DOI:10.19651/j.cnki.emt.2517856

融合改进 A* 算法和人工势场法的机器鱼路径规划*

王慧锬 陈 坤 何 丽 白康乐
(新疆大学机械工程学院 乌鲁木齐 830017)

摘 要: 为了解决传统路径规划算法在路径规划时效率较低、路径不平滑、动态避障效果较差等问题,本文提出了一种将 A* 算法方法与人工势场法(APF)相结合的混合路径规划方法。针对 A* 算法采用了动态加权法,根据机器鱼行进的位置及机器鱼与障碍物的距离为指标来调整启发函数的权重,同时采用搜索角度规则表,减少搜索领域以提高效率;另一方面,采用高斯滤波法对所获得的最佳路径进行曲线平滑。然后将改进后的 A* 算法生成的路径最为 APF 算法的搜索路径,在实现最短路径规划的基础上实现了动态避障。最后进行了仿真实验,将改进的 A* 算法应用在 4 种障碍物不同的地图和 6 种大小不同的地图上。实验结果表明,与原始 A* 算法相比,改进后的 A* 算法 4 种障碍物不同的地图中,搜索时间平均减少了 52.32%,搜索节点数平均减少了 56.60%,路径长度减少了 6.33%;在 6 种不同尺寸的地图下,搜索节点数平均减少了 49.60%,搜索时间平均减少了 40.89%,路径长度平均减少了 5.55%;融合算法可以在具有动态障碍物的地图下成功地进行动态避障及路径规划。

关键词: A* 算法;路径规划;机器鱼;动态加权;高斯滤波;人工势场法
中图分类号: TN820.4; TP242 **文献标识码:** A **国家标准学科分类代码:** 510.80

Path planning of robotic fish by combining improved A* algorithm and artificial potential field method

Wang Huitan Chen Kun He Li Bai Kangle
(College of Mechanical Engineering, Xinjiang University, Urumqi 830017, China)

Abstract: To solve the problems of traditional path planning algorithms, such as low efficiency in path planning, unsmooth paths, and poor dynamic obstacle avoidance, this paper proposes a path planning method that combines the A* algorithm with the APF algorithm. For the A* algorithm, a dynamic weighting method is used to adjust the weight of the heuristic function according to the position of the traveling robotic fish and the distance between the robotic fish and the obstacles as an index; then the Gaussian filtering method is used to curve-smooth the obtained optimal path. Then the path generated by the improved A* algorithm is used as the search path of the APF algorithm, and dynamic obstacle avoidance is carried out on the basis of realizing the shortest path planning. The results of simulation experiments show that the improved A* algorithm obstacle different maps, the average reduction of the search time is 52.32%, the average reduction of the number of search nodes is 56.60%, the average reduction of the path length is 6.33%; under different sizes of maps, the average reduction of the number of search nodes is 49.60%, the average reduction of the search time is 40.89%, the average reduction of the path length is 5.55%. The fusion algorithm can perform successful dynamic obstacle avoidance and path planning under maps with dynamic obstacles.

Keywords: A* algorithm; path planning; robotic fish; dynamic weighting; Gaussian filter; artificial potential field method

0 引 言

海洋占地球表面积的 71%,是人类重要的资源池。但海底环境复杂,完成各类工作难度较大,因此人类对海洋侦察探测技术需求强烈。随着仿生学和机器人学研究的不断进步,仿鱼水下推进技术已成为水下机器人领域研究的热点之一[1]。机器鱼具有高速、高效、高机动性的优势[2],加之其对水下环境微乎其微的影响,获得了极为快速的发展。

与其他移动机器人不同,机器鱼会面对未知且较陆地

收稿日期:2025-01-08
* 基金项目:国家自然科学基金(62063033)项目资助

更复杂的环境,复杂性和不确定性会给系统带来大量的干扰,这些干扰很大程度上降低了控制的效率和准确度。因此,路径规划对于保证机器鱼的路径安全性和最优性起着决定性的作用,并且需要设计合理、高效的算法让机器鱼快速、安全完成水下的工作[3]。

在机器鱼路径规划的实现中,算法是必不可缺的环节。为了获得更好的路径,许多学者研究了对算法的改进,朱红秀等[4]针对机器鱼的路径规划问题,采用了目标导向、路径点优化对 RRT 算法进行了改进,改进后的 RRT 算法在路径规划时间和路径长度上均有提升,但无法保证每次规划的路径都是最优的;Tian 等[5]提出了一种基于鲁棒优化的路径规划设计方法用于机器鱼路径规划,选择路径能耗作为目标函数,同时考虑路径平滑性和安全性作为约束条件,提高了机器鱼的效率,但未能有效处理动态障碍物的路径规划问题;Koca 等[6]提出了一种结合了动态模型和优化推进机制的机器鱼路径规划方案,减少了转向时的剧烈变化,但优化结果依赖于参数设置,参数选择不当可能影响机器鱼的性能;高英剑等[7]针对水下机器人的路径规划问题,采用了自适应评价函数、路径膨胀处理对 A* 算法进行了改进,优化后的规划时间明显减少,但路径长度略长于传统 A* 算法。

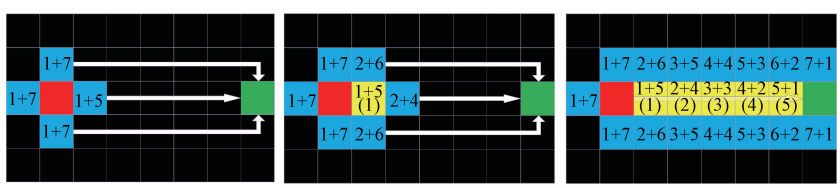
A-star 算法作为经典的路径规划算法,在计算最优路径方面表现出色,在机器人导航领域发挥关键作用[8]。为了获得更好的路径,许多学者研究了一些改进方法:丁子轩等[9]引入障碍率的概念来优化 A* 算法代价函数,并在 A* 算法生成的路径上使用贝塞尔曲线进行平滑处理,但仍存在无法达到全局最优解的情况;杨光永等[10]提出了一种结合调和 A* 算法和路径优化的方法,采用样条插值的方法进行曲线平滑处理;Wang 等[11]提出了 EBS-A* 算法,该算法引入了扩展距离,双向搜索规划并减少关键节点的数量和直角转弯的数量。然而,它需要向前和向后搜索,这需要更多的计算资源。Jing 等[12]解决了在 A* 算法中确定启发式函数的挑战,并比较了 3 种启发式函数的性能:曼哈顿距离、欧几里德距离和切比雪夫距离。然而,作者只关注启发式函数的选择,没有考虑其他可能影响 A* 算法性能的因素。Yan 等[13]将搜索邻域增大,但没有考虑路径的平滑性。Ju 等[14]提出了一种改进的 A-star 算法来解决特定条件下的路径规划问题,该算法能够找到更短的路径,但搜索时间不变。Li 等[15]设计了一种新的数据结构“值表”,取代了经典 A* 算法的 open 集合和 closed 集合,以提高检索效率,并采用堆排序算法优化节点排序效率,但路径长度不变。Lin 等[16]在改进 A* 算法中设置了冗余安全空间,以过滤掉无法通行的狭窄道路,将预判性规划策略和冗余拐点消除策略集成到算法中,以获得更好的路径,但时间是不变的,只考虑了安全性未考虑效率。Wu 等[17]提出了一种改进的双向 A-star 算法,同时考虑了距离和转弯角度,引入了偏移量 p 以优先处理更靠近目标的节点,但未考虑搜索的时间及路径平滑性。Wang 等[18]在传统的 A-star 算法中引入了八节点搜索方法,最大限度地减少了机器人转弯趋势,但该算法提高了运行时间。彭斌等[19]使用拐点识别的方法对 A* 算法进行改进,并与 DWA 算法进行结合,实现了动态避障,但在动态环境中需要较高的计算资源,降低系统的响应速度。高九州等[20]提出一种结合改进 A 算法和路径再优化的路径规划方法,能有效提升避障能力与路径规划效率,但路径长度会有小幅地增加。袁千贺等[21]采用关键点选取策略、剔除冗余节点的方法对 A* 算法进行改进,然后将改进的 A* 算法与 DWA 算法结合,提高了路径规划的效率和精度,但在障碍物之间的空间很狭窄时,算法的性能和效率可能下降。

针对目前机器鱼路径规划效率较低、路径不平滑、动态避障效果较差等问题,本文提出了一种改进的 A* 算法(I-A* 算法),旨在优化机器鱼的路径规划能力,并且减少了不必要的搜索领域,进而减少搜索节点,从而提高效率;然后对规划出的路径进行了高斯滤波曲线平滑处理,使生成的路径能够减少多余的拐点从而更加平滑。为了躲避全局最短路径中可能会出现的动态障碍物,将改进后的 A* 算法与 APF 算法进行结合,将通过改进 A* 算法得到的路径作为 APF 算法的初始路径,利用 APF 算法再次进行路径规划,以实现躲避动态障碍物的功能。最后通过仿真实验,验证了改进算法和结合算法的有效性。

1 传统 A* 算法与改进

1.1 传统 A* 算法

A* 算法是全局路径规划中常用的一种搜索算法,其结合了 Dijkstra 算法和广度优先算法的优点[22],提升了算法的搜索效率,其基本原理如图 1 所示。



(a) 开始搜索 (a) Start searching　(b) 搜索相邻节点 (b) Search adjacent nodes　(c) 搜索完成 (c) Search complete

图 1 A* 算法搜索示意图

Fig. 1 A* algorithm search principle

A* 算法到达目标点的代价评估可以用式(1)表示：

$$F(n)=G(n)+H(n) \tag{1}$$

其中，$F(n)$是从起点到目标点的总代价；$G(n)$表示从起点移动到指定点的代价；$H(n)$表示从指定点移动到目标点的代价。在 A* 算法中最关键的问题在于 $H(n)$的估计方法，这也是大多数改进的 A* 算法试图解决的方面，如果 $H(n)$的估计值等于其实际值，那么 A* 算法可以快速找到最短路径，如果 $H(n)$小于实际值，通过多次遍历可以找到最优解。

在 A* 算法中，常用的启发函数有欧几里得距离、切比雪夫距离、曼哈顿距离。欧几里得距离考虑了点之间的直线距离，而不仅仅是水平和垂直方向上的移动，适用于任何连续空间，可以准确地度量距离，而不受坐标系方向的影响，因此欧几里得距离可以看作是两点之间的最短直线路径长度。所以在本次研究中，采用欧氏距离作为 A* 算法的启发函数，从而用来计算代价，其公式为：

$$H(n)=\sqrt{(x_n-x_{goal})^2+(y_n-y_{goal})^2} \tag{2}$$

其中，x_n 与 y_n 代表 n 点的横纵坐标，x_{goal} 与 y_{goal} 代表终点的横纵坐标。

1.2　A* 算法的改进

改进 A* 算法的目的就是在保留规划出较优路径的前提下，尽量做到搜索节点减少、搜索速度提升。本文中主要采用动态加权法对评价函数 $F(n)=G(n)+H(n)$进行改进，在启发函数 $H(n)$前引入加权函数。权重系数的意义是允许为不同的节点分配不同的权重，以便更好地指导搜索过程，通过调整节点的权重，本文可以影响 A* 算法对路径的选择。对启发函数进行加权的作用主要有两个方面，一方面是影响搜索方向，加权启发函数可以使 A* 算法更倾向于朝着本文希望的方向搜索；另一方面是改变最终路径的选择，当存在多条路径估计代价相近时，节点的权重可以成为决定因素，通过对启发函数进行加权进而影响 A* 算法最终选择的路径。

但是权重系数的大小选择是一个需要重点解决的问题，由于本文需要确保算法能够在合理的时间内找到最佳路径，所以需要谨慎的选择加权值，这是一个需要重点解决的问题，过高或过低的权重可能导致算法无法找到最优解，或者搜索时间过长。本文不可能只考虑搜索速度而不考虑规划的路径，也就是不可能一直让 $W(n)$为某一个固定不变的值，此时就考虑使用动态加权的方式来缩短 A* 算法搜索的时间。其原则为，在搜索初期使用较高的权重可以减少计算资源的消耗，因为较少的节点被扩展，更关注速度和路径的快速发现；随着搜索的推进，权重逐渐降低，系统可以在计算资源相对充足的情况下，进行更加精细的路径优化，更加关注路径的安全性和精确性。

在保证上述改进的同时，再加入一个限制条件：出于安全性的考虑，如果算法搜索到了障碍物附近，那么就减少权重，增加搜索精度，保证路径的安全性；如果算法搜索的过程中离障碍物较远，那么保持它原本的权重继续搜索。

经过上述两个改进点，改进后的 A* 算法代价函数的公式为：

$$F(n)=G(n)+W(t,d)\times H(n) \tag{3}$$

其中，$W(t,d)$ 是一个同时依赖时间 t 和距离障碍物距离 d 的动态权重函数，其定义为：

$$W(t,d)=w(t)\times\phi(d) \tag{4}$$

其中，$w(t)$是时间 t 上的动态权重函数，$\phi(d)$是根据距离障碍物 d 调整的权重函数。时间权重函数定义为：

$$w(t)=\begin{cases}w_{start},t\leqslant t_{\mathrm{mid}}\\ \dfrac{w_{\mathrm{end}}-w_{\mathrm{start}}}{t_{\mathrm{end}}-t_{\mathrm{mid}}}(t-t_{\mathrm{mid}})+w_{\mathrm{start}},t_{\mathrm{mid}}<t<t_{\mathrm{end}}\\ w_{end},t>t_{\mathrm{end}}\end{cases} \tag{5}$$

其中，w_{start} 是初始权重，w_{end} 是结束权重，t_{mid} 是权重开始变化的时间，t_{end} 是权重达到最终值的时间。

距离权重函数可以根据节点 n 到最近障碍物的距离 d 来调整：

$$\phi(d)=\begin{cases}1,d\geqslant D\\ \alpha+(1-\alpha)\dfrac{d}{D},d<D\end{cases} \tag{6}$$

其中，d 为节点 n 到最近障碍物的距离；α 是一个小于 1 的常数，用来控制距离障碍物时的权重降低幅度；D 是距离障碍物的阈值，可以看到，当节点 n 到最近障碍物的距离 $d>$距离障碍物的阈值 D 时，$\phi(d)$的值为 1，此时就只考虑依赖时间 t 上的动态权重函数，如果 $d<D$ 时，则搜索精度会增加。

1.3　减少搜索领域

传统的 A* 算法是在父节点的周围进行 8 个方向的搜索[23]，在不存在障碍物的前提条件下，机器人可以向周围 8 个方向的子节点移动。当 A* 算法进行搜索时，不论在哪个位置，在其当前运动方向的反对角线上的方向是不需要被搜索到的，所以本文通过删除不需要搜索的节点来提升搜索效率。将传统的 8 个搜索节点方向改为 5 个，主要判断依据是如图 2 所示，如果以正北方向为 0°，也就是子节点 2 设置为 0°，那么子节点从 1～8 的角度分别为 315°、0°、45°、90°、135°、180°、225°、270°。

由此本文建立一个选取节点角度的规则表，如表 1 所示，包含了当前角度、保留的搜索角度以及舍弃的搜索角度，当搜索角度在如表 1 所示的给定的当前角度区间时，会保留 5 个角度的搜索，舍弃掉其余 3 个角度，这样可以避免搜索到多余角度上的节点，从而提高搜索效率。

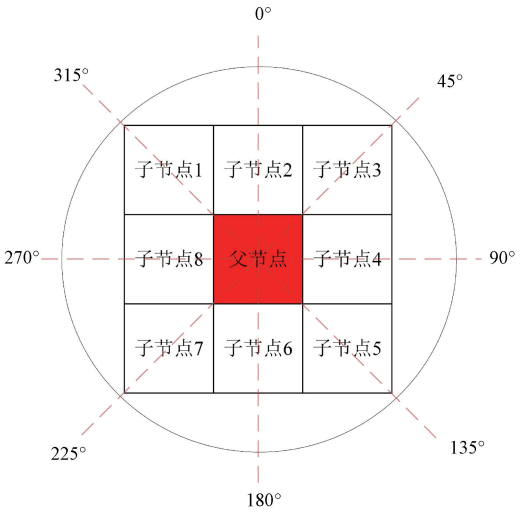


图 2　搜索角度图

Fig. 2　Search angle diagram

表 1　搜索角度规则表

Table 1　Search angle rule sheet

当前角度	保留的搜索角度	舍弃的搜索角度
0°～45°	315°、0°、45°、90°、135°	180°、225°、270°
45°～90°	0°、45°、90°、135°、180°	315°、225°、270°
90°～135°	45°、90°、135°、180°、225°	315°、0°、270°
135°～180°	90°、135°、180°、225°、270°	315°、0°、45°
180°～225°	315°、135°、180°、225°、270°	0°、45°、90°
225°～270°	315°、0°、180°、225°、270°	45°、90°、135°
270°～315°	315°、0°、45°、225°、270°	90°、135°、180°
315°～0°	315°、0°、45°、90°、270°	135°、180°、225°

2　高斯滤波平滑处理

机器鱼在水下运动的过程中,直角转弯或急剧变向会造成加速度和较大的抖动,影响其行驶安全性和精确性,同时会导致机器鱼的能耗显著增加[24],所以本文采用曲线平滑处理,对规划好的路径进行曲线平滑处理,将转弯或变向过程进行平滑过渡,减小加速度和抖动,提高机器鱼运动的平稳性。同时还可以使能耗得到有效的降低,提高能源利用效率,进一步优化路径长度,缩短行驶距离,提高效率。

高斯滤波是一种基于高斯函数的线性平滑滤波技术,通常用于信号或图像的平滑处理[25],高斯滤波实现简单、计算高效,参数调整灵活,稳定性好,在应对路径中的不规则性和突变时表现出良好的鲁棒性,而且高斯分布本身具有抗噪性强的特点,高斯滤波曲线平滑处理也能在一定程度上抑制噪声(即误差)影响。所以在本文中采用高斯滤波曲线平滑处理来平滑路径。其原理是通过将信号与一个高斯核(Gaussian Kernel)进行卷积从而实现减少噪声和细节,从而减少不连续性和突兀转角,使路径更自然和连续。高斯核的值服从高斯分布,高斯分布的一维公式为:

$$G(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{\left(-\frac{x^2}{2\sigma^2}\right)} \tag{7}$$

其中,σ 是标准差,决定了高斯函数的宽度,可以控制路径的平滑程度,较大的 σ 值会产生更加平滑的路径,而较小的 σ 值则保留更多的细节和曲率,$G(x)$代表在位置 x 处的高斯函数的值。

在一维信号处理中,高斯滤波的应用公式为:

$$y(i)=\sum_{j=-k}^{k}G(j)\times x(i+j) \tag{8}$$

$y(i)$是滤波后的信号,$x(i+j)$是原始信号,$G(j)$是以 j 为自变量的高斯函数值,k 是滤波窗口的半宽度,通常为 3σ 或 4σ。在路径规划中,可以理解为 $y(i)$是第 i 个平滑后的路径点,$x(i+j)$是平滑前的第 $i+j$ 个点的 x 坐标和 y 坐标。

高斯滤波用在路径规划上的过程为:

1)获取原始路径

获取由本文提出的 I-A* 算法生成的原始路径,这条路径通常由一系列离散的路径点组成,每个路径点包含二维坐标$(x,\ y)$。

2)确定高斯核

确定用于高斯滤波的高斯核,高斯核的大小和标准差(σ)决定了平滑效果的程度。标准差越大,平滑效果越明显。高斯核的大小通常与标准差 σ 有关。为了确保滤波效果,一般选取高斯核大小为 $6\sigma+1$ 或 $3\sigma+1$,以覆盖大部分高斯分布的权重,有助于卷积操作对称处理。而标准差 σ 决定了平滑的程度,较小的 σ 会使平滑效果较弱,但路径保留更多的细节;较大的 σ 会使平滑效果较强,路径更加光滑,细节减少。经过多次测试,本文选择效果最佳的标准差 $\sigma=1$,同时高斯核大小选取为 $6\sigma+1$。

3)卷积操作

分别对在步骤 1)获取路径的 x 坐标和 y 坐标进行处理,对路径中的每个坐标进行卷积操作,使用高斯核对 x 和 y 坐标分别通过公式进行卷积:

$$\begin{cases}x'_i=\sum_{j=-k}^{k}G(j)*x_{i+j}\\ y'_i=\sum_{j=-k}^{k}G(j)*y_{i+j}\end{cases} \tag{9}$$

其中,x'_i 和 y'_i 表示卷积后的新的 x 和 y 坐标,也就是经过曲线平滑处理处理后得到的新的路径上的 x 坐标和 y 坐标。

4)更新路径

将卷积后的平滑路径替代原始路径,作为一个新的路

径展现出来。

3　I-A* 算法结合 APF 算法

人工势场法(artificial potential field,APF)是一种非常经典的寻路方法,常用于移动机器人的局部路径规划。其主要思想是将机器人在周围环境中的运动,设计成一种抽象的人造引力场中的运动[26]。目标点对移动机器人产生“引力”,障碍物对移动机器人产生“斥力”,最后通过求合力来避开障碍物并且引导机器人从起始位置移动到目标位置。其具有数学描述简单、计算复杂度低、响应速度快、轨迹规划平滑等特点[27]。但由 APF 算法得到的规划路径不一定是最优的,并且在复杂环境中可能陷入局部最小值,导致路径规划失败。因此,APF 算法可能不符合本文路径规划应用背景的要求,所以本文将 APF 算法与 A* 算法相结合,保证了机器鱼与障碍物之间的安全距离,同时实现了对动态障碍物的躲避。

3.1　APF 算法

APF 算法需要通过建立引力场和斥力场来构成势场。引力势场主要与机器鱼和目标点间的距离有关,距离越大,机器鱼所受的势能值就越大;距离越小,机器鱼所受的势能值则越小,即目标位置对机器鱼产生引力,其大小随着机器鱼与目标位置的距离减小而减小,所以引力势场的函数为:

$$U_{att}(x)=\frac{1}{2}k_{att}(x-x_{goal})^2 \tag{10}$$

其中,$U_{att}(x)$ 是 x 处的引力势场,其大小反映了目标位置对机器鱼在该位置产生的引力势能的大小;k_{att} 是引力系数,它是一个正数,用于调节引力的强度,k_{att} 越大,在相同的位置偏差下,引力势能就越大,机器鱼受到目标位置的吸引力就越强。;x 是机器鱼当前位置,x_{goal} 是目标位置。引力的方向是指向目标位置的,这样就可以引导机器鱼向目标靠近。得到的引力向量为:

$$\boldsymbol{F}_{att}=[k_{att}(x-x_{goal}),k_{att}(y-y_{goal})]\times\boldsymbol{u}_n \tag{11}$$

其中,$\boldsymbol{u}_n$ 为从机器鱼当前位置指向目标点的单位方向向量。

斥力势场主要与障碍物有关,障碍物会对机器鱼产生斥力。决定障碍物斥力势场大小的因素是机器鱼与障碍物间的距离,当机器鱼未进入障碍物的影响范围时,其受到的势能值为零;在机器鱼进入障碍物的影响范围后,两者之间的距离越大,机器鱼受到的斥力就越小,距离越小,机器鱼受到的斥力就越大。斥力势场的势场函数为:

$$U_{rep}(x,y)=\begin{cases}\frac{1}{2}k_{rep}(\frac{1}{d-d_0}-\frac{1}{d_0})^2,d\leqslant d_0\\0,d>d_0\end{cases} \tag{12}$$

其中,$U_{rep}(x,y)$ 是在点(x,y)处的斥力势场;k_{rep} 是斥力系数,它是一个正数,用于调节斥力的强度;d_0 是一个距离阈值,表示机器鱼开始感受到斥力的距离;d 是机器鱼到障碍物的距离,采用欧氏距离来表示,其公式为:

$$d=\sqrt{(x-x_{obs})^2+(y-y_{obs})^2} \tag{13}$$

其中,x_{obs}、y_{obs} 表示障碍物的横纵坐标。

机器鱼所受的障碍物的斥力向量为:

$$\boldsymbol{F}_{rep}=\begin{cases}\frac{1}{2}k_{rep}(\frac{1}{d-d_0}-\frac{1}{d_0})^2\times\boldsymbol{u}_j,d\leqslant d_0\\0,d>d_0\end{cases} \tag{14}$$

其中,$\boldsymbol{u}_j$ 是从障碍物 j 指向机器鱼的单位方向向量。

3.2　融合算法

经过改进的 A* 算法成功地解决了水下已知环境下的全局路径规划难题,但它不能绕过动态的障碍物。APF 算法能够实现动态的避障功能,但它在全局路径信息方面存在不足[28]。因此,将改进 A* 算法与 APF 法相结合,以获得具备动态避障能力的全局最优路径。结合的算法使用改进 A* 算法用于在静态环境中进行全局路径规划,寻找最优路径。将规划好的最佳路径作为 APF 算法的初始路径,并且选取路径上的节点作为全局路径关键点。采用 APF 算法进行二次规划,用于在移动障碍物环境中动态调整路径,能够使机器鱼在运动过程中躲避动态障碍物,直至到达目标点。在结合过程中,A* 算法的代价函数会加上 APF 的影响。具体的合成代价可以表示为:

$$\tilde{F}(n)=G(n)+H(n)+\alpha\times\|\mathbf{F}_{rep}\|+\beta\times\|\mathbf{F}_{att}\| \tag{15}$$

其中,$\tilde{F}(n)$ 是结合后的代价函数。α 和 β 是权重系数,用于调整斥力和引力对代价函数的影响。结合算法流程如图 3 所示。

通过在改进 A* 算法的全局路径规划的基础上,结合 APF 算法进行局部路径规划,最终实现移动机器人动态避障,确保路径的可行性和安全性。

4　仿真实验与分析

在本章中,本文介绍了所提出算法的仿真结果,首先对比了曲线平滑处理前后的结果,接着对比了改进前后 A* 算法在大小相同、障碍物不同和障碍物相同、大小不同的两类地图下的仿真结果,然后通过复杂环境验证改进的 A* 算法结合 APF 算法进行动态避障的效果,最后通过和其他算法的对比,验证本文所提出算法的有效性。

4.1　曲线平滑处理对比结果

如图 4 所示,3 个结果图均为采用 I-A* 算法生成的路径,其中图 4(a)为未进行曲线平滑处理的路径,图 4(b)为采用文献[10]中的样条插值处理后的路径,图 4(c)为采用本文的高斯滤波曲线平滑处理后的路径。黑色形状代表障碍物,白色区域代表可行区域,左下角的蓝色圆点代表起点,右上角的红色圆点代表为终点,密集的青色 x 形状代表搜索过的节点,仿真实验得到的数据如表 2 所示。

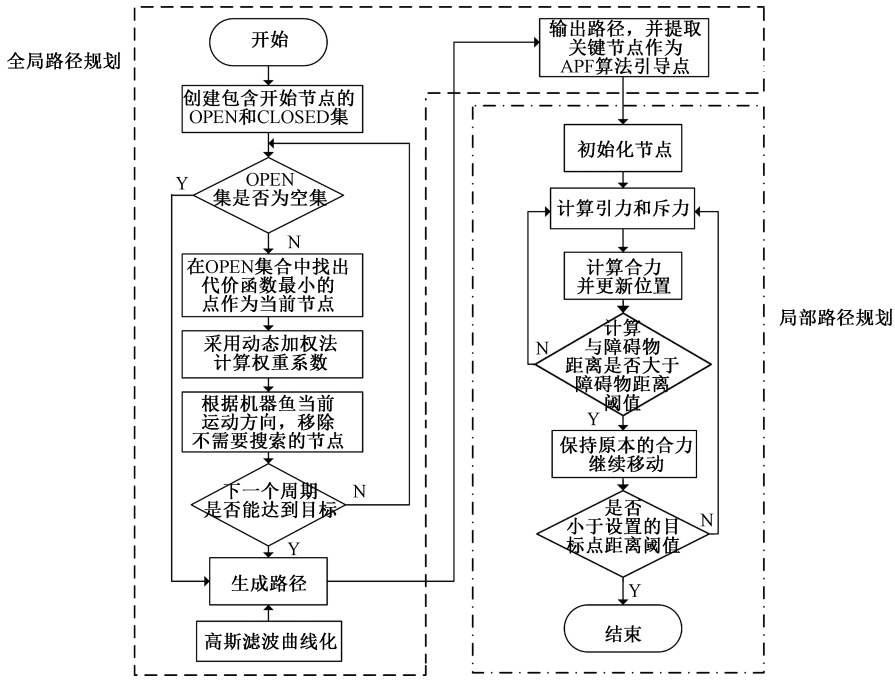
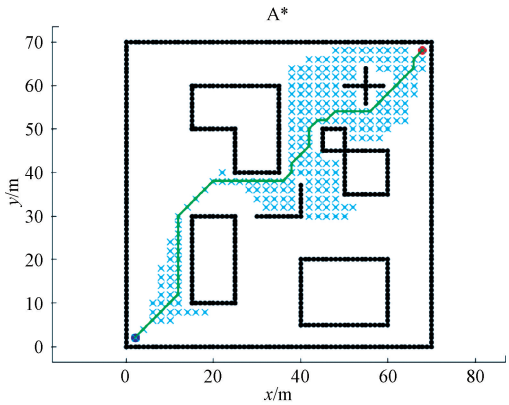
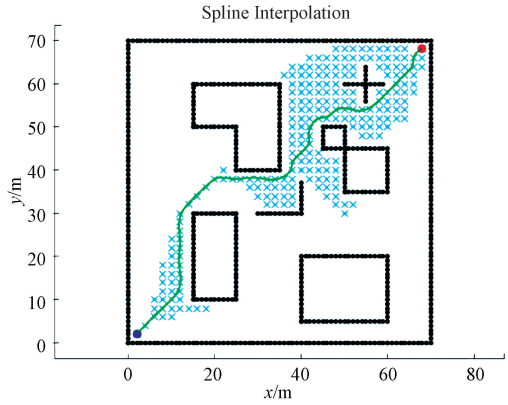


图 3　融合算法流程图

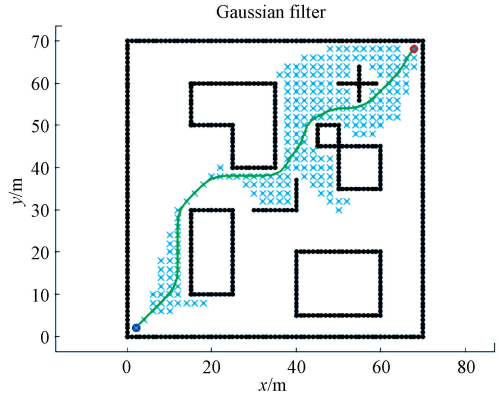
Fig. 3　The flow chart of the improved A* algorithm



(a) 未曲线平滑处理的路径
(a) Unsmoothed path diagram



(b) 样条插值处理后的路径
(b) Spline-interpolated path



(c) 高斯滤波处理后的路径
(c) Gaussian-filtered path

图 4　曲线平滑处理对比结果图

Fig. 4　Comparison of curve smoothing results

表 2　曲线化方法仿真对比结果

Table 2　Simulation results of different curvilinear methods

曲线化方法	路径长度/m	拐点个数	累计角度/(°)
无	108.569	14	627.5
文献[10]	106.217	9	490.583
高斯滤波	101.843	6	242.651

可以看到,图 4(a)中的路径相对较为曲折,能够躲避障碍物但是需要进行折线行进;图 4(b)中的路径平滑且流畅,折点更少,路径长度相对于未曲线化的路径缩短了 2.166%,累计角度减少了 21.819%,但是从结果图可以看

出，文献[10]应用的样条插值仅对折点处进行了平滑，未考虑总体路径的平滑性；图 4(c)中的路径平滑且流畅，路径长度相对于未曲线化的路径缩短了 6.195%，累计角度减少了 61.331%；相对于文献[10]采用的方法，路径缩短了 4.118%，累计角度减少了 50.538%。相比于样条插值法平滑的路径，高斯滤波得到的路径总体上更为平滑，折点更少，累计角度更少，平滑效果更好。

4.2　A* 算法与改进 A* 算法的对比仿真实验

出于比较目的，在本节中使用计算时间、搜索节点数和路径长度来评估改进 A* 算法在不同规格及不同障碍物的地图进行路径规划的效率。由于 A* 算法的最优性，其所找到的路径一定为最优路径，这也就导致搜索路径的父节点的数量与最终搜索到的最优路径长度也是相同的，但是由于本文采用了曲线平滑处理处理，会让曲线优化后的路径长度由于原始最优路径，同时搜索时间是具有变化性的，所以本文进行了 30 次仿真实验并对搜索时间取平均值。

1）具有不同障碍物的地图

由于机器鱼在水下工作，水下环境通常使得机器鱼需要避开障碍物或者是探索障碍物里面，本文选择合适的起点和终点，构建具有不同障碍物的地图，以验证机器鱼在不同障碍物下的路径规划的能力。在本节中，采用梳型障碍物、凹型障碍物、小型障碍物和大型障碍物 4 种地图进行模拟实验，将 I-A* 算法与现有的 A* 算法进行比较。通过这些障碍物的实验，可以全面评估水下机器人在不同复杂环境下的适应能力。其余所有模拟条件和设置参数均相同，地图尺寸设置为 70 m×70 m，每种地图的起点和终点设置如表 3 所示。

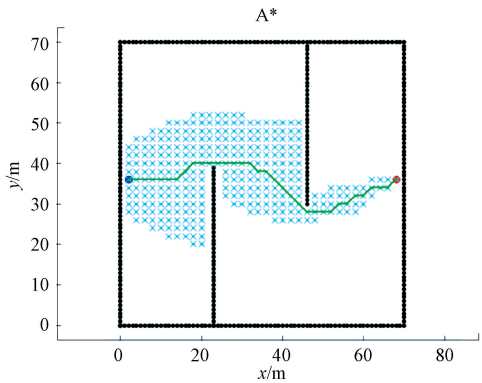
表 3　4 种地图的起点与终点坐标设置

Table 3　Start and end point coordinate settings for the four maps

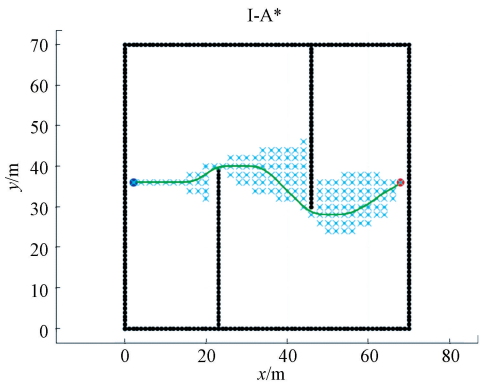
地图种类	起点	终点
具有梳型障碍物	(2，2)	(68，68)
具有凹型障碍物	(2，36)	(50，36)
具有小型障碍物	(2，2)	(68，68)
具有大型障碍物	(2，2)	(68，68)

进行仿真实验后得到的仿真结果图对比如图 5 所示，在仿真结果图中，黑色形状代表障碍物，白色区域代表可行区域，在图 5(a)～(d)中，左侧的圆点代表起点，右侧的圆点代表为终点，在图 5(e)～(h)中，左下角的圆点代表起点，右上角的圆点代表为终点，密集的青色 x 形状代表搜索过的节点。

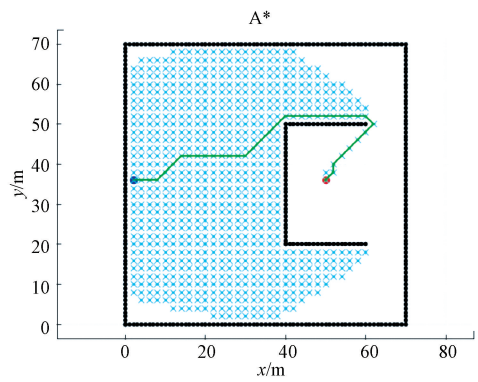
图 5(a)和(b)、(c)和(d)、(e)和(f)、(g)和(h)分别表示了改进前和改进后的梳型地图、凹型地图、小型障碍物地图和大型障碍物地图的搜索节点和搜索到的最优路径。



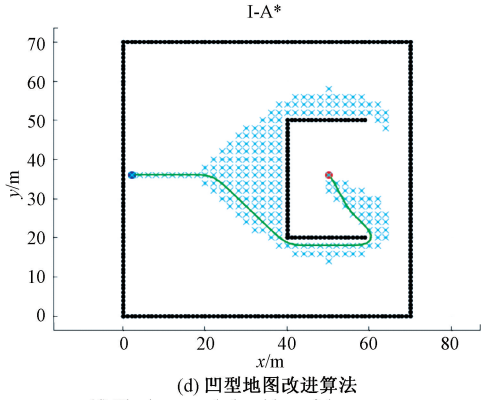
(a) 梳型地图原始算法
(a) The original algorithm of the comb map



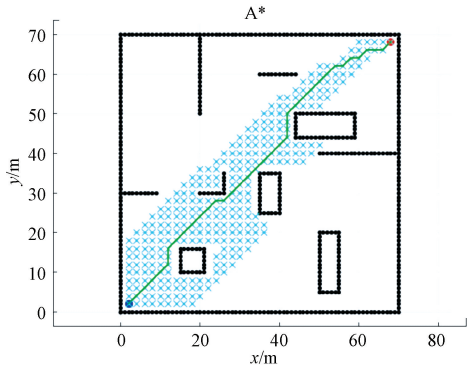
(b) 梳型地图改进算法
(b) The improved algorithm of the comb map



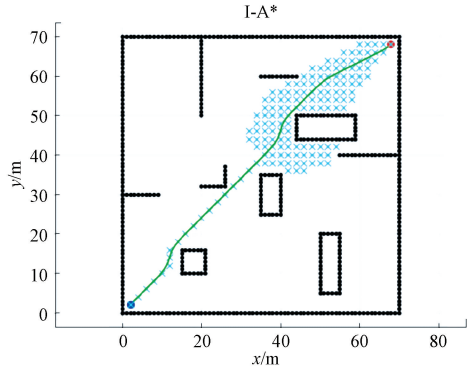
(c) 凹型地图原始算法
(c) The original algorithm of the concave map



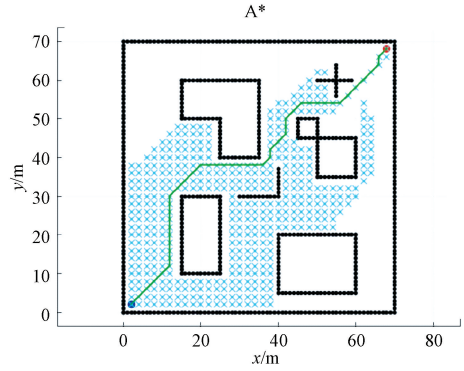
(d) 凹型地图改进算法
(d) The improved algorithm of the concave map



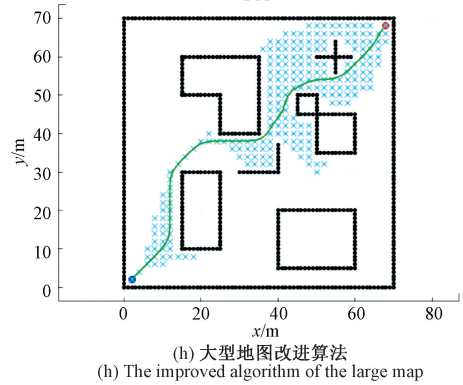
(e) 小型地图原始算法
(e) The original algorithm of the minor map



(f) 小型地图改进算法
(f) The improved algorithm of the minor map



(g) 大型地图原始算法
(g) The original algorithm of the large map



(h) 大型地图改进算法
(h) The improved algorithm of the large map

图 5 A* 算法与 I-A* 算法在不同种类的地图对比

Fig. 5 Comparison of A* and I-A* algorithms on different types of maps

图 5(b)、(d)、(f)、(h)与图 5(a)、(c)、(e)、(g)相比,I-A* 算法生成的最优路径所搜索的节点数量在路径初期是大量减少的,接近终点的时候是增加的,这符合本文提出的动态加权的规律。

图 6 为在梳型障碍物地图、凹型障碍物地图、小型障碍物地图以及大型障碍物地图下搜索节点数的对比图;图 7 为 4 种不同种类的地图下搜索时间及搜索时间标准差的对比图;图 8 为 4 种不同种类的地图下最佳搜索路径长度的对比图。由图 6 可以得出,在梳型障碍物下,节点减少了 53.40%;在凹型障碍物下,减少了 71.28%;在小型障碍物下,减少了 50.33%;在大型障碍物下,减少了 51.40%,减少搜索节点的数量较高。

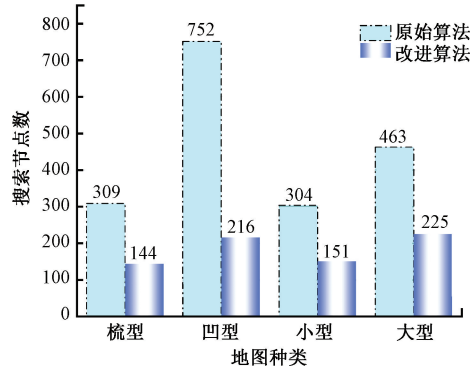


图 6 不同种类的地图下搜索节点对比

Fig. 6 Number of search nodes with different obstacle maps

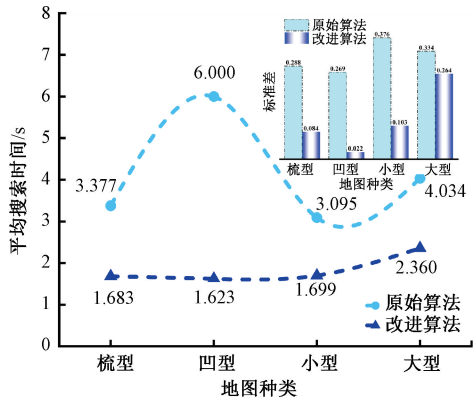


图 7 不同种类的地图下平均搜索时间对比

Fig. 7 Average search time with different obstacle maps

由图 7 可以得到,在 4 种地图中,I-A* 算法搜索时间的标准差全部小于 A* 算法,因此 I-A* 算法在搜索时间方面具有更加稳定的特点。由于 A* 与 I-A* 算法搜索到的最优路径是唯一的,而本文采取了曲线平滑处理处理,会省略掉一些折点,因此会使路径长度有一个小幅的减少。结合图 7 和图 8 可以计算出,在具有梳型障碍物的地图中,平均搜索时间减少了 49.74%,路径长度减少了 5.99%;在具凹型障碍物的地图中,平均搜索时间减少了 72.95%,路径长度减少了 6.70%;在具有小型障碍物的地图中,平均搜

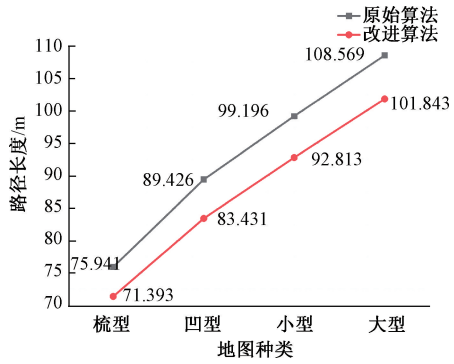


图 8　不同种类的地图下路径长度对比

Fig. 8　Path lengths with different obstacle maps

索时间减少了 45.10%，路径长度减少了 6.44%；在具有大型障碍物的地图中，平均搜索时间减少了 41.48%，路径长度减少了 6.20%。由此可知，本文提出的 I-A* 算法在应对不同种类的地图时对搜索时间、搜索节点数和路径长度均有提高，这证明了算法的有效性。

2）不同规格地图仿真实验

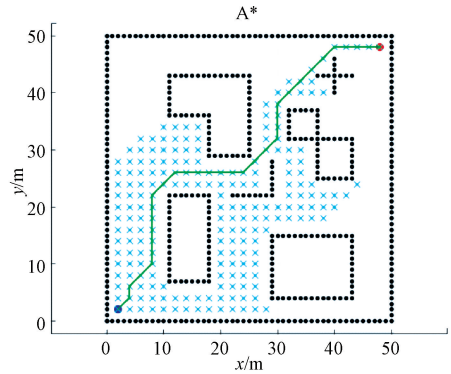
在本节中，建立复杂的障碍物地图，改变地图大小，对不同大小地图的路径规划进行对比，以验证机器鱼在不同大小的地图里也具有较佳的路径规划的能力。在这些地图里，起点设置为地图最左下角，目标点的位置随着地图的大小而变化，通常是地图中距离起点最远的点，也就是右上角。使用纵横比为 1∶1，在不同尺寸的地图中对传统 A* 算法和 I-A* 算法进行了比较，地图里的障碍物大小是按照比例进行放大的，评价指标是搜索时间（单位为 s）、搜索节点和路径长度（单位为 m）。从 50 m 大小的长度开始，以 10 m 为步长，对 A* 算法和改进的 I-A* 算法在 50 m×50 m、60 m×60 m、70 m×70 m、80 m×80 m、90 m×90 m 和 100 m×100 m 六个尺寸的地图上进行比较，其余模拟条件和参数均相同，从而确定改进的路径规划算法的优势。起始点和目标点设置在 6 张地图的相应位置，如表 4 所示。

表 4　六种大小的地图的起点与终点坐标设置

Table 4　Start and end point coordinate settings for six sizes of maps

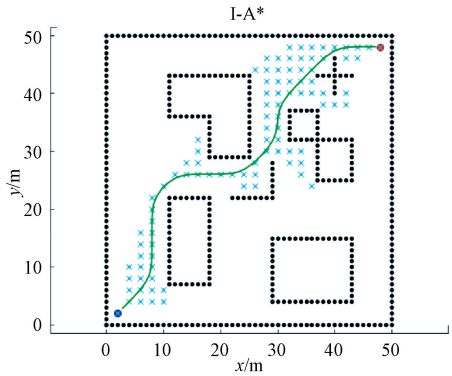
地图大小/m	起点	终点
50×50	(2, 2)	(48, 48)
60×60	(2, 2)	(58, 58)
70×70	(2, 2)	(68, 68)
80×80	(2, 2)	(78, 78)
90×90	(2, 2)	(88, 88)
100×100	(2, 2)	(98, 98)

进行仿真实验后得到的仿真结果图对比如图 9 所示。



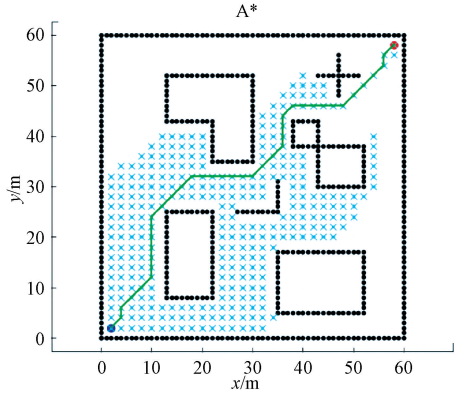
(a) 50×50原始算法

(a) 50×50 original algorithm



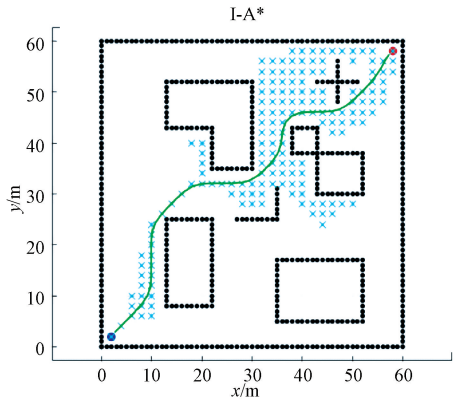
(b) 50×50改进算法

(b) 50×50 improved algorithm



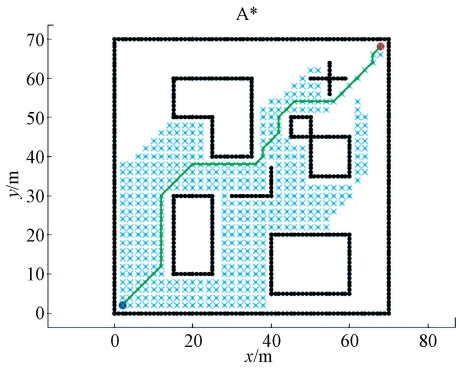
(c) 60×60原始算法

(c) 60×60 original algorithm

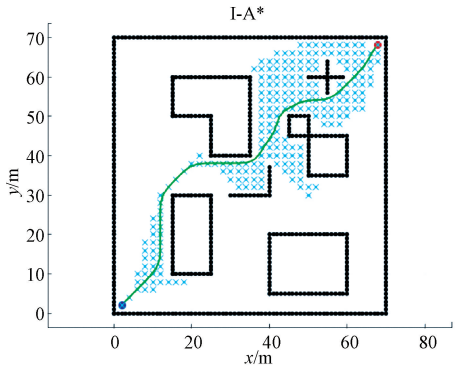


(d) 60×60改进算法

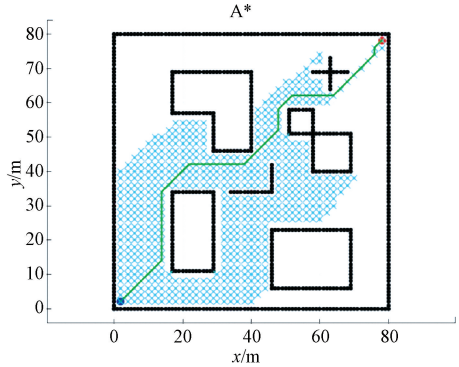
(d) 60×60 improved algorithm



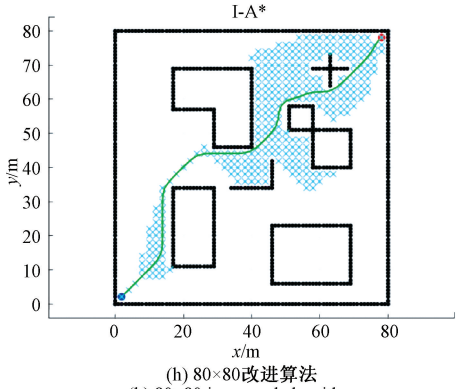
(e) 70×70原始算法
(e) 70×70 original algorithm



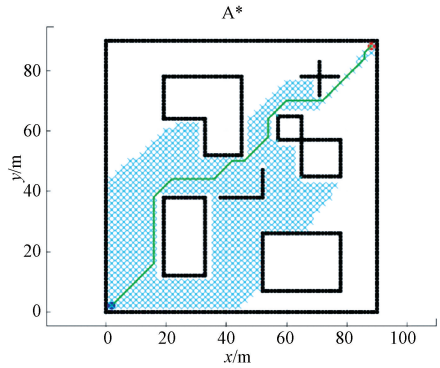
(f) 70×70改进算法
(f) 70×70 improved algorithm



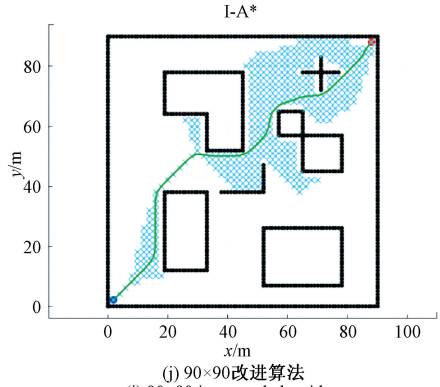
(g) 80×80原始算法
(g) 80×80 original algorithm



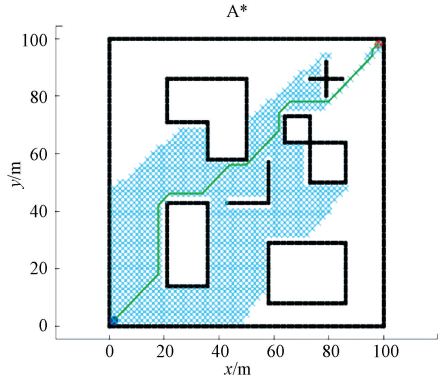
(h) 80×80改进算法
(h) 80×80 improved algorithm



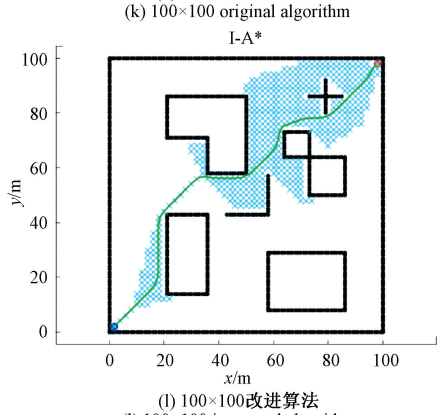
(i) 90×90原始算法
(i) 90×90 original algorithm



(j) 90×90改进算法
(j) 90×90 improved algorithm



(k) 100×100原始算法
(k) 100×100 original algorithm



(l) 100×100改进算法
(l) 100×100 improved algorithm

图 9 A* 算法与 I-A* 算法在不同大小的地图对比

Fig. 9 Comparison of A* and I-A* algorithms on maps of different sizes

图 9(a)和(b)、图 9(c)和(d)、图 9(e)和(f)、图 9(g)和图 9(h)、图 9(i)和(j)、图 9(k)和(l)分别表示了在尺寸为 50～100 m 的地图下改进前与改进后的搜索节点和搜索到的最优路径。可以看到，图 9(b)、(d)、(f)、(h)、(j)、(l)与图 9(a)、(c)、(e)、(g)、(i)、(k)相比，I-A* 算法生成的最优路径所搜索的节点数量在路径初期是大量减少的，接近终点的时候是增加的，这符合本文提出的动态加权的规律。

图 10 为分别在 50 m×50 m、60 m×60 m、70 m×70 m、80 m×80 m、90 m×90 m 和 100 m×100 m 6 个地图下的搜索节点数对比图；图 11 为 6 种不同大小的地图下搜索时间及搜索时间标准差的对比图；图 12 为 6 种不同大小的地图下最佳搜索路径长度的对比图。

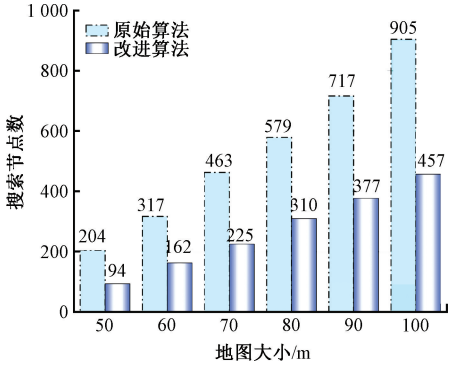


图 10　不同大小的地图下搜索节点数对比

Fig. 10　Number of search nodes with different map sizes

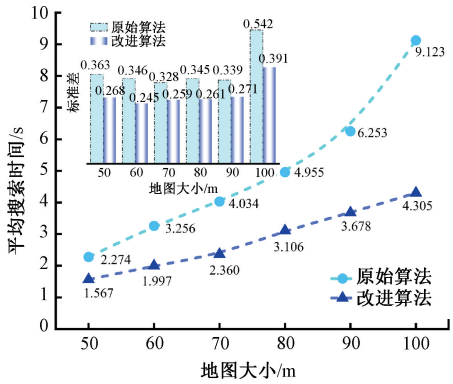


图 11　不同大小的地图下搜索时间对比

Fig. 11　Average search time with different map sizes

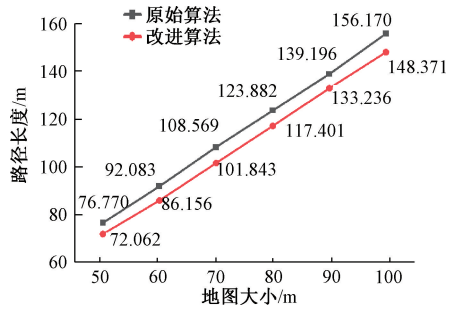


图 12　不同大小的地图下路径长度对比

Fig. 12　Path lengths with different map sizes

由图 11 可以得到，在 6 个地图中，I-A* 算法搜索时间的标准差全部小于 A* 算法，因此 I-A* 算法在搜索时间方面具有更加稳定的特点。同时图 10 和图 11 显示，A* 算法的搜索时间与搜索节点有呈指数增长的趋势，而 I-A* 算法的搜索时间与搜索节点是线性增长的趋势。由此可知，本文提出的 I-A* 算法在应对不同大小的地图时对搜索时间、搜索节点数均有较大地提升。

由图 10、11 及 12 可以得到，在 6 种不同尺寸的地图下，节点平均减少了 49.60%，搜索时间平均减少了 40.89%，同样地，由于 A* 与 I-A* 算法搜索到的最优路径是唯一的，而本文采取了曲线平滑处理处理，会省略掉一些折点，因此会使路径长度有一个小幅的减少，平均减少了 5.55%。

总体而言，I-A* 算法在所有示例中明显表现更好，具有更少的搜索节点、搜索时间和路径长度，证明了改进算法的有效性。

4.3　改进 A* 算法结合 APF 算法动态避障仿真实验

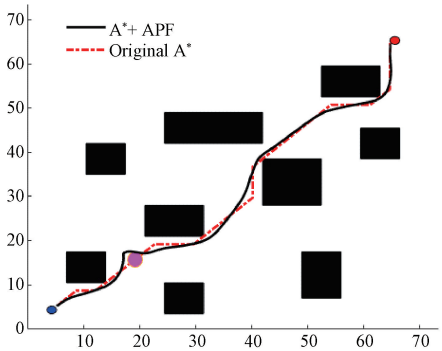
在本节中，在地图里建立多静态障碍物及动态障碍物，以验证 A* 算法结合 APF 算法实现动态避障的效果。选取 70×70 大小的地图，在仿真结果图中，黑色形状代表障碍物，白色区域代表可行区域，左下角的圆点代表起点，右上角的圆点代表为终点，地图中间的圆点代表移动的障碍物，实线为改进的 A* 算法结合 APF 算法进行路径规划得到的路径，虚线为原始的 A* 算法进行路径规划得到的路径。分别对地图中具有一个动态障碍物和两个动态障碍物的情况进行了仿真实验，实验参数的设置如表 5 所示，仿真结果如图 13 所示。

表 5　实验参数

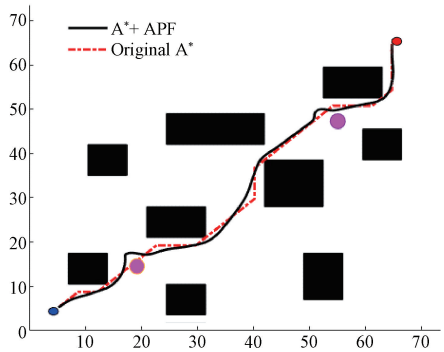
Table 5　Experimental parameters

地图大小	起点	终点	第 1 个动态障碍物的起点	运动范围	第 2 个动态障碍物的起点	运动范围
70×70	(2, 2)	(68, 68)	(20, 12)	(20,8)～(20,15)	(50,45)	(45,55)～(55,45)

图 13(a)为具有一个动态障碍物的路径规划结果图，图 13(b)图为具有两个动态障碍物后路径规划结果图。其中，第 1 个动态障碍物为纵向运动，第 2 个障碍物为横向运动。可以看到，原始的 A* 算法能够找到目标点，躲避静态障碍物，实现正确的路径规划，但只能对静态障碍物进行躲避，其生成的路径是直接穿过动态障碍物的，无法躲避动态障碍物。而改进后的 A* 算法融合 APF 算法在遇到动态障碍物时可以重新规划一条安全路径，成功实现动态



(a) 具有一个动态障碍物
(a) Results with a dynamic obstacle



(b) 具有两个动态障碍物
(b) Results with two dynamic obstacles

图 13　动态避障仿真结果

Fig. 13　Dynamic obstacle avoidance simulation results

避障。在机器鱼遭遇动态障碍物的情况下,它能够成功地进行局部避障,并返回到最优的全局规划方案,这表明融合算法在路径平滑性方面表现良好,并且与全局规划路径具有很高的重合度。

表 6 比较了本研究中提到的算法,从表 6 中可以观察到,本文提出的融合算法既能兼顾全局最优又能兼顾局部最优,确保机器鱼能够安全且稳定地绕过动态障碍,并安全抵达预定的目标位置。

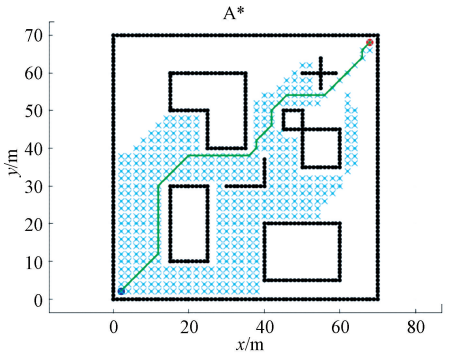
表 6　本文所述算法对比

Table 6　Comparison of the algorithms described in this paper

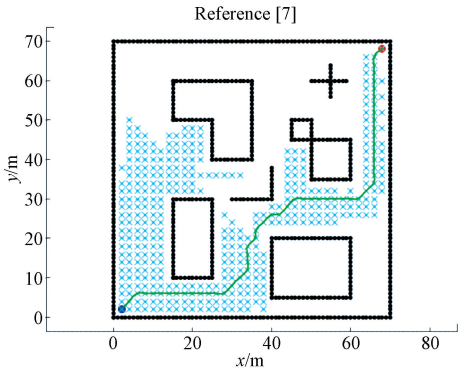
路径规划算法	全局路径规划	曲线平滑	局部路径规划	动态避障
A*	√	×	×	×
I-A*	√	√	×	×
APF	×	×	√	×
I-A* 和 APF	√	√	√	√

4.4　本文算法与其他算法仿真对比实验

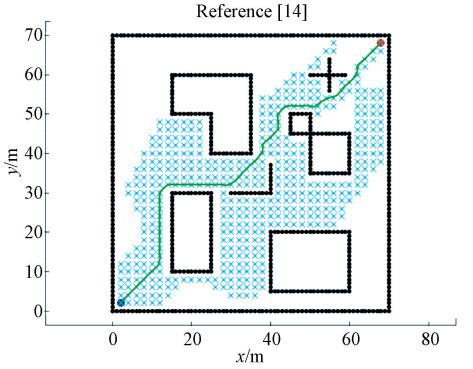
为了验证本文改进 A* 算法的优势和先进性,选取了文献[7]、文献[14]的算法在具有相同障碍物的环境下与原始 A* 算法与本文提出的 I-A* 算法进行了对比,仿真结果图如图 14 所示。



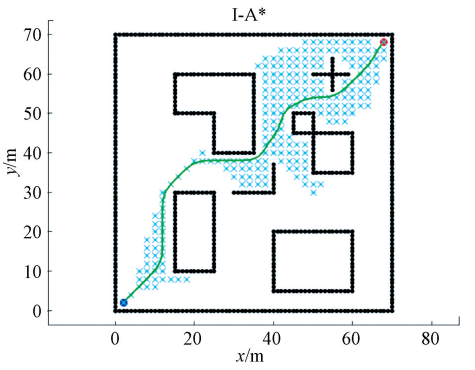
(a) 原始A*算法
(a) The original A* algorithm



(b) 文献[7]算法
(b) Algorithm of literature [7]



(c) 文献[14]算法
(c) Algorithm of literature [14]



(d) I-A*算法
(d) I-A* algorithm

图 14　不同算法仿真结果对比

Fig. 14　Comparison of simulation results of different algorithms

在图 14(a)为原始的 A* 算法的仿真结果图，图 14(b)为文献[7]中所提出方法的仿真结果图，图 14(c)为文献[14]中所提出方法的仿真结果图，图 14(d)为本文所提出的 I-A* 算法的仿真结果图。可以看到，本文提出的 I-A* 算法的搜索节点数相较于其他 3 个算法明显减少，且路径更平滑。通过仿真实验得出的仿真数据如表 7 所示。

表 7　不同算法下路径规划仿真数据对比

Table 7　Comparison of simulation data for path planning under different algorithms

算法	平均搜索时间/s	搜索节点数	路径长度/m
原始 A* 算法	4.034	463	108.569
文献[7]算法	3.082	393	110.247
文献[14]算法	4.009	437	102.591
I-A* 算法	2.360	225	101.843

由表 7 可得，本文提出的 I-A* 算法相比原始 A* 算法，路径规划的搜索时间平均减少了 41.50%，搜索节点数减少了 51.40%，路径长度减少了 6.20%；与文献[7]提出的算法相比，搜索时间平均减少了 23.43%，搜索节点数减少了 42.75%，路径长度减少了 7.62%；与文献[14]提出的算法相比，搜索时间平均减少了 41.13%，搜索节点数减少了 48.51%，路径长度减少了 0.73%。仿真实验结果表明，本文提出的 I-A* 算法在搜索时间、搜索节点数与路径长度上更有优势，效率更高。

同样地，为了验证本文提出的改进的 A* 算法融合 APF 算法的优势，选取了文献[19]提出的 A* 算法融合动态窗口法(DWA 算法)进行对比，得到的仿真结果图如图 15 所示。

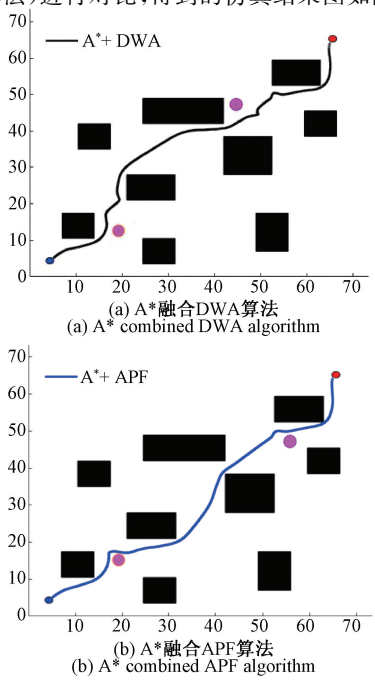


图 15　不同融合算法仿真结果图

Fig. 15　Simulation results of different fusion algorithms

在图 15 中，图 15(a)为文献[19]所用的 A* 融合 DWA 的仿真结果图，图 15(b)为本文所提出的 I-A* 算法融合 APF 算法的仿真结果图，得出的仿真数据如表 8 所示。

表 8　不同算法下动态避障仿真数据对比

Table 8　Comparison of simulation data for dynamic obstacle avoidance under different algorithms

算法	搜索时间/s	运行时间/s	路径长度/m
文献[19]算法	2.315	37.483	131.547
I-A* 融合 APF 算法	2.167	33.867	114.573

由表 8 可以得到，文献[19]所用的 A* 融合 DWA 算法在搜索路径的时间上与本文提出的 I-A* 融合 APF 算法相差无几，差值为 0.148 s，但在运行时间和路径长度上有较大差距。本文所提出的融合算法比文献[19]所用的算法在运行时间上缩短了 9.65%，也可以理解为在躲避动态障碍物时，本文提出的融合算法用时更短，比文献[19]所用的算法减少了 9.65%。在路径长度方面，由仿真结果图图 15 以及表 8 的数据均可以体现，本文提出的融合算法得到的动态避障路径长度与文献[19]所用的融合算法得到的路径长度相比减少了 12.90%。仿真结果表明，在能够保证得到安全路径的前提下，本文所提出的融合算法具有较快的搜索时间、运行时间以及较短的路径长度，这使得机器鱼在面对动态障碍物时能够有效且快速地进行动态避障并到达目标点。

5　结　　论

针对传统路径规划算法在路径规划时效率较低、路径不平滑、动态避障效果较差等问题，本文提出了一种基于改进 A* 算法和 APF 算法的融合算法。该算法首先对 A* 算法采用动态加权的方法，主要是在时间和与障碍物的距离两个方面进行了动态加权，使搜索最佳路径的时间和节点数实现初步的减少。在此基础上，将 A* 算法的搜索领域减少，解决了由于 A* 算法会搜索不必要的搜索领域导致规划速度慢的问题。接着对改进后的最佳路径进行了高斯滤波曲线平滑处理处理，使生成的最佳路径更加平滑、流畅。然后将改进的 A* 算法与 APF 算法融合，将 I-A* 算法得到的路径作为 APF 算法的初始路径，用 APF 算法进行二次规划，实现动态避障。仿真结果表明本文提出的算法在不同地图下的搜索时间、搜索节点数以及路径长度方面均具有较好的优势，同时，不论是横向动态障碍物还是纵向动态障碍物，融合算法都能够成功的进行动态避障，并能在避障后继续按照 I-A* 算法的路径进行运动。但本研究对路径规划的探索仅限于二维空间，未来的研究将扩展到三维空间，以增强机器鱼在复杂多变的海洋环境中的导航和规避能力。

参考文献

[1] YU J, WANG M, DONG H, et al. Motion control and motion coordination of bionic robotic fish: A review[J]. Journal of Bionic Engineering, 2018, 15: 579-598.

[2] LI C, SI Q, ZHAO J, et al. A robot path planning method using improved Harris Hawks optimization algorithm[J]. Measurement and Control, 2024, 57(4): 469-482.

[3] 胡鑫一,蔡振宇,左可文,等. 机器鱼的运动控制以及路径规划算法研究进展[J]. 船舶工程,2022,44(S1):455-458.
HU X Y, CAI ZH Y, ZUO K W, et al. Research progress of motion and path planning algorithm for machine fish movement[J]. Ship Engineering, 2022, 44(S1):455-458.

[4] 朱红秀,郑权,杜闯,等. 改进 RRT 算法用于电磁驱动机器鱼路径规划[J]. 火力与指挥控制,2020,45(10):100-105.
ZHU H X,ZHENG Q,DU CH, et al. Application of improved RRT algorithm in path planning of electromagnetically driven robotic fish [J]. Fire Control & Command Control, 2020, 45(10): 100-105.

[5] TIAN Q, WANG T, WANG Y, et al. Robust optimization design for path planning of bionic robotic fish in the presence of ocean currents[J]. Journal of Marine Science and Engineering, 2022, 10(8): 1109.

[6] KOCA G O, KORKMAZ D, BAL C, et al. Implementations of the route planning scenarios for the autonomous robotic fish with the optimized propulsion mechanism[J]. Measurement, 2016, 93: 232-242.

[7] 高英剑,郭平. 基于改进 A* 算法的遥控水下机器人路径规划[J]. 工业仪表与自动化装置, 2023(3):75-79,121.
GAO Y J, GUO P. Path planning of remotely operated vehicle based on improved A* algorithm[J]. Industrial Instrumentation & Automation, 2023(3): 75-79,121.

[8] WANG P, LIU Y, YAO W, et al. Improved A-star algorithm based on multivariate fusion heuristic function for autonomous driving path planning[J]. Journal of Automobile Engineering, 2023, 237(7): 1527-1542.

[9] 丁子轩. 基于 A* 算法与贝塞尔曲线的 AUV 路径规划算法研究[J]. 海岸工程,2024,43(3):225-234.
DING Z X. Research on path planning algorithm for autonomous underwater vehicles(AUV) based on A* algorithm and Bézier curves[J]. Coastal Engineering, 2024, 43(3): 225-234.

[10] 杨光永,戈一航,晏婷,等. 基于调和 A* 算法在移动机器人中的研究[J]. 现代电子技术,2022,45(4):171-176.
YANG G Y, GE Y H, YAN T, et al. Research on mobile robot based on harmonic A* algorithm[J]. Modern Electronics Technique, 2022, 45(4):171-176.

[11] WANG H, LOU S, JING J, et al. The EBS-A* algorithm: An improved A* algorithm for path planning[J]. PloS One, 2022, 17(2): e0263841.

[12] JING X R, YANG X J. Application and improvement of heuristic function in A* algorithm[C]. 2018 37th Chinese Control Conference (CCC). IEEE, 2018: 2191-2194.

[13] YAN W, FANG F J, KAIFENG D, et al. Improved global path planning a algorithm based on node optimization[J]. Computer Measurement & Control, 2023, 31(6): 143-148.

[14] JU C, LUO Q H, YAN X Z. Path planning using an improved a-star algorithm[C]. 2020 11th International Conference on Prognostics and System Health Management(PHM-2020 Jinan). IEEE, 2020: 23-26.

[15] LI J, YU CH P, ZHANG Z, et al. Improved a-star path planning algorithm in obstacle avoidance for the fixed-wing aircraft [J]. Electronics, 2023, 12(24): 5047.

[16] LIN ZH, WU K, SHEN R L, et al. An efficient and accurate a-star algorithm for autonomous vehicle path planning [C]. IEEE Transactions on Vehicular Technology, 2023, 73(6): 9003-9008.

[17] WU H F, JIAO CH Y, MEI X J, et al. Fast bidirectional a-star algorithm based on radar image data fusion for polar ship route planning[C]. 2024 9th International Conference on Computer and Communication Systems (ICCCS). IEEE, 2024: 95-101.

[18] WANG J Q, NING ZH, CHAO M. ART path planning method based on the 3D steering angle weighted A-star algorithm[J]. International Journal of Vehicle Information and Communication Systems, 2024, 9(3): 256-275.

[19] 彭斌,王力,杨思霖. 基于改进 A* 算法和动态窗口算法的自动导引小车轨迹规划[J]. 计算机应用,2022,42(S1):347-352.
PENG B, WANG L, YANG S L, et al. Trajectory planning of automated guided vehicle based on improved

A* algorithm and dynamic window approach[J]. Journal of Computer Applications, 2022, 42(S1):347-352.

[20] 高九州,刘育航. 基于改进 A* 算法和再优化的避障路径规划[J]. 电子测量技术,2024,47(21):21-27.
GAO J ZH, LIU Y H. Obstacle avoidance path planning based on improved A* algorithm and re-optimization[J]. Electronic Measurement Technology, 2024, 47(21): 21-27.

[21] 袁千贺,魏国亮,田昕,等. 改进 A* 和 DWA 融合的移动机器人导航算法[J]. 小型微型计算机系统,2023,44(2):334-339.
YUAN Q H, WEI G L, TIAN X, et al. Mobile robot navigation method based on fusion of improved A* algorithm and dynamic window approach[J]. Journal of Chinese Computer Systems, 2023, 44(2):334-339.

[22] 丰雪艳,李振璧. 融合改进的 A* 算法和动态窗口法的机器人路径规划[J]. 兰州文理学院学报(自然科学版),2024,38(1):50-54,65.
FENG X Y, LI ZH B. Robot path planning by combining improved A* algorithm and dynamic window method[J]. Journal of Lanzhou University of Arts and Science(Natural Sciences), 2024, 38(1):50-54,65.

[23] 吴飞龙,郭世永. 融合改进 A* 和动态窗口法的 AGV 动态路径规划[J]. 科学技术与工程,2020,20(30):12452-12459.
WU F L,GUO SH Y. Dynamic path planning of AGV based on fusion of improved A* and dynamic window approach[J]. Science Technology and Engineering, 2020, 20(30):12452-12459.

[24] LIAO T J, CHEN F, WU Y T, et al. Research on path planning with the integration of adaptive A-star algorithm and improved dynamic window approach[J]. Electronics, 2024, 13(2): 455.

[25] YAO M, DENG H G, FENG X Y, et al. Global path planning for differential drive mobile robots based on improved BSGA* algorithm[J]. Applied Sciences, 2023, 13(20): 11290.

[26] 许万,程兆,朱力,等. 一种基于改进人工势场法的局部路径规划算法[J]. 电子测量技术,2022,45(19):83-88.
XU W, CH ZH, ZHU L, et al. A local path planning algorithm based on improved artificial potential field method[J]. Electronic Measurement Technology, 2022, 45(19):83-88.

[27] 音凌一,向凤红. 融合改进灰狼优化算法和人工势场法的路径规划[J]. 电子测量技术,2022,45(3):43-53.
YIN L Y, XIANG F H. Path planning combined with improved grey wolf optimization algorithm and artificial potential field method[J]. Electronic Measurement Technology, 2022, 45(3):43-53.

[28] 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.
MA Z Y, ZHU X G, MA L D. Research on robot path planning based on improved A* and DWA[J]. Modern Electronics Technique, 2024, 47(20): 177-186.

作者简介

王慧锬,硕士研究生,主要研究方向为仿生机器人。
E-mail:18647628472@163.com

陈坤(通信作者),博士,教授,主要研究方向为流体机械工程仿生、仿生机器人、机械优化设计。
E-mail:chenkun@xju.edu.cn