

doi: 10.7690/bgzd.2021.07.019

室内移动机器人运动规划与导航算法优化

牟春鹏, 汪正涛, 陶卫军

(南京理工大学机械学院, 南京 210096)

摘要: 为提高动态规划算法的效率与安全性, 设计一种适用于动态复杂环境运动规划的优化算法。从状态搜索空间、路径重规划和路径平滑性 3 个方面对跳点搜索 (jump point search, JPS) 路径规划算法进行优化, 通过与动态窗口法 (dynamic window approach, DWA) 轨迹规划算法融合对其评价函数进行优化, 实现基于动态窗口方式的实时避障, 并对 A*、JPS 算法和改进 JPS 算法的仿真测试进行对比实验。实验结果证明了该优化算法的可行性与安全性。

关键词: 移动机器人; 环境建模; 路径规划; 轨迹规划; 算法优化

中图分类号: TP242 **文献标志码:** B

Algorithm Optimization for Indoor Mobile Robot Motion Planning and Navigation

Mu Chunpeng, Wang Zhengtao, Tao Weijun

(School of Mechanical Engineering, Nanjing University of Science & Technology, Nanjing 210096, China)

Abstract: In order to improve the efficiency and safety of dynamic programming algorithm, an optimization algorithm suitable for dynamic complex environment motion planning was designed. The jump point search (JPS) is optimized in the following 3 aspects: the state search space, the path re-planning and the path smoothness. Integrated with the dynamic window approach (DWA) trajectory planning algorithm of which the evaluation function is optimized, the real-time obstacle avoidance is realized based on the dynamic window mode. The simulation test and the comparison experiment of the A*, the JPS algorithm and the improved JPS algorithm are performed. The experimental results show the effectiveness and the safety of the proposed trajectory planning optimization algorithm.

Keywords: mobile robots; environmental modeling; path planning; trajectory planning; algorithm optimization

0 引言

运动规划是移动机器人自主移动必须的关键技术, 包括路径规划和轨迹规划 2 个过程^[1]。其中, 路径规划的目的是规划出一条从当前位置自主移动到目标的最优无碰路径^[2], 而轨迹规划则要求在路径规划的基础上加入时间序列信息, 以满足机器人在速度空间上的平稳性和可控性等要求^[3]。

目前, 静态路径规划开始大量应用于机器人自主定位与导航的实际应用中。常用的路径规划算法有 A*算法、RRT 算法与 Dijkstra 算法等^[4-5]。随着移动机器人的工作环境日益复杂化和非结构化, 需要预先环境建图的静态路径规划已无法满足需求。动态路径规划方法则能通过多种传感器对信息完全未知或部分未知的环境进行识别、建模与自主导航, 开发适用于动态规划的算法显得日益重要。已有的动态路径规划算法包括 LPA*算法、D*算法与跳点搜索 (JPS) 算法等^[6-8], 在效率与安全性上还需进一

步提高。

从提高动态规划算法的效率与安全性出发, 笔者提出一种基于反向搜索和路径拼接的改进 JPS 算法, 来提高 JPS 算法在动态环境和未知环境下的效率与安全性; 同时, 通过对算法中的评价函数进行优化, 并融合动态窗口法 (DWA) 轨迹规划算法, 解决其在动态规划中容易限于局部最优解的情况, 并通过仿真对比与室内移动机器人样机实验进行了有效验证。

1 机器人框架

笔者开发的室内移动机器人采用两轮差速驱动底盘, 采用同步带传动方式来驱动行走轮。在机器人前部安装有 Ust-10lx 型激光雷达, 同时在机器人圆周均布 8 个超声波传感器, 其结构如图 1 所示。机器人采用模块化设计, 按功能模块划分为运动控制、定位导航和人机交互模块, 控制系统框架如图 2 所示。

收稿日期: 2021-03-21; 修回日期: 2021-04-20

作者简介: 牟春鹏 (1996—), 男, 山东人, 硕士, 从事智能机械与机器人应用技术研究。E-mail: 1258746393@qq.com。

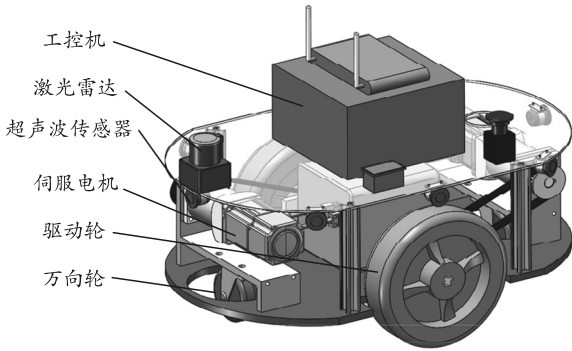


图 1 室内移动机器人结构

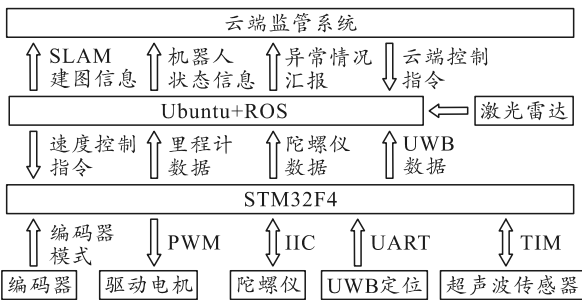


图 2 室内移动机器人控制系统框架

2 环境地图建模及现有算法分析

2.1 环境地图建模

路径规划是一个包括环境建模和路径搜索策略 2 个子问题的带约束优化问题^[9]，其中环境建模的主要方法有可视图法、自由空间法和栅格法等。

在动态规划中，移动机器人在自主导航过程中同时维护静态地图层、局部动态障碍物层和安全代价层 3 层栅格地图。静态地图层主要是由电子地图或者 SLAM 建立的静态地图；局部障碍物层主要是将传感器测量的障碍物信息实时增量式补充到静态地图层的未知区域，实现对未知区域的探索和建图；安全代价层是根据机器人实际尺寸对静态地图层和局部障碍物层进行一定程度的膨胀，以保证机器人与障碍物保持安全距离。其环境地图建模如图 3 所示。

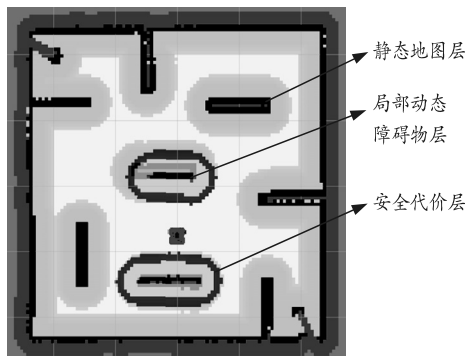


图 3 环境地图建模

2.2 JPS 算法

JPS 算法是基于 A* 算法的一种改进启发式正向搜索算法，适用于地形移动成本一致的静态栅格地图。JPS 算法的基本思想是通过修剪原则减少搜索过程中的中间点，极大地缩短路径搜索时间^[10]，搜索速度较 A* 算法提高了 1 个数量级以上，基本原理如下：

JPS 算法属于一种启发式搜索算法，通过估计函数式(1)来确定节点的拓展方向。

$$f(n)=g(n)+h(n)。(1)$$

式中： $f(n)$ 为从起始状态到目标状态的估计代价； $g(n)$ 为耗散函数，表示目标节点 n_{goal} 到节点 n 的实际代价； $h(n)$ 为启发函数，表示节点 n 到起点 n_{start} 的估计代价。

JPS 算法通过修剪 x 的邻居集合(neighbors(x))中无需进行评估的节点，实现减少搜索过程中间点以达到更快的搜索速度。根据父节点 $p(x)$ 到 x 的移动方向是直线运动还是对角线运动采用不同的邻居修剪原则。如果节点 x 是目标点，则不需要进行任何修剪。

直线移动：

$$L(\langle p(x), \dots, u_i \rangle | x) \leq \text{len}(p(x), u_i, x)。(2)$$

对角线移动：

$$L(\langle p(x), \dots, u_i \rangle | x) < \text{len}(p(x), u_i, x)。(3)$$

式中： L 为路径的长度函数； x 为该路径不经过 x 的节点； $u_i \in \text{neighbors}(x)$ 。

自然邻居：应用直线或对角线原则后，剩余的节点成为 x 的自然邻居。

强迫邻居：当 neighbors(x)存在障碍物时，根据修剪原则无法修剪的非自然邻居称为节点 x 的强迫邻居，跳点的判断条件见算法 1。

算法 1 jump point

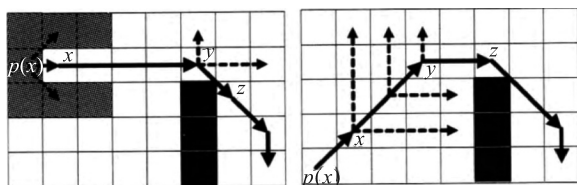
输入： x ：当前点， d ：方向， s ：起点， g ：目标点
输出： n ：跳点

- 1: $n \leftarrow \text{step}(x, \vec{d})$
- 2: **if** n is an obstacle or is outside the grid **then**
 return null
- 3: **if** $n = g$ **then return** n
- 4: **if** $\exists n' \in \text{neighbours}(n)$ s.t. n' is forced **then**
 return n
- 5: **if** \vec{d} is diagonal **then**
- 6: **for all** $i \in \{1, 2\}$ **do**

7: **if** $\text{jump}(n, \vec{d}_i, s, g)$ is not null **then**
 return n

8: **return** $\text{jump}(n, \vec{d}, s, g)$

JPS 算法通过迭代当前节点的继承跳点，并将继承跳点中 f 估计值最小的节点再进行拓展。图 4 为 JPS 算法中的跳点示例。



(a) 直线跳点

(b) 对角线跳点

图 4 JPS 算法中的跳点示例

3 改进 JPS 路径规划算法

为了提高 JPS 算法在动态环境和未知环境下的效果，提出了基于反向搜索和路径拼接的改进 JPS 算法，使其能够有效地利用原解空间进行快速路径规划，并对规划的路径进行贝塞尔曲线平滑。

3.1 改进的状态搜索空间

如图 5 所示，状态搜索空间根据起始规划点为起点，目标点分为正向搜索和反向搜索 2 种^[11]。当环境地图部分未知或动态改变时，采用正向搜索过程的 JPS 算法是反复地进行重新规划，搜索效率十分低。采用反向搜索过程仍能有效利用之前规划产生的解空间，仅需要做少量的解空间修改就能得到新的规划解。

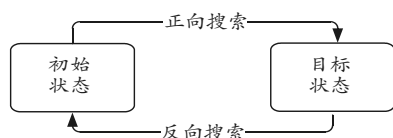


图 5 状态搜索空间

改进 JPS 算法的核心是假设未知区域都是自由空间。为了适应未知环境和动态环境，采用由目标状态到初始状态的反向搜索，增量式地实现全局路径规划。

改进 JPS 算法的路径搜索过程见算法 2。

算法 2 Improve JPS Compute Path

输入: s : 起点, g : 目标点

输出: path : 路径链表

1: **initialize** 将目标点 g 放入 open 列表, 计算 $f(g)$
 2: **repeat**
 3: **if** open 表为空 **then return** no path

4: $x \leftarrow \text{open}$ 表中 f 最小的节点

5: x 从 open 表删除, 插入 close 表

6: **if** $x=s$ **then return** fmd path

7: $\text{Successors}(x) \leftarrow \text{FIND JUMP}(x)$

8: $m \leftarrow \text{Successors}(x)$ 中最小节点

9: **if** $m=s$ **then**

10: 从起点开始, 寻找其父节点, 并插入

Path 链表

11: **return path**

12: **if** $m \in \text{close}$ **then** 调过 m 节点

13: **if** $m \notin \text{close}$ 且 $m \notin \text{open}$ **then**

14: 将 m 插入 open 列表, 设置父节点为 x

15: **if** $m \in \text{open}$ 且 $f(m) < f(x)$ **then**

16: 更新 open 表中 m 的 f, g, h 值, 设置父节点为 x

17: **until**

3.2 改进后的重规划过程

JPS 算法规划的路径由线段组成, 采用节点 x 及其父节点 $p(x)$ 坐标值 $(x, p(x))$ 表示一条线段。通过单链表存储整条路径, 机器人在移动过程中会从解空间删除已经过节点。

当环境地图发生变化且原路径陷入障碍物中, 将当前位置的父节点设置为路径链表首, 从当前位置开始检查线段的可通行性, 如果线段经过障碍物, 则将线段中的节点设置为起点, 父节点设置为终点, 对本条线段进行改进 JPS 路径规划, 利用链表特点进行路径拼接添加到原解空间。动态路径规划过程见算法 3。

算法 3 dynamic, JPS

输入: s : 起点, g : 目标点

1: **Initialize Map**

2: **repeat**

3: Improve JPS Compute Path(s, g)

4: **while** Map no change **do**

5: **if** robot pass n **then**

6: remove x from path 链表

7: $\text{top} \leftarrow p(x)$

8: 设置机器人当前位置 cur 的父节点

为 top

9: $n \leftarrow \text{cur}$

10: **while** $n=g$ **do**

11: **if** x to $p(x)$ exist obstacle **then**

12: Improve JPS Compute Path($x, p(x)$)

13: $x \leftarrow p(x)$

14: **until** top=null

3.3 路径平滑处理

改进 JPS 算法获取的路径是由多条线段组合而成，没有考虑到路径平滑性。在保证路径可行性的基础上，对路径进行 2 阶贝塞尔曲线平滑。

$$P(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, t \in [0,1]。 \quad (4)$$

如图 6 所示，2 阶贝塞尔曲线使用 3 个控制点就可以确定平面内唯一一条 2 阶贝塞尔曲线，且曲线具有仿射变换不变的特性，任意旋转和平移都不会影响其形状。根据改进 JPS 算法提供的路径端点作为控制点，对全局路径规划进行平滑性处理。

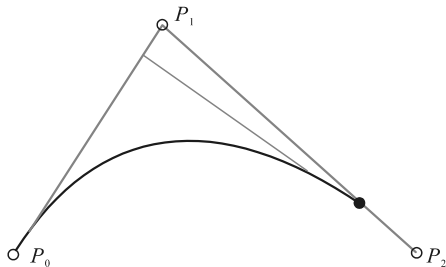


图 6 2 阶贝塞尔曲线平滑

4 基于 DWA 的动态导航算法优化

动态窗口法是根据机器人实际运动模型模拟其在一定时间下的运动轨迹^[12]，然后对运动轨迹建立评价标准以获取最优轨迹。动态窗口法根据机器人实时检测到的局部窗口环境信息进行在线实时路径规划，动态窗口随机器人滚动前进，使机器人具有良好的动态避障性能，同时可以对机器人运动轨迹进行平滑处理^[13]。由于每次规划都是基于当前窗口，缺乏有效的全局信息，容易陷入局部最优解，无法找到从起点至目标点的全局最优路径。

改进 JPS 算法进行全局路径规划将全局目标分解成局部目标，使用 DWA 轨迹规划算法逐个达到局部目标。通过修改 DWA 算法的轨迹评价函数，对轨迹规划与全局路径规划进行重合度评分，如果重合度太低则认为当前滑动窗口内出现动态障碍物或环境变化，触发改进的 JPS 算法进行动态全局规划。这样的改进可以有效地解决 DWA 轨迹规划容易陷入局部最优解的问题。

4.1 运动模型

动态窗口法需要机器人运动模型以模拟机器人的轨迹。假设机器人采用差分驱动底盘，不能实现纵向运动，一组 (v_i, w_i) 就代表了一段运动轨迹，其运

动学模型如图 7 所示。

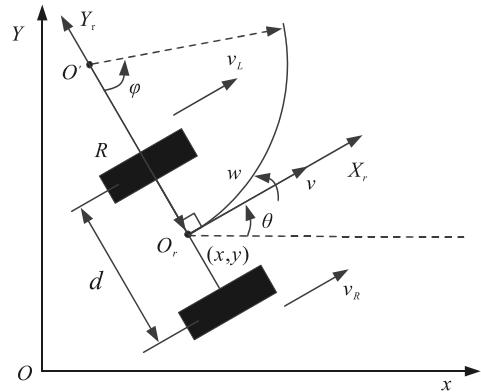


图 7 机器人运动模型

$$\begin{aligned} \theta(t) &= \int \omega(t) dt \\ x(t) &= \int v(t) \cos(\theta(t)) dt \\ y(t) &= \int v(t) \sin(\theta(t)) dt。 \end{aligned} \quad (5)$$

在速度空间 (v_i, w_i) 中存在着无数组速度。根据动力学模型可以将采样速度控制在一定范围内。

4.2 优化评价函数融合

$$G(v, \omega) = \sigma(\alpha \cdot \text{coincide}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega))。 \quad (6)$$

式中： σ 为平滑函数； α, β, γ 为各项的加权系数，根据机器人任务需求对加权系数进行调整； $\text{coincide}(v, \omega)$ 是评价机器人在当前速度空间 (v, ω) 下，模拟轨迹的终点与路径规划轨迹重合度。 $\text{dist}(v, \omega)$ 代表机器人在当前轨迹上与最近的代价地图膨胀障碍物之间的距离； $\text{velocity}(v, \omega)$ 用来评价当前轨迹的速度空间 (v, ω) 的大小。

为了检测动态障碍物和提高运行平稳性，对 DWA 轨迹规划算法进行 2 点优化：

1) 通过 DWA 轨迹规划维护滑动窗口检测动态障碍物和未知区域的地图，当 $\text{coincide}(v, \omega)$ 评分过低，则认为环境地图发生变化，反馈给改进 JPS 算法进行动态路径重规划；

2) 对 $\text{velocity}(v, \omega)$ 评价函数进行优化，对角速度添加惩罚机制以增加机器人运行平稳性。

5 仿真与实验

5.1 路径规划算法性能对比

为验证和对比算法效果，对 A* 算法、JPS 算法和改进 JPS 算法分别针对于静态地图和动态地图进行算法仿真测试。仿真测试地图选用 GPPC (grid based path planning competition) 比赛中的 Pairs 地

图，所用地图如图 8 所示。



图 8 256×256 的 Pairs 地图

本次实验分别使用 A*算法、JPS 算法和改进 JPS 算法进行仿真与对比实验，设置 2 组实验，比较传统 A*算法与改进 JPS 算法在静态环境中的性能，比较 JPS 算法与改进 JPS 算法在动态环境下重规划的性能，实验硬件为 CPU 型号 Core i5，主频 3.2 GHz，内存 4 GB 的工控机，对比结果如表 1、表 2 所示。

表 1 静态环境下传统 A*算法与改进 JPS 算法比较

算法	路径长度/px	搜索时间/ms	搜索节点数
A*	1 904	0.301 198 0	4 167
改进 JPS	1 477	0.036 030 1	1 081
对比	1.289	8.359 600 0	3.854 7

表 2 动态环境下原 JPS 算法与改进 JPS 算法比较

算法	路径长度/px	搜索时间/ms	搜索节点数
原 JPS	1 433	0.041 394 0	1 307
改进 JPS	1 489	0.002 090 8	16
对比	0.962 3	19.798 100 0	81.687 5

实验结果表明，改进 JPS 算法在静态环境下较传统 A*算法在各方面性能都有明显提升。在动态环境下，采用反向搜索空间和路径拼接的改进 JPS 算法牺牲了部分路径长度部分的性能，大幅提高了搜索速度和效率。

5.2 真实环境下运动规划实验

从验证提出算法的有效性出发，采用开发室内移动机器人样机进行了自主导航实验。测试实验环境为 5 m×5 m 办公室环境，采用 5 cm 分辨率灰度栅格地图存储环境信息，共有 100×100 个栅格，环境地图存储共需要 19.5 kB 存储空间。实验过程使用的机器人是两轮差分驱动机器人，最高速度是 1.5 m/s，最大加减速度是 0.5 m/s²，最大尺寸是 45 cm×60 cm，可以实现原地转向。运动规划算法实验如图 9 所示，机器人执行速度如图 10 所示。

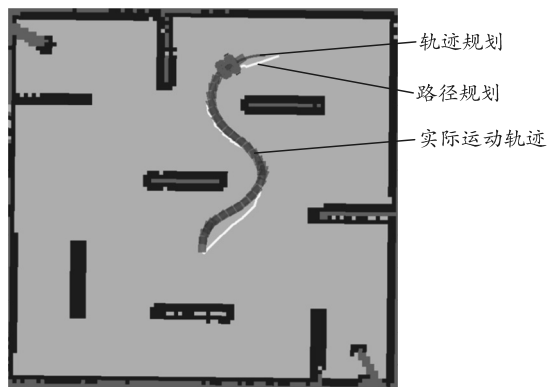


图 9 运动规划算法实验

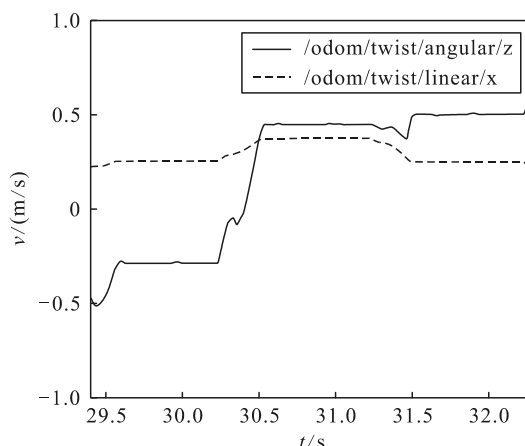


图 10 机器人执行速度

从上述实验结果可看出：改进的 JPS 算法规划在环境地图中规划的全局路径与实际障碍物始终保持了一定的安全距离，且全局规划路径经过 2 阶贝塞尔曲线平滑后具有更好的平滑性。从图 9 可以看出，机器人执行的实际运动轨迹与全局路径规划重合程度较高且整体轨迹平滑。图 10 中，机器人速度变化平滑符合机器人实际运动。

6 结论

针对移动机器人在未知环境和动态环境中进行动态运动规划中的效率与安全性问题，采用反向搜索空间和路径拼接的方式对 JPS 路径规划算法进行改进，解决了原算法在动态路径规划下效率低下和无法有效利用原解空间的问题，并对路径进行贝塞尔曲线平滑；同时与优化评价函数的 DWA 轨迹规划算法进行融合，通过维护滑动窗口的方式进行实时避障和轨迹规划，形成了一种适用于复杂环境可靠稳定的动态运动规划算法。仿真对比与实验验证了笔者提出的改进 JPS 算法的有效性，显示其在性能、效率和安全性方面具有明显的优势和提升，具有良好的应用前景。

参考文献:

[1] 王小忠, 孟正大. 机器人运动规划方法的研究[J]. 控制工程, 2004(3): 280-284.

[2] 曲道奎, 杜振军, 徐殿国, 等. 移动机器人路径规划方法研究[J]. 机器人, 2008, 30(2): 97-101.

[3] 余冬冬. 移动机器人避障与轨迹规划[D]. 杭州: 浙江大学, 2017.

[4] 陈若男, 文聪聪, 彭玲, 等. 改进 A*算法在机器人室内路径规划中的应用[J]. 计算机应用, 2019, 39(4): 1006-1011.

[5] 张文, 刘勇, 张超凡, 等. 基于方向 A*算法的温室机器人实时路径规划[J]. 农业机械学报, 2017, 48(7): 22-28.

[6] STENTZ A. The Focussed D* Algorithm for Real-Time Replanning[C]. Montreal, Quebec, Canada. 1995: 1652-1659.

[7] 张凡, 蔡涛, 刘文达, 等. 基于改进 JPS 算法的电站巡

(上接第 86 页)

4 结论

仿真结果证明: 在漂移性扰动情况下, 该模型可快速实现机器人手爪夹持力大小的稳定安全控制, 可为弹药装药装配、火炸药制备等高危产品生产制造过程中机器人的设计选型和安全控制提供理论基础。

参考文献:

[1] 牟春鹏, 孙家升, 张涵, 等. 便携式反恐排爆机器人[J]. 兵工自动化, 2020, 39(8): 86-90.

[2] 谢辉, 任国全, 李建中, 等. 军用地面机器人总体试验方法设计[J]. 兵工自动化, 2020, 39(6): 85-91.

[3] 李敬业, 高学山, 高锐, 等. 人机共存环境下巡检机器人自主移动与避障方法[J]. 兵工自动化, 2020, 39(7): 83-88.

[4] 胡寿松. 自动控制原理[M]. 7 版. 北京: 科学出版社, 2019: 144.

[5] OGATA K. Modern Control Engineering[M]. Fifth

检机器人路径规划[J]. 电子测量技术, 2020, 43(8): 10-16

[8] 刘张雷, 史忠科. 一种基于路网变化的动态路径规划策略[J]. 交通运输系统工程与信息, 2010, 10(3): 147-152.

[9] 孙俊岭. 基于电子海图智能潜水器路径规划研究[J]. 海洋工程, 2011, 29(4): 135-140.

[10] 杨凤满, 张奇志, 周亚丽. 家庭服务机器人路径规划的跳点搜索算法[J]. 北京信息科技大学学报(自然科学版), 2018, 33(3): 85-89.

[11] 徐开放. 基于 D*Lite 算法的移动机器人路径规划研究[D]. 哈尔滨: 哈尔滨工业大学, 2016.

[12] 许玉龙. 基于 ROS 的移动机器人建图导航技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2017.

[13] 程传奇, 郝向阳, 李建胜, 等. 融合改进 A*算法和动态窗口法的全局动态路径规划[J]. 西安交通大学学报, 2017, 51(11): 137-143.

Edition. BEIJING: Publishing House of Electronics Industry, 2019: 267.

[6] 刘洲峰, 孙红艳, 安菲. 基于 Matlab 的智能控制系统仿真[J]. 现代电子技术, 2007(4): 178-180.

[7] 曹宗宗. 基于状态空间模型的非均匀采样系统的辨识方法研究[D]. 大连: 大连理工大学, 2015: 35-36.

[8] 李峰, 张雪松, 岳云. 一种基于喷涂机器人的自适应运动控制系统研究与应用[J]. 机床与液压, 2017, 45(15): 53-59.

[9] MA O, ANGELES J. Architecture Singularities of Platform Manipulators[J]. IEEE International Conference on Robotics & Automation, 2002, 10(15): 1542-1547.

[10] ZHAO H L, T-T Lee. Research On Multiobjective Optimization Control for Unknown Nonlinear Systems[C]//Proceedings of IEEE 2003 International Conference on Fuzzy Systems, May 25-28, St. Louis, MO, USA: Automatic Control, 2003: 22-25.

[11] 李二超, 李战明, 李炜. 基于视觉的机器人模糊自适应阻抗控制[J]. 中南大学学报(自然科学版), 2011, 42(2): 409-413.