了路径的曲率波动。路径曲率的稳定性有所提高，证明本改进算法能更好地处理路径规划中的平滑性问题，减少路径的急转弯，从而提升无人船跟踪路径时航行的效率与安全性。

四种算法的参数见表 1。从表 1 中可以看出，在路径长度相同的情况下，改进启发函数在路径规划效率与终点航向控制方面展现出显著优势。传统曼哈顿距离算法虽然扩展节点数最少、搜索时间最短，但其终点航向角偏差为 261.1°，未能满足靠泊角度约束要求，表明其在复杂转向场景中适应性不足。欧几里得距离算法因过度扩展节点导致搜索时间增加，仍无法满足终点航向要求。相比之下，切比雪夫算法虽成功将终点航向角偏差调整为 53.9°，但其扩展节点数和搜索时间显著偏高，暴露出计算效率的不足。本改进 A* 算法通过融合曼哈顿距离算法的障碍规避能力与修正项的对角线距离调节机制，在保证满足约束终点航向角偏差较小的同时，比切比雪夫函数算法扩展节点数降低 35%，搜索时间优化 18.5%。本改进算法在确保路径安全性的前提下，既模拟船只实际运动中小转向的需求，避免路径过度弯曲，又减少无效节点扩展，从而在计算效率与运动控制精度之间实现平衡。因此，本改进 A* 算法在复杂水域路径规划场景中更具实用价值。

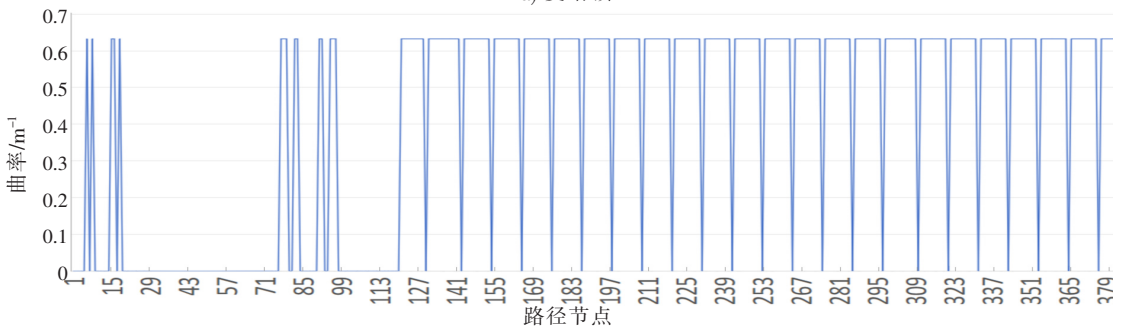
表 1 传统启发函数与改进启发函数参数对比

Tab. 1 Comparison between traditional heuristic function and improved heuristic function

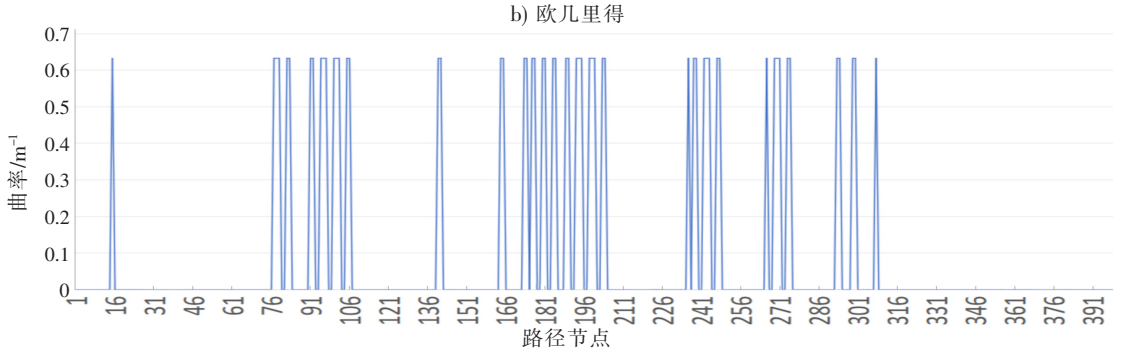
算法	路径长度/m	扩展节点数/个	搜索时间/s	终点航向与泊位偏差	平均曲率/m^{-1}
曼哈顿	474.40	40 200	0.5014	261.1°	0.0197
欧几里得	474.40	61 942	0.7664	68.9°	0.4227
切比雪夫	474.40	79 606	1.1384	53.9°	0.0895
改进 A^* 算法	474.40	52 143	0.9283	53.9°	0.0442



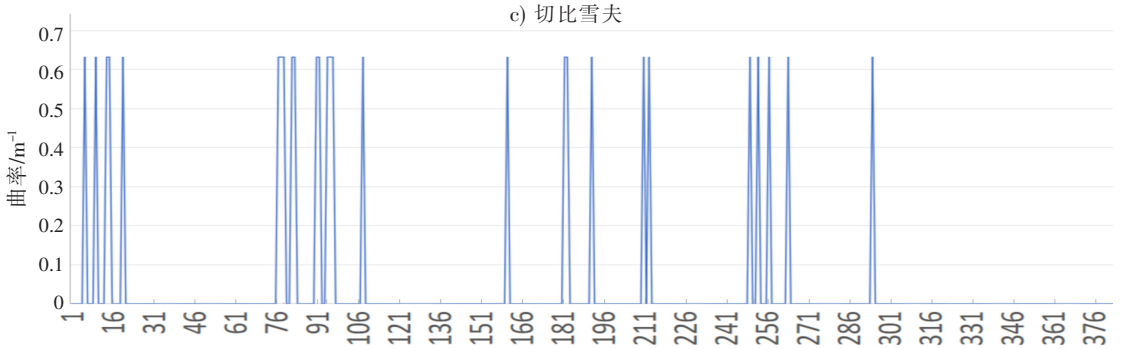
a) 曼哈顿



b) 欧几里得



c) 切比雪夫



d) 本改进算法

图 7 路径曲率对比

Fig.7 Path curvature comparison

2.4　自适应控制点仿真结果

将改进 A* 算法中的节点作为控制点输入到贝塞尔曲线中进行平滑处理，得到的路径如图 8 所示。然而，路径终点的航向角仍未完全与泊位角平行，存在一定偏差。为了解决这一问题，采用自适应控制点优化方法，调整贝塞尔曲线末端的控制点，从而进一步优化路径，使得船舶能够更加精确地与码头朝向平行。针对该码头，缩放因子设定为 1，平滑因子设定为 0.5，贝塞尔曲线自适应控制点起点为（125，352），最终的优化靠泊路径如图 9 所示，路径曲率对比如图 10 所示，仿真数据见表 2。

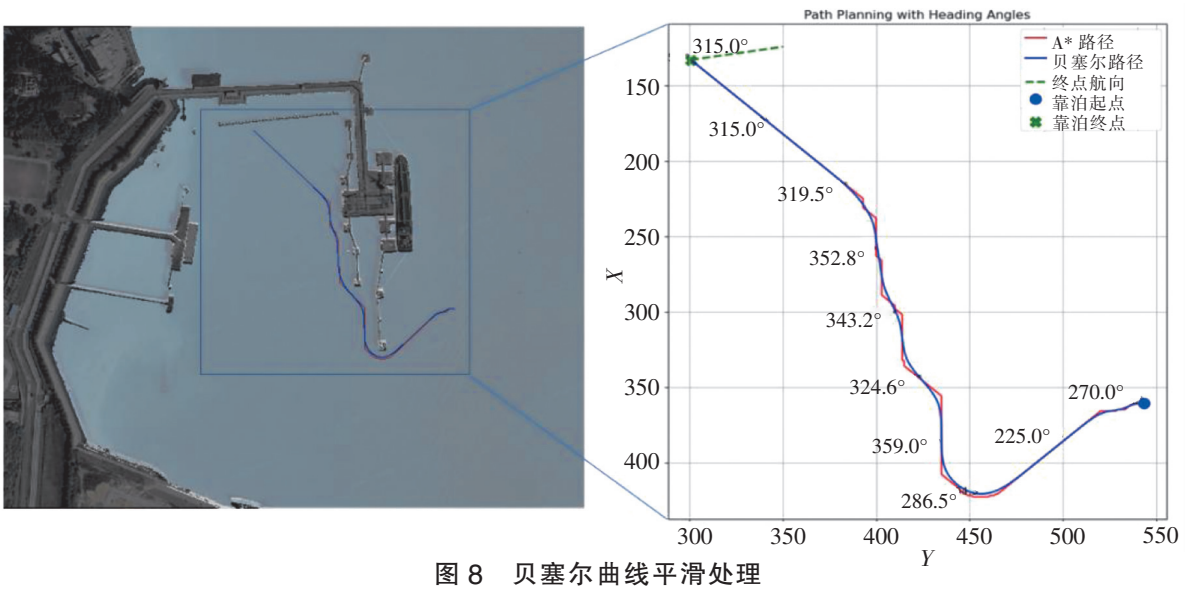


图 8　贝塞尔曲线平滑处理

Fig.8　Bezier curve smoothing process

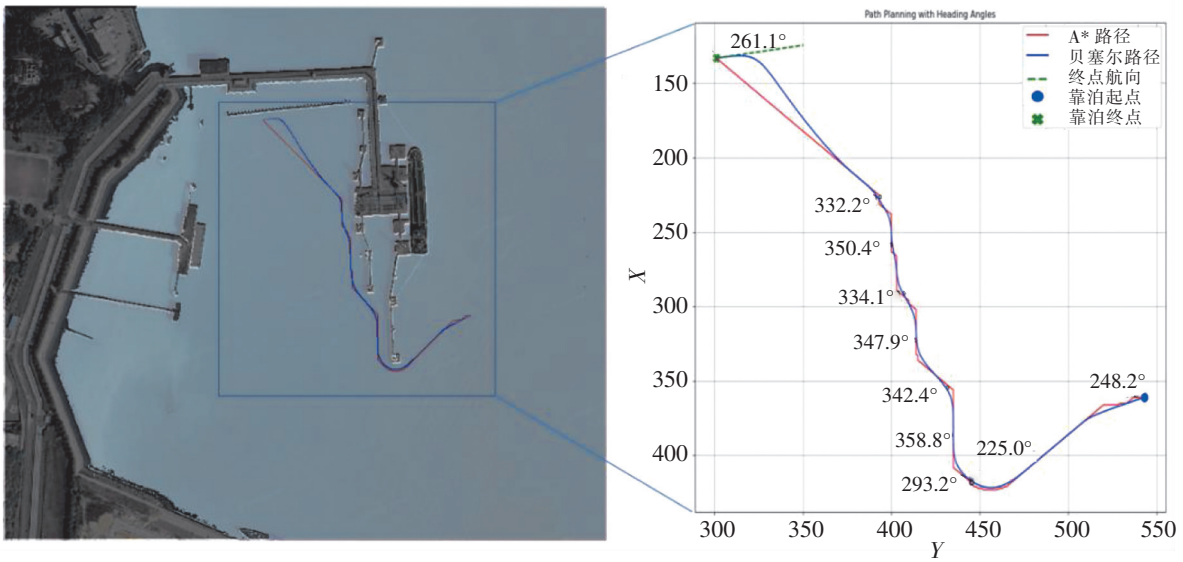


图 9　改进贝塞尔曲线平滑处理

Fig.9　Improved Bezier curve smoothing process

从图 10 可以看出，通过自适应优化末端控制点，贝塞尔曲线成功消除了急转弯和不平滑的问题，显著减少了路径曲率的波动。从表 2 的数据可以看出，相较于已优化的 A* 路径，改进后的贝塞尔曲线路径长度减少了 1.6%，平滑度提升了 58.8%。优化后的

路径更平稳，减少了不必要的航向调整，提高了航行效率和安全性。靠泊终点航向角偏差由 35.0°调整至 1.1°，生成符合无人船靠泊作业要求的行驶路径。改进后的路径对复杂港口环境适应性更强，能为船舶提供稳定的导航指引，确保靠泊顺利进行。

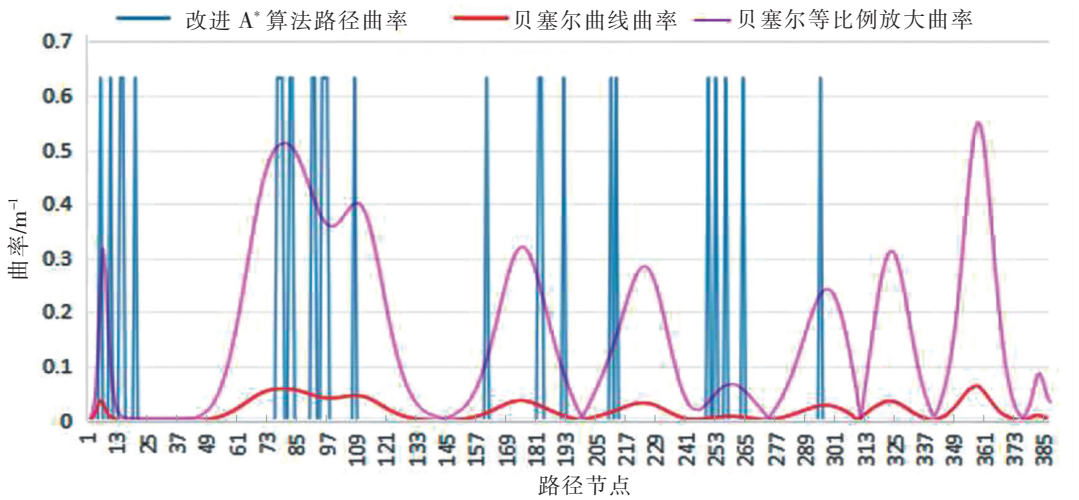


图 10 改进算法靠泊路径曲率对比

Fig.10 Curvature comparison of berthing path using improved algorithm

表 2 厦门港 A 区仿真数据

Tab. 2 Simulation data of Xiamen Port Area A

算法	路径长度/m	最大曲率/m^{-1}	平均曲率/m^{-1}	终点航向角偏差/(°)
改进 A* 算法路径	474.40	0.6324	0.0442	35.0
高阶贝塞尔路径	466.78	0.0809	0.0182	1.1

2.5 算法适用性验证

为了验证该算法在不同码头靠泊路径规划中的适用性，本研究靠泊路径的仿真验证实验选择在两个码头进行。表 3 为仿真地图的起点与目标点分布。

表 3 仿真地图起点与目标点分布

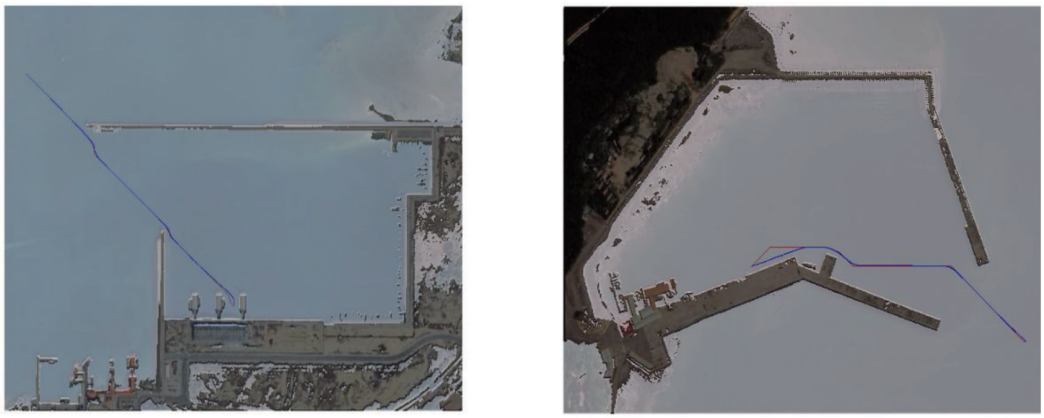
Tab. 3 Distribution of start and goal points in simulation maps

项目	地图 1	地图 2
实际经纬度范围	(24°25.38′N,118°17.05′W)—(24°24.55′N,118°18.00′W)	(24°25.44′N,118°15.44′W)—(24°25.28′N,118°16.08′W)
仿真地图靠泊起点	(104,33)	(422,609)
仿真地图泊位终点	(459,350)	(328,247)

本文路径规划算法在这两个地图上规划的靠泊路径及其曲率对比如图 11 和图 12 所示，对应地图路径上关键航向角如图 13 所示，具体的仿真数据见表 4 ~5。

由表 4 ~5 的数据可见，本改进 A* 算法在不同码头的靠泊终点朝向上均能满足靠泊航向角的要求。结合优化后的贝塞尔算法，路径长度进一步缩短，且平滑度得到显著提升。在地图 1 中，改进后的贝塞尔曲线路径长度减少了 1%，平滑度提升了 85%，终点航向角偏差减小至 5.9°；在地图 2 中，改进后的贝塞尔曲线路径长度减少了 2.9%，平

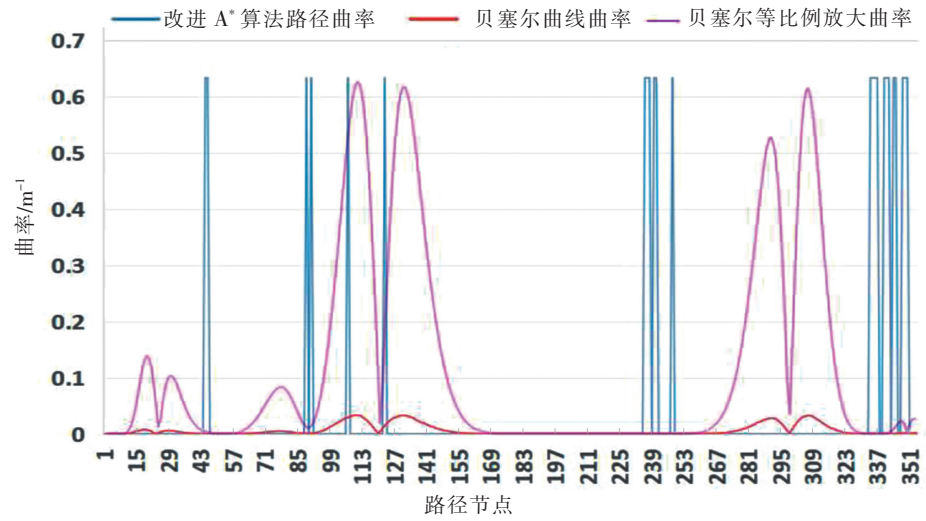
滑度提升了 76%，终点航向角偏差减小至 2.4°。这些结果表明，本文提出的算法在不同码头环境中具有较强的鲁棒性。



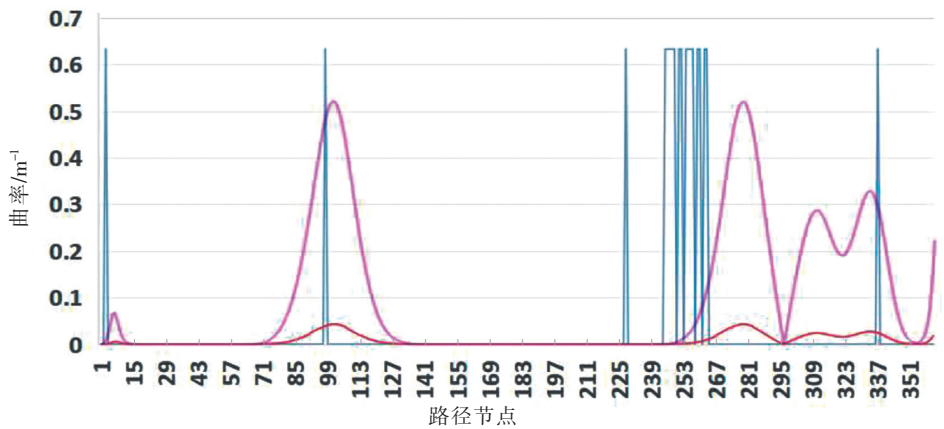
a) 地图 1 靠泊路径　　b) 地图 2 靠泊路径

图 11　不同地图靠泊路径

Fig.11　Berthing paths in different map scenarios



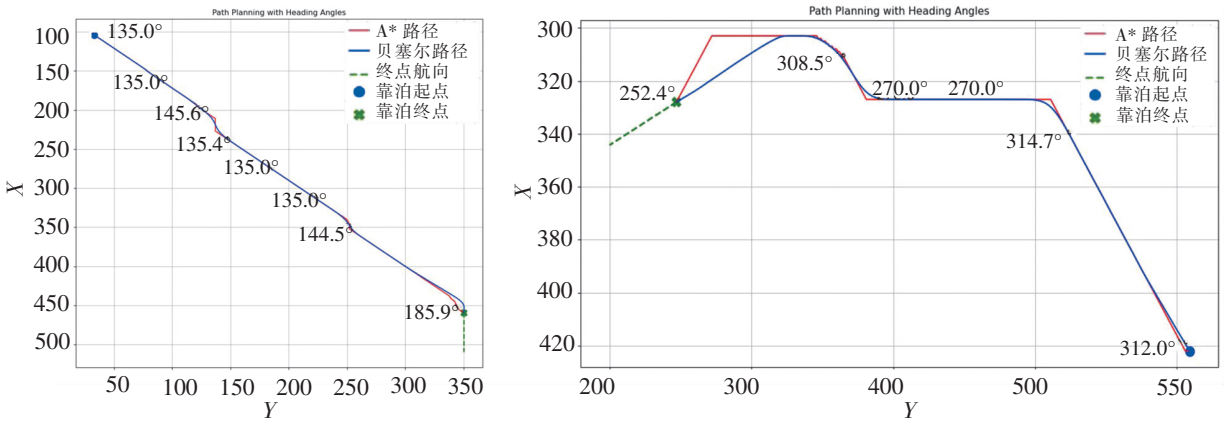
a）地图 1 靠泊路径曲率对比



b) 地图 2 靠泊路径曲率对比

图 12　改进算法在不同地图上的靠泊路径曲率对比

Fig.12　Curvature comparison of different berthing paths using improved algorithm



a) 地图 1 关键航向角路径图　　b) 地图 2 关键航向角路径图

图 13　关键航向角路径图

Fig.13　Key heading-angle trajectory diagram

表 4　不同地图的仿真数据

Tab. 4　Simulation data under different map scenarios

项目	地图 1	地图 2
自适应控制点起点	(430,353)	(315,288)
搜索时间/s	0.1775	0.7178
扩展节点/个	11 827	34 171
A*终点航向角偏差	45°	25°
贝塞尔终点航向角偏差	5.9°	2.4°

表 5　实验数据

Tab. 5　Experimental data

地图	算法	路径长度/m	最大曲率/m^{-1}	平均曲率/m^{-1}
地图 1	改进 A* 算法	486.31	0.6325	0.0429
	高阶贝塞尔	481.40	0.0857	0.0061
地图 2	改进 A* 算法	421.65	0.6325	0.0333
	高阶贝塞尔	409.50	0.0429	0.0080

3　结论

本文针对传统 A* 算法在靠泊路径规划中的冗余节点多，以及到达泊位终点时难以满足靠泊终点航向角的问题，通过引入 JPS 跳点选取机制和优化启发函数，并结合贝塞尔曲线路径平滑处理，以及自适应末端控制点调整，有效提升了路径规划的效率和航向。具体为：

1）贝塞尔曲线优化后的路径长度减少 1.6%，路径曲率波动显著降低，平滑度提升了 58.8%。即，优化后的路径更平稳，减少了不必要的航向调整，提高了航行效率和安全性。

2）通过调整控制点位置，使路径末端切线与码头方向平行，确保船舶靠泊时的精确姿态。自适应调整控制点后的路径终点航向角与泊位偏差趋近于 0°。

3）在不同码头环境中，改进 A* 算法结合贝塞尔曲线优化后，路径长度减少 1% ~ 2.9%，路径平滑度提升 76% ~85%，靠泊终点航向角偏差均趋近于 0°。

[参 考 文 献]

[1] Wang L, Zhu D, Pang W, et al. A survey of underwater search for multi-target using multi-AUV: task allocation, path planning, and formation control[J]. Ocean Engineering, 2023, 278: 114393.

[2] Zhu Y, Zhang G, Chu R, et al. Research on escape route planning analysis in forest fire scenes based on the improved A* algorithm[J]. Ecological Indicators, 2024, 166: 112355.

[3]Zhao J, Deng C, Yu H, et al. Path planning of unmanned vehicles based on adaptive particle swarm optimization algorithm [J]. Computer Communications, 2024, 216: 112-129.

[4]殷宏亮,朱洪波. 融合目标导向和跳跃点改进 A* 算法的煤矿巡检机器人路径规划[J]. 煤矿机械,2024,45(12):180-184.

[5]沈跃,孙浩,沈亚运,等. 基于改进 A* 算法的水空两栖机器人多目标路径规划[J]. 农业工程学报,2025,41(6):62-70.

[6]马佳宇,谭功全,郑钊,等. 基于改进 A* 算法的分层平滑安全路径规划[J/OL]. 重庆工商大学学报(自然科学版). [2024-12-04]. https://link.cnki.net/urlid/50.1155.N.20241204.1104.002.

[7]朱泺谕,胡明,吴湘,等. 基于 A* 算法奖惩机制改进并优化路径选择的 AGV 路径规划[J]. 计算机集成制造系统,2025,31(8):2786-2796.

[8]杨兰英,李超,邹海锋,等. 基于改进蚁群算法与 A* 算法相融合的机器人路径规划优化[J]. 系统仿真学报,2025,37(11):2956-2965.

[9]马自勇,朱星光,马立东. 改进 A* 和 DWA 的机器人路径规划研究[J]. 现代电子技术,2024,47(20):177-186.

[10]方祝平,李理,唐荣,等. 融合改进 A* 和速度障碍法的无人机局部避障规划[J]. 电光与控制,2025,32(7):7-12.

[11]Xie L, Xue S, Zhang J, et al. A path planning approach based on multi-direction A* algorithm for ships navigating within wind farm waters[J]. Ocean Engineering, 2019, 184: 311-322.

[12]Fan B, Guo L. An improved JPS algorithm for global path planning of the seabed mining vehicle[J]. Arabian Journal for Science and Engineering, 2024, 49(3): 3963-3977.

[13]Abed M S, Lutfy O F, Al-Doori Q F. A review on path planning algorithms for mobile robots[J]. Engineering and Technology Journal, 2021, 39(5A): 804-820.

[14]Zinage V V, Ghosh S. Generalized shape expansion-based motion planning in three-dimensional obstacle-cluttered environment[J]. Journal of Guidance, Control, and Dynamics, 2020, 43(9): 1781-1791.

[15]Wang C, Yang J, Zhang B. A fault diagnosis method using improved prototypical network and weighting similarity-Manhattan distance with insufficient noisy data[J]. Measurement, 2024, 226: 114171.

[16]Toledo F J, Galiano V, Blanes J M, et al. Photovoltaic single-diode model parametrization: an application to the calculus of the Euclidean distance to an I-V curve[J]. Mathematics and Computers in Simulation, 2024, 225: 794-819.

[17]Zhang F, Zhang L, Liu Z, et al. An improved dempster-shafer evidence theory based on the chebyshev distance and its application in rock burst prewarnings[J]. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, 2024, 10(1): 04023055.

[18]Reda M, Onsy A, Haikal A Y, et al. Path planning algorithms in the autonomous driving system: a comprehensive review[J]. Robotics and Autonomous Systems, 2024, 174: 104630.

(责任编辑　朱雪莲　英文审校　周云龙)