

doi:10.6041/j.issn.1000-1298.2021.02.002

# 移动机器人平滑 JPS 路径规划与轨迹优化方法

黄健萌 吴宇雄 林谢昭

(福州大学机械工程及自动化学院, 福州 350116)

**摘要:** 针对目前路径规划方法存在的平滑性和效率问题,在 JPS 算法基础上提出了兼顾平滑性与搜索效率的路径规划方法,并利用多项式进行了轨迹优化。首先,提出 2 个优化目标对路径序列进行优化处理;然后,对 JPS 搜索规则进行改进,得到更多有价值的路径,并对每条路径进行平滑处理,再以一定规则进行选择;最后,使用多段高阶多项式对所得路径进行轨迹优化,研究时间分配问题,从而加快迭代效率。通过仿真实验和与其他算法的对比证明了本文方法的可行性和有效性。结果表明,在不同障碍物密度环境下,本文路径规划方法得到了平滑性良好的路径,相对平滑后处理 JPS,长度减少了 0.48% ~ 1.80%,总转折角减少了 16.93% ~ 52.75%,利用余弦函数进行时间分配加快了轨迹优化的迭代效率,通过实验验证得到了良好的效果。

**关键词:** 移动机器人; 路径规划; 轨迹优化; JPS 算法; Lazy-Theta\* 算法

**中图分类号:** TP242.6      **文献标识码:** A      **文章编号:** 1000-1298(2021)02-0021-09

**OSID:**



## Smooth JPS Path Planning and Trajectory Optimization Method of Mobile Robot

HUANG Jianmeng WU Yuxiong LIN Xiezhao

(College of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350116, China)

**Abstract:** Aiming at the problems of smoothness and efficiency in current path planning methods, based on jump point search (JPS), a path planning method considering both smoothness and search efficiency was proposed, and the trajectory was optimized by polynomial. Firstly, in order to improve the smoothness of the path, two optimization objectives were proposed to optimize the path sequence, that was, each corner of the path was on the vertex of the obstacle grid, and the obstacles that contacted with the path corner were located on the side with the angle less than 180 degrees. Then, the search rules of JPS were improved to get more valuable paths, and each path was smoothed, and then the path selection was made by weighing the length and angle with certain rules. Finally, polynomial was used to optimize the path, and the time allocation of the trajectory optimization method was studied to speed up the iterative efficiency. The feasibility and effectiveness of this method were proved by simulation and comparison with other algorithms. The results showed that the method had the advantages of high efficiency of JPS algorithm, and effectively solved the problems of redundant points and frequent turning points in path planning of JPS algorithm. In different density environments, compared with the smoothed JPS algorithm, the path planning method reduced the path length by 0.48% ~ 1.80%, and the cumulative turning angle was decreased by 16.93% ~ 52.75%. In the large environment, the proposed method had similar smoothness to Lazy-Theta\*, but it had higher search efficiency. The cosine function was used to allocate time to accelerate the iterative efficiency of trajectory optimization, and good results were obtained through experiments.

**Key words:** mobile robot; path planning; trajectory optimization; JPS algorithm; Lazy-Theta\* algorithm

收稿日期: 2020-04-21 修回日期: 2020-08-07

基金项目: 福建省自然科学基金项目(2016J01227、2017J01692)

作者简介: 黄健萌(1973—),女,教授,博士,主要从事摩擦学与机器人运动规划研究,E-mail: hjmfzu@163.com

## 0 引言

随着移动机器人被广泛应用于农业等诸多领域,移动机器人导航技术逐渐成为研究热点。路径规划是移动机器人完成导航任务的核心技术之一,其目的是让机器人在充满障碍物的复杂环境中找到从起点到终点的最佳无碰撞路径<sup>[1]</sup>,通常是以规划出的路径最短作为优化目标。目前常用的路径规划方法有栅格法、人工势场法、可视图法等<sup>[2]</sup>,以及蚁群算法、粒子群算法等智能算法<sup>[3-4]</sup>。栅格法简单有效,是目前使用最广泛的一种方法。

栅格法中,Dijkstra 算法<sup>[5]</sup>搜索时只考虑已花费成本,虽然能保证找到最短路径,但会扩展大量无用节点。BFS 算法采用启发函数快速接近目标点,不考虑已花费成本,其效率高于 Dijkstra,但不能保证找到最短路径。 $A^*$  算法<sup>[6]</sup>兼顾了两者的优点,在考虑已花费成本的同时采用启发式搜索,并能保证找到最短路径。但  $A^*$  算法规划的路径通常是一组曲折路径,仅是离散空间上的最优解,与连续空间上实际最优解有较大差距。“视野线”法<sup>[7-9]</sup>常被用来对  $A^*$  算法规划的路径进行平滑后处理,但只在后处理时使用平滑方法往往无法使路径跨过障碍物,优化效果有限,平滑后的路径有时仍与最优解存在较大差距。为此,Theta\* 算法<sup>[10]</sup>提出了任意角度规划路径的新思路,该方法在  $A^*$  算法的基础上,采取实时收缩父节点,即边搜索边平滑的策略,与采用后处理平滑的路径规划方法相比,能更好地接近最优解。在此基础上许多学者进一步提出了 Lazy-Theta\*、Strict-Theta\*、Batch-Theta\* 等 Theta\* 类算法<sup>[11-13]</sup>,在效率和平滑处理等方面进行改进。由于 Theta\* 类算法是在  $A^*$  算法基础上的改进,因此随着栅格环境的增大,存在扩展节点急剧增加、搜索时间大幅延长的情况。文献[14-16]提出的跳点搜索(Jump point search, JPS)算法采用了完全不同的节点扩展方式,与  $A^*$  算法相比, JPS 算法有效减少了需要维护的节点数量,尤其是在大栅格环境中,有效提高了寻路效率,但 JPS 所规划路径仍然只是离散空间的最优路径。

采用栅格法规划的路径通常无法直接用于机器人运动,需进行轨迹优化处理。圆切线<sup>[8,17]</sup>方法虽然能生成可行轨迹,但并未考虑机器人运动的速度、惯性等物理参数。通过微分平坦<sup>[18]</sup>,多项式对各类机器人<sup>[19-20]</sup>在轨迹优化方面有着广泛应用,相比单段多项式,多段多项式可以更好地适应复杂环境。文献[21]利用多段高阶多项式的特性将加速度相关导数作为优化目标。文献[22-23]将多段多项

式约束在安全范围内,以避免轨迹发生碰撞。目前,主要依靠迭代获取多段多项式轨迹优化的最优解。

本文在 JPS 算法的基础上,提出一种平滑 JPS (SJPS)算法。首先针对现有平滑方法的缺点提出一种路径平滑方法,对 JPS 搜索规则进行修改,遍历对称路径,并利用本文平滑方法对每条所得路径进行平滑处理,再权衡长度与角度两指标,最后对所得初始路径用多段多项式进行轨迹优化,改进时间分配问题,加快迭代效率,并结合 SJPS 的特点对端点进行合理约束。

## 1 路径平滑

“视野线”方法在部分情况下存在去除必要节点,留下冗余节点的问题。如图 1 所示,路径上每个栅格中心点均为路径点,若按“视野线”方法平滑,则路径将无改变,因为起点与转折点连线检查会将原本必要的节点去除。为得到稳定且良好的平滑效果,本文提出以 2 个目标对路径点序列进行优化,即路径每个折角处均在障碍物栅格的顶点上,且与路径转角进行接触的障碍物位于转角小于  $180^\circ$  一侧。

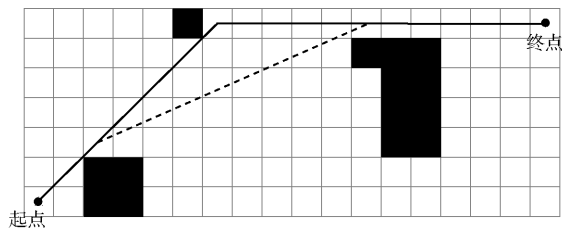


图 1 “视野线”平滑

Fig. 1 “Line of sight” smooth

### 1.1 路径单侧有障碍物的情形

本文路径平滑遵循以下步骤:

(1) 将起点作为平滑路径点的第 1 个点,记为  $q_c(x_c, y_c)$ 。

(2) 将  $q_c$  依次与后续 JPS 路径点连接,直到找到连线  $L_m$  经过障碍物,对应连接路径点  $q_m(x_m, y_m)$ ,并记录前一条没经过障碍物的连线  $L_{m-1}$ ,对应路径点  $q_{m-1}(x_{m-1}, y_{m-1})$ ,并将点  $q_{m-1}$  和  $q_m$  之间的路径离散成单个栅格移动的路径点。若连接到了终点也未经过障碍物,则终点为最后一个平滑路径点,结束搜索。

(3) 再将点  $q_c$  依次与步骤(2)中离散的路径点(包括  $q_m$ )连接,直到找到一条经过障碍物的连线作为新的  $L_m$ ,对应连接点作为新的  $q_m$ ,前一条连线记为新的  $L_{m-1}$ ,对应路径点为新的  $q_{m-1}$ 。

(4) 从  $L_m$  经过的障碍物栅格中,从可能顶点中找到一栅格顶点  $q_j(x_j, y_j)$ ,使点  $q_c$  和  $q_j$  的连线与

$L_m$  的夹角达到最大, 则  $q_j$  为新的平滑路径点。将  $q_j$  作为新的起始点  $q_c$  从步骤(2)开始新一轮搜索。

以图2的情况为例, 首先将起点作为第1个平滑路径点, 并作为搜索起点  $q_c$ , 从第2个点(4,5)开始, 将后续路径点依次与  $q_c$  连接, 每次与  $q_c$  连接的点记为  $q_m$ 。直到连接  $q_c$ (2,3) 和点(8,11)时, 其连线经过障碍物, 则将两点间的路径离散成单个栅格移动的路径点, 再将点  $q_c$  依次与离散的路径点连接。连接到点(5,8)时, 两点连线经过障碍物栅格(5,7), 记为  $L_m$ 。然后选取障碍物栅格左上角顶点(4.5,7.5)作为下一个平滑路径点, 并以其为新的起始点继续搜索。点(4.5,7.5)作为起始点搜索时, 连接到终点也未经过障碍物, 因此将终点作为最后一个平滑路径点。最终产生的平滑路径即图2中的虚线路径。

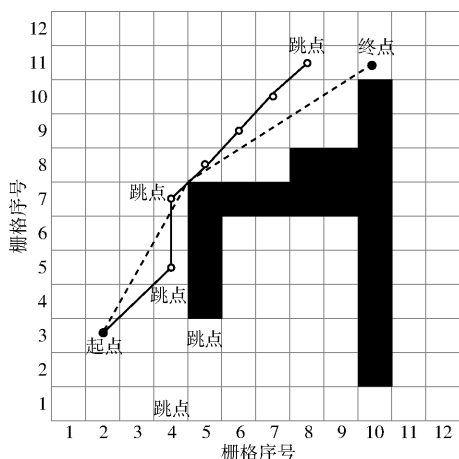


图2 平滑示意图

Fig. 2 Smoothing diagram

## 1.2 平滑路径点选取

对于步骤(4)中障碍物栅格顶点的选取, 使用一定的规则。如图3所示, 当经过障碍物的连线  $L_m$  非水平或垂直时, 挑选连线上经过的障碍物, 连线水

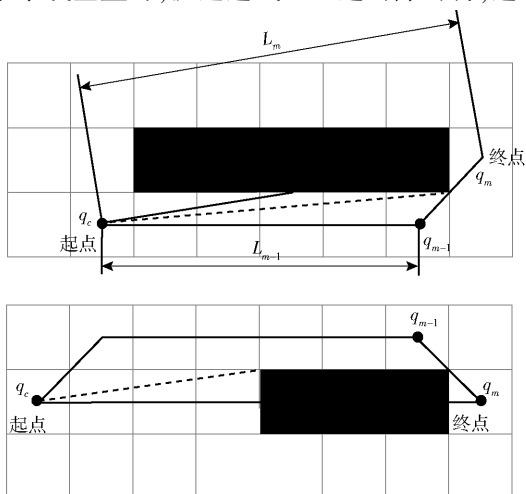


图3 栅格选取

Fig. 3 Grid selection

平或垂直时, 则只挑选第1个遇到的障碍物栅格。对于挑选的栅格  $(x_{obs}, y_{obs})$ , 每个栅格有4个顶点, 但对于每次平滑路径点搜索只有一个方向的顶点有可能被选取, 延用步骤(2)中的  $q_{m-1}$ 、 $q_c$  和步骤(3)中的  $q_m$  标记, 可能被选取的顶点应落在由  $q_m$ 、 $q_c$  和  $q_{m-1}$  3点形成的三角形区域内。若步骤(2)中没有按要求离散, 则三角形区域将存在被遗漏的障碍物栅格。如图4所示, 顶点的选取有16种可能情况。先将点  $q_m$  到点  $q_c$  和  $q_{m-1}$  方向的两向量  $V_{m,c}$  和  $V_{m,m-1}$  又乘判断两向量的顺逆时针关系, 再根据  $q_m$  和  $q_c$  连线斜率和  $V_{m,c}$  的朝向选择顶点  $(x_j, y_j)$

$$\begin{cases} \begin{cases} x_j = x_{obs} - 0.5AC \\ y_j = y_{obs} - 0.5AC \end{cases} & (B < 0) \\ \begin{cases} x_j = x_{obs} - 0.5AC \\ y_j = y_{obs} + 0.5AC \end{cases} & (B > 0) \\ \begin{cases} x_j = x_{obs} - 0.5C \\ y_j = y_{obs} + 0.5C \end{cases} & (A > 0, D = 0) \\ \begin{cases} x_j = x_{obs} + 0.5C \\ y_j = y_{obs} + 0.5C \end{cases} & (A < 0, D = 0) \\ \begin{cases} x_j = x_{obs} + 0.5D \\ y_j = y_{obs} + 0.5D \end{cases} & (A > 0, C = 0) \\ \begin{cases} x_j = x_{obs} + 0.5D \\ y_j = y_{obs} - 0.5D \end{cases} & (A < 0, C = 0) \end{cases} \quad (1)$$

其中

$$A = \frac{|V_{m,c} \times V_{m,m-1}|}{V_{m,c} \times V_{m,m-1}} \quad (2)$$

$$B = \frac{|y_m - y_c|}{|x_m - x_c|} \frac{|y_m - y_c|}{|x_m - x_c|} \quad (3)$$

$$C = \frac{|y_c - y_m|}{y_c - y_m} \quad (4)$$

$$D = \frac{|x_c - x_m|}{x_c - x_m} \quad (5)$$

式(1)分别对应图4中的16种情况, 对应情形的向量位于哪一侧则选取该侧顶点。

选出各障碍物栅格顶点  $(x_j, y_j)$  后, 将其与  $q_c$  连线, 并求其连线与  $L_m$  的夹角  $\theta$  为

$$\theta = \max \left( \arccos \frac{V_1 V_2}{|V_1| |V_2|} \right) \quad (6)$$

其中

$$V_1 = [x_m - x_c, y_m - y_c] \quad (7)$$

$$V_2 = [x_j - x_c, y_j - y_c] \quad (8)$$

从所有夹角中选出最大夹角所对应顶点  $q_j$  作为下一个平滑路径点和新的搜索起点  $q_c$ 。

## 1.3 路径双侧均有障碍物的情形

当路径双侧均有障碍物时, 可能出现如图5a所示情况。图5a中虚线路径为按上述规则优化后的平滑路径, 图5b中虚线为理想平滑路径。可知最后

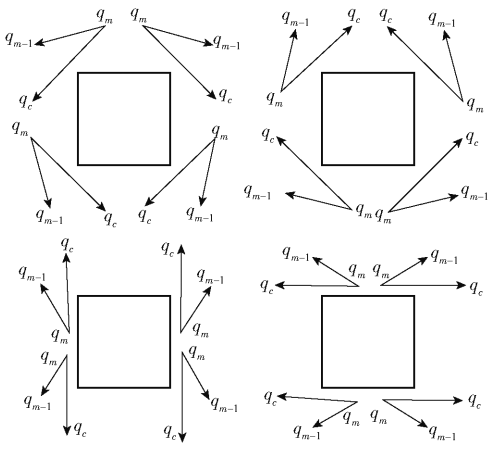


图4 平滑点选取

Fig. 4 Smooth point selection

一个转折处,转折角小于  $180^\circ$  处所在区域并无障碍物,不满足第 2 个目标。这是因为寻找第 3 个新路径点时,在从终点开始逐个连线进行搜索的过程中,受到了右侧障碍物的干扰。

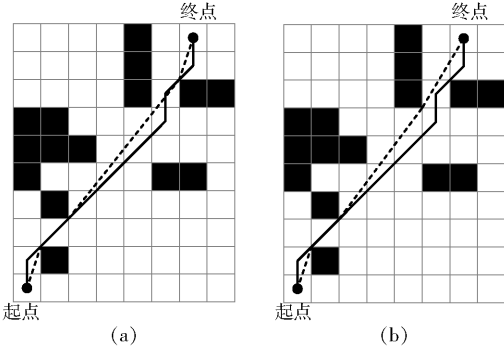


图5 路径双侧均有障碍物时的平滑情况

Fig. 5 Smooth situation when there were obstacles on both sides of path

为此,在上述平滑路径搜索步骤中增加 1 个局部平滑步骤。每 3 个连续路径点产生一折角,从第 3 个新路径点开始,每产生一个路径点  $O_3(x_3, y_3)$ , 则设其前 2 个点依次为  $O_2(x_2, y_2)$ 、 $O_1(x_1, y_1)$ ,  $W_1(W_2)$  为  $O_2$  指向  $O_1(O_3)$  的单位向量。如图 6 所示,3 点连线在障碍物一侧的夹角为

$$\theta' = \alpha + \beta + \gamma = 90^\circ + \beta + \gamma \quad (9)$$

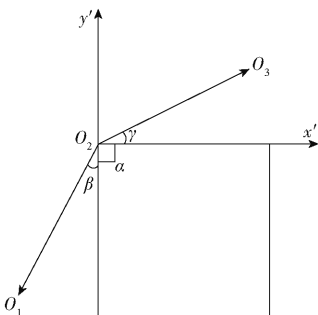


图6 错误点排除示意图

Fig. 6 Error point exclusion diagram

若该转角  $\theta' < 180^\circ$ , 则有

$$\beta + \gamma < 90^\circ \Rightarrow \beta < 90^\circ \text{ 且 } \gamma < 90^\circ \quad (10)$$

因此在图 6 所示的局部坐标系下,  $W_1(W_2)$  总指向第三(一)象限,则 2 个单位向量的和向量  $W_{12}(x'_{12}, y'_{12})$  满足

$$\begin{cases} x'_{12} = \cos\gamma - \sin\beta > \cos(90^\circ - \beta) + \sin\beta = 0 \\ y'_{12} = \sin\gamma - \cos\beta < \sin(90^\circ - \beta) - \cos\beta = 0 \end{cases} \quad (11)$$

由式(11)可知,  $W_{12}$  总指向局部坐标系第四象限,即障碍物应在象限,于是可得符合要求的转角所对应的障碍物坐标  $(x_p, y_p)$  为

$$(x_p, y_p) = \text{round}((x_2, y_2) + W'_{12}) \quad (12)$$

式中  $W'_{12}$  ——  $W_{12}$  的单位向量

$\text{round}()$  —— 四舍五入函数

通过上述方式可判断  $O_2$  处所在转角小于  $180^\circ$  一侧对应栅格是否为障碍物,其他方向判断与此同理。

若不符合,则如图 7 所示,将  $O_1$  作为搜索起点,将  $O_3$  作为搜索终点,并将  $O_2$  和  $O_3$  的连线  $\overline{O_2O_3}$  以一定步长  $s$  进行离散,开始进行搜索。将搜索得到的除  $O_2$  和  $O_3$  以外的平滑路径点,作为  $O_1$  的后续平滑路径点,再以  $O_3$  为搜索起点继续进行平滑路径搜索。其中步长  $s$  为

$$s = \frac{x_3 - x_2}{\text{ceil}(\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2})} \quad (13)$$

式中  $\text{ceil}()$  —— 往正无穷大方向取整的函数

步长  $s$  过长将导致选取平滑路径点时遗漏障碍物。

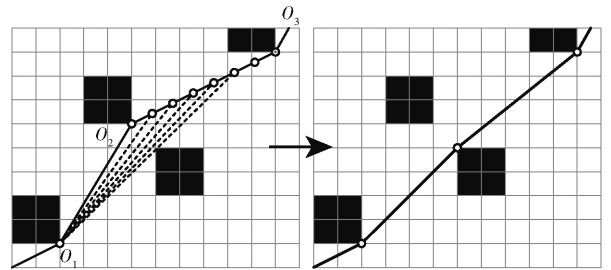


图7 局部平滑示意图

Fig. 7 Local smoothing diagrams

## 2 平滑 JPS 算法

### 2.1 搜索规则

平滑 JPS(SJPS)算法与原 JPS 算法相比有以下区别:

(1) 终止条件。在开放列表第 1 次弹出终点时, SJPS 不停止搜索, 将第 1 次得到的路径(未平滑)的长度作为约束值  $f_2 = l_1$ , 继续扩展所有使启发函数满足  $f(n) \leq f_2$  的跳点, 以遍历已搜索路径, 直到不再有满足要求的跳点。图 8 是一个简单的例子, 图 8a 为 JPS 路径。在 SJPS 中, 终点被弹出前,



只有跳点 1、2、3、4 和起点被弹出, 终点首次被弹出后将计算当前所得路径, 即图 8a 路径的长度, 作为约束值  $f_c$ , 继续扩展开放列表中满足  $f(n) \leq f_c$  的跳点, 因此跳点 5 将继续被扩展。

(2) 更新与记录父节点方式。1 个跳点可能被 2 个以上跳点所扩展, 在 JPS 中, 只记录使启发函数最小的跳点, 旧父节点将被完全舍弃, 而在 SJPS 中, 被舍弃的父节点将被单独保存, 仅在回溯路径时被使用, 并且即使被扩展跳点已处于关闭列表仍然会进行更新。

(3) 路径回溯。在 JPS 中, 终点被弹出后, 从终点开始循环查找父节点, 直到找到起点。在 SJPS 中, 如区别(2)所述, 回溯路径时 1 个节点可能有多个父节点, 此时将对每个父节点均进行路径回溯。如图 8b, 终点有 2 个父节点为跳点 3 和 5, 回溯路径如图 8b 所示两条对称路径。每条回溯的路径, 均对其进行平滑处理, 如图 8c 所示。最后从所获平滑路径中, 选取最理想的路径为最终路径, 如图 8d 所示。

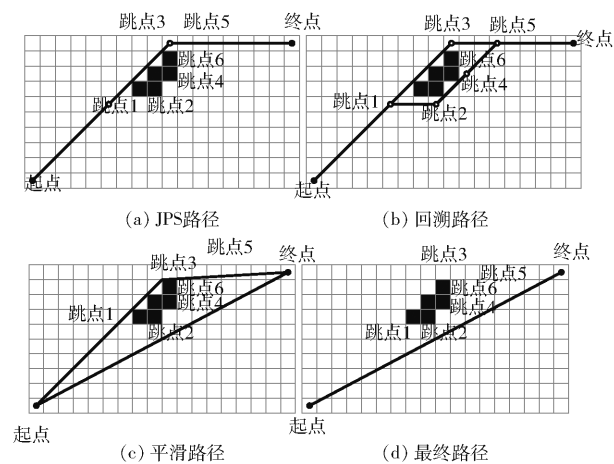


图 8 SJPS 搜索过程

Fig. 8 Search process of SJPS

## 2.2 路径选择

大多数路径规划方法都以长度作为指标进行寻找, 但在一些情况下, 更长一点的路径可能会有更少的转折和总转角。因此, 设立阈值  $L_z$ , 对于每一条回溯的路径, 在平滑后记录其长度和总转折角。先将第 1 条平滑路径作为当前路径长度  $l_{now}$  和总转折角  $\phi_{now}$ , 然后遍历其余平滑路径长度  $l_{other}$  和总转折角  $\phi_{other}$  进行对比, 若满足约束条件

$$\begin{cases} l_{other} - l_{now} \leq L_z \text{ 且 } \phi_{now} - \phi_{other} \geq w_1 L_z \\ l_{other} < l_{now} \text{ 且 } \phi_{other} - \phi_{now} \leq w_2 L_z \end{cases} \quad (14)$$

式中  $w_1$ ——角度倾向权重

$w_2$ ——长度倾向权重

其中之一, 则将  $l_{other}$  和  $\phi_{other}$  作为新的  $l_{now}$  和  $\phi_{now}$ , 对应路径作为新的当前路径, 并继续遍历对比。最后

所得当前路径则为最终路径。

## 3 轨迹优化

### 3.1 基于多段多项式的轨迹优化

直线路径不利于机器人的运动, 在 SJPS 规划路径基础上利用多项式进行轨迹优化处理。将多项式各阶导数作为轨迹的位置、速度( $v$ )、加速度( $a$ )和加加速度( $j$ )等参数, 单段多项式难以适应较为复杂的环境, 因此使用多段高阶多项式可以更好地在各种场景进行轨迹优化。为了限制加速度导数, 使用五次多项式。使用多段多项式平滑路径, 首先将轨迹按路径点分段, 设共分成  $m$  段。每小段路径拟合为一多项式曲线, 则对第  $n$  段路径有

$$f_n(t) = \sum_{i=0}^5 p_i t^i \quad (15)$$

式中  $p_i$ ——多项式系数  $t$ ——时间

对于整条路径, 则有

$$\begin{cases} f_1 = \sum_{i=0}^5 p_{1,i} t^i & (T_0 \leq t \leq T_1) \\ f_2 = \sum_{i=0}^5 p_{2,i} t^i & (T_1 < t \leq T_2) \\ \vdots \\ f_m = \sum_{i=0}^5 p_{m,i} t^i & (T_{m-1} < t \leq T_m) \end{cases} \quad (16)$$

式中  $T_0, T_1, \dots, T_m$ ——路径端点代表的时刻

由式(16)可知, 通过求导可得任意时刻  $t$  的位置( $P=f(t)$ )、速度( $v=f'(t)$ )等参数并施加约束。结合本文平滑路径点的特点, 先约束多项式曲线经过起、止点, 再限制其他路径点只朝向与其接触的障碍物相反的方向(如 1.3 节所述与  $W_{12}$  相反的方向)

$$\begin{cases} f_1(T_0) = P_1 \\ f_m(T_m) = P_{m+1} \\ f_i(T_{i-1}) - S \leq P_i \quad (i=2, 3, \dots, m) \end{cases} \quad (17)$$

式中  $P_i$ ——第  $i$  个路径点

$S$ ——约束放宽的长度

$m$  段多项式对应  $m+1$  个路径点, 约束的放宽对解的质量有重要影响。每个路径点只限制了一段多项式曲线, 因此需施加连续性约束, 使不同段的多项式曲线在路径点处位移、速度和加速度连续

$$f_i^{(k)}(T_i) - f_{i+1}^{(k)}(T_i) = 0 \quad (i=1, 2, \dots, m-1; k=1, 2, 3) \quad (18)$$

可根据需要选取最小化目标函数的导数阶数, 此处以最小化  $j$  为目标以增加运动平稳性, 因此轨迹优化问题将转换为如下最优化问题

$$\begin{cases} \min \int_0^T (f^{(3)}(t))^2 dt \\ \text{s. t. } \begin{cases} f_1(T_0) = P_1 \\ f_m(T_m) = P_{m+1} \\ f_i(T_{i-1}) - S \leq P_i \quad (i = 2, 3, \dots, m) \\ f_i^{(k)}(T_i) - f_{i+1}^{(k)}(T_i) = 0 \\ \quad (i = 1, 2, \dots, m-1; k = 1, 2, 3) \end{cases} \end{cases} \quad (19)$$

该优化问题属于QP问题。优化后的一些物理量可能超过预设值,可对优化后的时间比例乘一系数

$$T'_i = \left( \frac{f_{\max}^k(t)}{M} \right)^{1/k} T_i \quad (i = 0, 1, \dots, m) \quad (20)$$

式中  $M$ ——速度等物理量预期值

以此控制相关物理量的最高值,这不会影响优化的效果<sup>[24]</sup>。

优化时的轨迹存在重新碰撞障碍物的可能,可通过约束多项式端点的方式避开障碍物。如图9所示,当某段轨迹被检测到碰撞后,进行连线检查,当有连线未经障碍物时,将该连线端点,及上一经过障碍物的连线的其中一个端点,3个点作为路径点按前述方法做平滑,得到新的平滑路径点作为轨迹的约束,使轨迹经过该点。

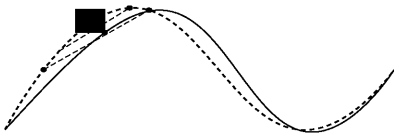


图9 躲避障碍物

Fig. 9 Avoid obstacles

### 3.2 时间分配

上述QP问题,实际上可以看作是寻找最优时间分配问题。通常先给定一个时间分配初值,再用迭代的方法寻找最优解。对于时间初值的给定,目前有采用匀速运动推算时间,即按路径长度比例分

$$T_i = \begin{cases} \frac{s'_i - s_{\text{init}} + S_{\text{Fsum}}}{v_{\max}} + \frac{2\pi}{w_v} - T_{\text{Fsum}} & (S_{\text{Fsum}} \leq s_{\text{init}} \leq \min(S_{\text{Bsum}}, S_{\text{Fsum}} + s'_i)) \\ \frac{2\pi}{w_v} - T_{\text{Fsum}} & (S_{\text{Fsum}} + s'_i \leq s_{\text{init}} \leq S_{\text{Bsum}}) \\ \frac{s'_i}{v_{\max}} & (\min(S_{\text{Fsum}}, S_{\text{Bsum}}) \geq s_{\text{init}}) \\ \frac{s'_i - s_{\text{init}} + S_{\text{Bsum}}}{v_{\max}} + \frac{2\pi}{w_v} - T_{\text{Bsum}} & (S_{\text{Bsum}} \leq s_{\text{init}} \leq \min(S_{\text{Fsum}}, S_{\text{Bsum}} + s'_i)) \\ \frac{2\pi}{w_v} - T_{\text{Bsum}} & (S_{\text{Bsum}} + s'_i \leq s_{\text{init}} \leq S_{\text{Fsum}}) \\ \frac{s'_i - 2s_{\text{init}} + S_{\text{Fsum}} + S_{\text{Bsum}}}{v_{\max}} + \frac{4\pi}{w_v} - T_{\text{Fsum}} - T_{\text{Bsum}} & (\max(S_{\text{Fsum}}, S_{\text{Bsum}}) < s_{\text{init}}) \end{cases} \quad (26)$$

式中  $S_{\text{Fsum}}, S_{\text{Bsum}}$ —— $s'_i$ 之前、之后的位移

配时间的方法<sup>[21]</sup>,虽然能收敛到最优解,但缺乏合理性,初值与最优解偏差较大,将增大迭代难度,增加运算时间。合理的时间初值可以大大减少迭代次数。

针对起止为静止状态的时间分配,设移动机器人从起点开始,平稳加速到最大允许速度时匀速运动,接近终点时再平稳减速。若以最小化 $j$ 为目标,则加速期间加速度应有一个平缓的变化率,因此借助余弦函数设加速度函数为

$$a = \frac{w_j(1 - \cos(w_v t))}{w_v} \quad (21)$$

$$\text{其中 } w_v = \sqrt{\frac{2\pi w_j}{v_{\max}}} \quad (22)$$

式中  $w_j$ ——期望的最大 $j$ 值

$v_{\max}$ ——最大允许速度

且从静止加速到最大允许速度的时间为 $2\pi/w_v$ 。为使加速度不大于给定值 $a_{\max}$ ,令

$$w_j \leq \frac{2\pi a_{\max}^2}{v_{\max}} \quad (23)$$

到达最大允许速度前的位移函数为

$$s_{\text{prol}}(t) = \frac{w_j}{2w_v^2} t^2 + \frac{w_j}{w_v^3} (\cos(w_v t) - 1) \quad (24)$$

优化后的轨迹往往长于初始轨迹,因此将初始路径乘一常数 $w_s$ ,即 $s'_i = w_s s_i$ ,并令 $s_{\text{init}} = s_{\text{prol}}(2\pi/w_v)$ ,对于首段位移的时间分配,有

$$\begin{cases} s'_i = \frac{w_j}{2w_v^2} t_1^2 + \frac{w_j}{w_v^3} (\cos(w_v T_1) - 1) & (s'_i \leq s_{\text{init}}) \\ T_1 = \frac{2\pi}{w_v} + \frac{s'_i - s_{\text{init}}}{v_{\max}} & (s'_i > s_{\text{init}}) \end{cases} \quad (25)$$

末段位移时间的计算与首段位移同理。对于首、末段之间位移 $s'_i$ 的时间分配则有

$$(S_{\text{Fsum}} \leq s_{\text{init}} \leq \min(S_{\text{Bsum}}, S_{\text{Fsum}} + s'_i))$$

$$(S_{\text{Fsum}} + s'_i \leq s_{\text{init}} \leq S_{\text{Bsum}})$$

$$(\min(S_{\text{Fsum}}, S_{\text{Bsum}}) \geq s_{\text{init}})$$

$$(S_{\text{Bsum}} \leq s_{\text{init}} \leq \min(S_{\text{Fsum}}, S_{\text{Bsum}} + s'_i))$$

$$(S_{\text{Bsum}} + s'_i \leq s_{\text{init}} \leq S_{\text{Fsum}})$$

$$(\max(S_{\text{Fsum}}, S_{\text{Bsum}}) < s_{\text{init}})$$

$T_{\text{Fsum}}, T_{\text{Bsum}}$ —— $s'_i$ 之前、之后的累计时间

### 4 仿真与实验分析

Matlab R2019b 提供了 ROS 与导航相关工具箱,为验证本文方法,在 15-6300HQ 型便携式计算机上使用 Matlab R2019b 软件与 ROS 平台对本文算法进行仿真与实验。

#### 4.1 与 Lazy-Theta\* 对比

Theta\* 类算法为得到更短的路径,通常将路径点设置于栅格顶点而非中心点。为方便对比,将 SJPS 的起始点在平滑前设置为当前栅格的栅格顶点,使其与 Lazy-Theta\* 的起始点一致。此处 SJPS 在路径选择中,将  $L_s$  设为 1 倍栅格长度,  $w_1, w_2$  均设为 30,启发函数采用欧几里得距离。Lazy-Theta\* 为八邻域搜索,启发函数同样采用欧几里得距离。在长宽为  $40 \times 40$  场景随机分布 40 个边长为 1.5 倍栅格长度的方形障碍物,以不同精度划分为栅格环境分别进行 20 次实验并取平均值,结果如图 10 所示,图 10 横坐标为横/纵栅格数。

由图 10a 可知,与 Lazy-Theta\* 相比, SJPS 的路径长度稍短,总体较为接近。SJPS 平滑性<sup>[25]</sup>(总转折角)明显低于 Lazy-Theta\*。由图 10b 可知, Lazy-Theta\* 和 SJPS 在低栅格精度时,搜索效率接近,随着栅格精度提高, Lazy-Theta\* 的运行时间因为搜索节点大幅增加而急剧增长,而 SJPS 得益于跳点搜索,运行时间随精度的提高上升较为平缓。其中栅格数为  $400 \times 400$  时, Lazy-Theta\* 的平均运行时间为 21.897 s, SJPS 的平均运行时间为 0.247 s。

#### 4.2 与 JPS 对比

在长宽为  $40 \times 40$  场景随机分布 160 个边长为 1 的方形障碍物并划分为  $100 \times 100$  栅格环境,在该环境进行的 SJPS 与 JPS 对比示例如图 11 所示。

图 11b SJPS 在路径选择中,将  $L_s$  设为 1 倍栅格长度,  $w_1, w_2$  均设为 30,启发函数两者均为欧几里得

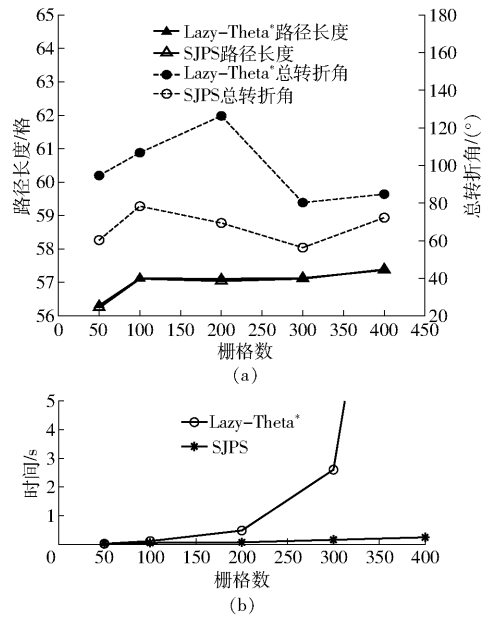


图 10 相同环境下 SJPS 与 Lazy-Theta\* 对比结果

Fig. 10 Results of comparison between SJPS and Lazy-Theta\* under the same environment

距离。如图 11a 所示,仅在后处理对路径进行平滑,无法使路径跨越障碍物,除非将平滑处理变成另一意义上的路径规划,因而很难接近实际最优解。图 11b 中, SJPS 通过挑选有价值的路径,以本文方法逐一进行平滑,再以一定方法进行路径选取,以此可以得到更好的路径。将障碍物数量设为 160 (10% 障碍物)、320 (20% 障碍物) 和 480 个 (30% 障碍物) 分别进行 20 次实验并取平均值,其数据如表 1 所示。由表 1 可知,与平滑后处理的 JPS 相比, SJPS 对长度和总转角有改善作用,运行时间增加 7.06% ~ 10.60%。对使用后处理的 JPS,后处理的时间为 0.8 ~ 1.2 ms。

#### 4.3 轨迹优化

对图 11b 所得 SJPS 路径进行轨迹优化,设起点和终点为静止状态,以栅格长度为单位,允许速度、

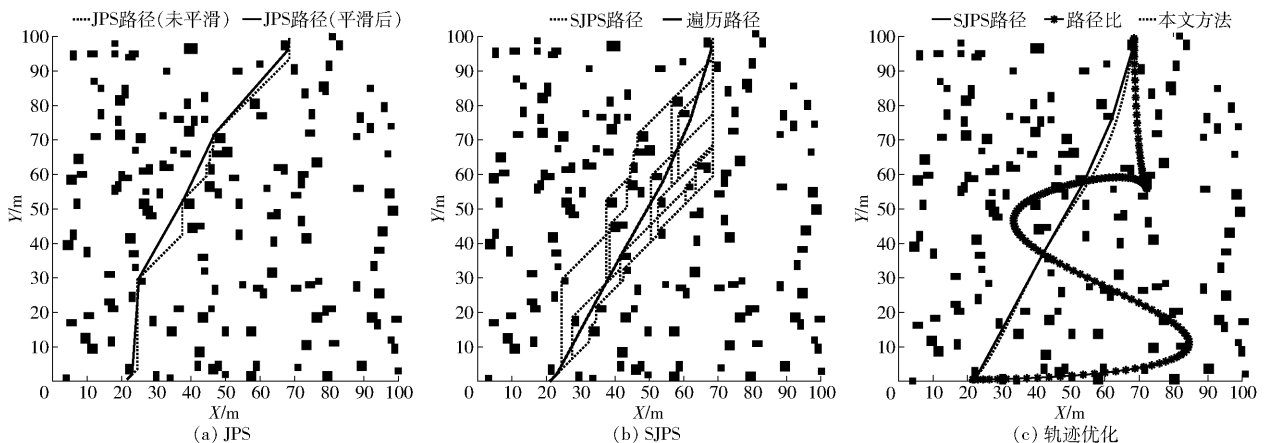


图 11 相同环境下 SJPS 与 JPS 对比结果

Fig. 11 Results of comparison between SJPS and JPS algorithms under the same environment

表1 100 × 100 栅格环境下比较结果

Tab. 1 Results comparison in 100 × 100 grid environment

障碍物比例/%	方法	长度/格	长度变化率/%	总转折角/(°)	总转折角变化率/%	时间/s	时间变化率/%	扩展跳点数/个
10	JPS(平滑后处理)	118.536	-1.80	76.008	-52.75	0.217	10.60	486.600
	SJPS	116.397		35.912		0.240		498.600
20	JPS(平滑后处理)	118.823	-1.11	130.459	-29.34	0.224	8.48	521.500
	SJPS	117.509		92.179		0.243		526.000
30	JPS(平滑后处理)	119.489	-0.48	228.981	-16.93	0.237	7.06	567.375
	SJPS	118.920		190.224		0.255		569.875

加速度最大值分别为  $v_{\max} = 0.25 \text{ m/s}$ 、 $a_{\max} = 0.25 \text{ m/s}^2$ ， $w_j$  为 1， $w_s$  为 1.1，结果如图 11c 所示。其中实线为最终优化结果，虚线为本文方法初始解（未考虑碰撞），与最终解贴合程度较高，星号线为相同总时间下按路径比分配时间的初始解（未考虑碰撞），与最终解偏差较大。两种初始解均能收敛于同一最优解。

图 11c 的 3 种解对应规划轨迹的速度、加速度、

加加速度如图 12 所示。本文方法初始解与最终解的各项参数吻合程度较好，采用路径比分配时间初始解的方法，在首、末段轨迹带来较大冲击，从而影响了其他轨迹。但通过迭代，两种初始解均获得了同一最优时间分配，最终结果符合最小化加加速度的目标。本文所得初始解在第 7 次迭代接近最终解，耗时约 1.2 s。按路径比方法所得初始解，在第 37 次迭代收敛于同一最终解，耗时约 6.5 s。

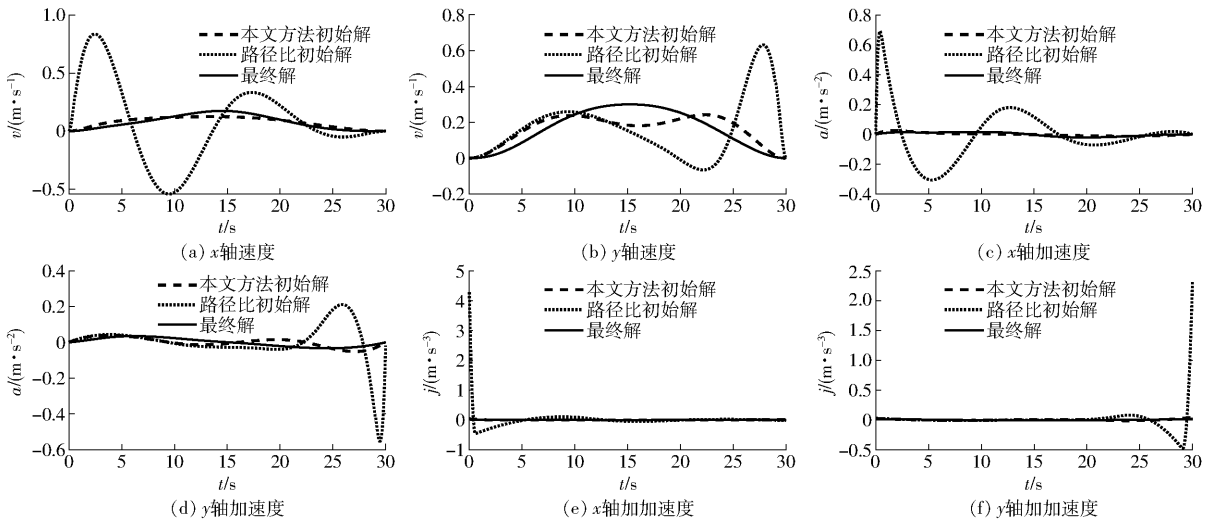


图 12 不同解的速度、加速度、加加速度变化曲线

Fig. 12 Variation of speed, acceleration and jerk of different solutions

#### 4.4 实验验证

使用两轮差速移动机器人，利用 Matlab 与 ROS 对移动机器人进行联合控制并将相关数据发送到 Matlab，在如图 13a 所示的环境进行实验，基于 Gmapping 方法建立如图 13b 所示栅格地图，设膨化

安全距离 0.12 m， $v_{\max} = 0.15 \text{ m/s}$ 、 $a_{\max} = 0.15 \text{ m/s}^2$ ， $w_j$  为 0.5， $w_s$  为 1.05，起点为 (0, -0.98 m)，终点为 (3.04 m, -1.54 m)。实际跟随误差约为 0.015 m，规划最高车速 0.150 m/s，最大车速跟随误差约 0.015 m/s。



图 13 实验环境

Fig. 13 Lab environment



## 5 结论

(1) 提出一种平滑 JPS 路径规划方法, 将改进后的 JPS 算法与本文的平滑方法相结合, 对路径增设了角度指标, 以较小的长度牺牲, 换取平滑性更为良好的路径。结果表明, 在不同障碍物密度环境下 SJPS 相对平滑后处理的 JPS 长度减少了 0.48% ~ 1.80%, 总转折角减少了 16.93% ~ 52.75%, 搜索

时间增加 7.06% ~ 10.60%。与 Lazy-Theta\* 的对比结果表明, 随着栅格环境精度的提高, SJPS 的搜索时间增加较为平缓, 不会出现搜索时间急剧增加的情况, 且 SJPS 路径亦具有良好的平滑性。

(2) 使用多段高阶多项式对初始路径进行优化, 结合余弦函数提出了一种时间分配方法, 可以更好地加快迭代速度, 并结合 SJPS 的特点对多项式端点进行约束。通过仿真及实验验证了本文方法的可行性。

## 参 考 文 献

- [1] CHANG S R, HUH U Y. Curvature-continuous 3D path-planning using QPMI method[J]. International Journal of Advanced Robotic Systems, 2015, 12(6): 76.
- [2] 吴乙万, 黄智. 基于动态虚拟障碍物的智能车辆局部路径规划方法[J]. 湖南大学学报(自然科学版), 2013, 40(1): 33-37. WU Yiwan, HUANG Zhi. Dynamic virtual obstacle based local path planning for intelligent vehicle[J]. Journal of Hunan University (Natural Sciences), 2013, 40(1): 33-37. (in Chinese)
- [3] 黄辰, 费继友, 刘洋, 等. 基于动态反馈 A\* 蚁群算法的平滑路径规划方法[J/OL]. 农业机械学报, 2017, 48(4): 34-40, 102. HUANG Chen, FEI Jiyu, LIU Yang, et al. Smooth path planning method based on dynamic feedback A\* ant colony algorithm [J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(4): 34-40, 102. [http://www.j-csam.org/jcsam/ch/reader/view\\_abstract.aspx?file\\_no=20170404&flag=1](http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20170404&flag=1). DOI: 10.6041/j.issn.1000-1298.2017.04.004. (in Chinese)
- [4] 张超, 李擎, 陈鹏, 等. 一种基于粒子群参数优化的改进蚁群算法及其应用[J]. 北京科技大学学报, 2013, 35(7): 955-960. ZHANG Chao, LI Qing, CHEN Peng, et al. Improved ant colony optimization based on particle swarm optimization and its application[J]. Journal of University of Science and Technology Beijing, 2013, 35(7): 955-960. (in Chinese)
- [5] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [6] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [7] THOREP C E. Path relaxation: path planning for a mobile robot[C]//Proceedings of the National Conference on Artificial Intelligence, AAAI-84, USA: William Kaufmann, 1983: 318-321.
- [8] 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/OL]. 农业机械学报, 2017, 48(7): 22-28. ZHANG Wen, LIU Yong, ZHANG Chaofan, et al. Real-time path planning of greenhouse robot based on directional A\* algorithm[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(7): 22-28. [http://www.j-csam.org/jcsam/ch/reader/view\\_abstract.aspx?file\\_no=20170703&flag=1](http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20170703&flag=1). DOI: 10.6041/j.issn.1000-1298.2017.07.003. (in Chinese)
- [9] 劳彩莲, 李鹏, 冯宇. 基于改进 A\* 与 DWA 算法融合的温室机器人路径规划[J/OL]. 农业机械学报, 2021, 52(1): 14-22. LAO Cailian, LI Peng, FENG Yu. Path planning of greenhouse robot based on fusion of improved A\* algorithm and dynamic window approach[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(1): 14-22. [http://www.j-csam.org/jcsam/ch/reader/view\\_abstract.aspx?file\\_no=20210102&flag=1](http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?file_no=20210102&flag=1). DOI: 10.6041/j.issn.1000-1298.2021.01.002. (in Chinese)
- [10] NASH A, DANIEL K, KOENIG S, et al. Theta\*: anyangle path planning on grids[C]//Proceedings of the AAAI Conference on Artificial Intelligence, Menlo Park, USA: AAAI, 2007: 1177-1183.
- [11] NASH A, KOENIG S, TOVEY C. Lazy Theta\*: any-angle path planning and path length analysis in 3D[C]//24th AAAI Conference on Artificial Intelligence and the 22nd Innovative Applications of Artificial Intelligence Conference, Menlo Park, USA: AAAI, 2010: 147-154.
- [12] SHUNHAO O, HON W L. Strict Theta\*: shorter motion path planning using taut paths[C]//26th International Conference on Automated Planning and Scheduling, Menlo Park, USA: AAAI, 2016: 253-257.
- [13] VIET H D, NGUYEN D T, HOANG H V, et al. Batch-Theta\* for path planning to the best goal in a goal set[J]. Advanced Robotics, 2015, 29(23): 1537-1550.
- [14] HARABOR D, GRASTIEN A. Online graph pruning for pathfinding on grid maps[C]//25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, Menlo Park, USA: AAAI, 2011: 1114-1119.
- [15] HARABOR D, GRASTIEN A. The JPS pathfinding system[C]//5th International Symposium on Combinatorial Search. Menlo Park, USA: AAAI, 2012: 207-208.
- [16] HARABOR D, GRASTIEN A. Improving jump point search[C]//24th International Conference on Automated Planning and Scheduling, Menlo Park, USA: AAAI, 2014: 128-135.
- [17] REEDS J A, SHEPP L A. Optimal paths for a car that goes both forwards and backwards[J]. Pacific Journal of Mathematics, 1990, 145(2): 367-393.

- jcsam/ch/reader/view\_abstract.aspx? file\_no = 20130618&flag = 1. DOI: 10. 6041/j. issn. 1000-1298. 2013. 06. 018. (in Chinese)
- [16] LACEY P M C. Developments in the theory of particle mixing[J]. *Journal of Applied Chemistry*, 1954, 4(5): 257 - 268.
- [17] 毛娅, 范雄, 江智. 基于 EDEM 的连续性立式搅拌机混合特性研究[J]. *计算机仿真*, 2018, 35(12): 181 - 184, 415. MAO Ya, FAN Xiong, JIANG Zhi. A study of the mixing characteristics in the continuous vertical blender based on EDEM[J]. *Computer Simulation*, 2018, 35(12): 181 - 184, 415. (in Chinese)
- [18] ACA J, MILADA P, LATO P, et al. DEM/CFD analysis of granular flow in static mixers[J]. *Powder Technology*, 2014, 266: 240 - 248.
- [19] LEMIEUX M, LÉONARD G, DOUCET J, et al. Large-scale numerical investigation of solids mixing in a v-blender using the discrete element method[J]. *Powder Technology*, 2008, 181(2): 205 - 216.
- [20] 苑进, 刘勤华, 刘雪美, 等. 配比变量施肥中多肥料掺混模拟与掺混腔结构优化[J/OL]. *农业机械学报*, 2014, 45(6): 125 - 132. YUAN Jin, LIU Qinhua, LIU Xuemei, et al. Simulation of multi-fertilizers blending process and optimization of blending cavity structure in nutrient proportion of variable rate fertilization [J/OL]. *Transactions of the Chinese Society for Agricultural Machinery*, 2014, 45(6): 125 - 132. [http://www.j-csam.org/jcsam/ch/reader/view\\_abstract.aspx? file\\_no = 20140620&flag = 1](http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx? file_no = 20140620&flag = 1). DOI: 10. 6041/j. issn. 1000-1298. 2014. 06. 020. (in Chinese)
- [21] 苑进, 辛振波, 牛子孺, 等. 基于 RVM 的配比变量排肥掺混均匀度离散元仿真及验证[J]. *农业工程学报*, 2019, 35(8): 37 - 45. YUAN Jin, XIN Zhenbo, NIU Ziru, et al. Discrete element model simulation and verification of fertilizer blending uniformity of variable rate fertilization based on relevance vector machine[J]. *Transactions of the CSAE*, 2019, 35(8): 37 - 45. (in Chinese)
- [22] LAMMERS B, HEINRICHS J, BUCKMASTER D. Method helps in determination of forage, TMR particle size requirements for cattle [J]. *Feedstuffs*, 1996, 68(41): 14 - 20.
- [23] 田富洋, 陈玉华, 宋占华, 等. 自走式全混合日粮制备机设计与试验[J/OL]. *农业机械学报*, 2020, 51(3): 106 - 114. TIAN Fuyang, CHEN Yuhua, SONG Zhanhua, et al. Design and experiment of self-propelled TMR preparation mixer for silage straw feed[J/OL]. *Transactions of the Chinese Society for Agricultural Machinery*, 2020, 51(3): 106 - 114. [http://www.j-csam.org/jcsam/ch/reader/view\\_abstract.aspx? file\\_no = 20200312&flag = 1](http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx? file_no = 20200312&flag = 1). DOI: 10. 6041/j. issn. 1000-1298. 2020. 03. 012. (in Chinese)
- [24] RIPPEL C M, JORDAN E R, STOKES S R. Evaluation of particle size distribution and ration uniformity in total mixed rations fed in northcentral texas 1, 2[J]. *The Professional Animal Scientist*, 1998, 14(1): 44 - 50.
- [25] 王德福, 张全国. 全混合日粮混合质量评价指标试验方法的研究[J]. *农业工程学报*, 2007, 23(5): 126 - 129. WANG Defu, ZHANG Quanguo. Experimental methods for evaluating indexes of processing quality of total mixed ration[J]. *Transactions of the CSAE*, 2007, 23(5): 126 - 129. (in Chinese)

~~~~~

(上接第 29 页)

- [18] 程英英. 基于微分平坦的轮式移动机器人轨迹规划[D]. 长春: 吉林大学, 2008. CHENG Yingying. Trajectory planning of wheeled mobile robot based on differential flatness[D]. Changchun: Jilin University, 2008. (in Chinese)
- [19] SHOMIN M, HOLLIS R. Fast, dynamic trajectory planning for a dynamically stable mobile robot[C] // 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2016: 3636 - 3641.
- [20] 袁静妮, 杨林, 唐晓峰, 等. 基于改进 RRT\* 与行驶轨迹优化的智能汽车运动规划[J/OL]. *自动化学报*, 2020, 46(X): 1 - 10. YUAN Jingni, YANG Lin, TANG Xiaofeng, et al. Autonomous vehicle motion planning based on improved RRT\* algorithm and trajectory optimization [J/OL]. *Acta Automatica Sinica*, 2020, 46(X): 1 - 10. DOI: 10. 16383/j. aas. c190607. (in Chinese)
- [21] MELLINGER D, KUMAR V. Minimum snap trajectory generation and control for quadrotors [C] // IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2011: 2520 - 2525.
- [22] LEOBARDO C M, DAVID G G, RODRIGO A L, et al. A hybrid method for online trajectory planning of mobile robots in cluttered environments[J]. *IEEE Robotics and Automation Letters*, 2017, 2(2): 935 - 942.
- [23] CHEN J, LIU T, SHEN S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments[C] // IEEE International Conference on Robotics & Automation, Piscataway, USA: IEEE, 2016: 1476 - 1483.
- [24] 孟德, 王田苗. 一种移动机器人全局最优路径规划算法[J]. *机器人*, 2008, 30(3): 217 - 222. MENG Cai, WANG Tianmiao. A global optimal path planning algorithm for mobile robot[J]. *Robot*, 2008, 30(3): 217 - 222. (in Chinese)
- [25] RICHTER C, BRY A, ROY N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments[C] // 16th International Symposium of Robotics Research, Springer Verlag, 2016: 649 - 666.