

doi: 10.7690/bgzdh.2026.04.019

改进 RRT 算法的移动机器人路径规划

党涛, 曹东

(南京航空航天大学自动化学院, 南京 210006)

摘要: 针对传统快速随机搜索树 (rapidly-exploring random tree, RRT) 算法在复杂环境中的收敛速度较慢、路径规划质量较低且在转弯时可能不符合机器人运动学约束导致转弯质量较低等问题, 提出在 RRT 算法的采样基础上融合 Dijkstra 算法的思想。改进后的算法由 RRT 算法遍历地图构成初始无向图路径图后, 再引入 Dijkstra 算法的贪心思想来搜索代价更小更优的路径。在保证避障的前提下, 对搜索得到的路径使用贝塞尔曲线进行平滑处理, 并将改进后的算法应用于仿真环境。结果表明, 改进算法规划能力优于原算法。

关键词: 路径规划; RRT 算法; Dijkstra 算法; 贝塞尔曲线; 移动机器人

中图分类号: TP242.6 **文献标志码:** A

Path Planning of Mobile Robot Based on Improved RRT Algorithm

Dang Tao, Cao Dong

*(College of Automation Engineering, Nanjing University of
Aeronautics and Astronautics, Nanjing 210006, China)*

Abstract: In view of the problems of slow convergence speed of traditional rapidly-exploring random tree (RRT) algorithm in complex environments, low path planning quality, and may cause problems such as low turning quality when cornering. The idea of integrating Dijkstra algorithm on the sampling basis of RRT algorithm is proposed. The improved algorithm is composed of the RRT algorithm traversing the map to from the initial undirected graph path map, and then introduces the greedy idea of the Dijkstra algorithm to search for a less expensive and better path. Under the premise of ensuring obstacle avoidance, the path obtained by the search is smoothed by using Bezier curve. After applying the improved algorithm to the simulation environment, the results show that the planning ability of the improved algorithm is better than the traditional RRT algorithm.

Keywords: path planning; RRT algorithm; Dijkstra algorithm; Bezier curve; mobile robot

0 引言

近年来, 机器人智能导航技术研究成为热门, 其中路径规划作为机器人导航的核心技术更加受到研究人员的重视。为了使得机器人能够快速平稳且安全地运行到目标位置, 为机器人运动规划出一条合理无障碍的轨迹是路径规划主要内容^[1]。根据对周围环境信息掌握程度的不同, 路径规划可分为局部路径规划和全局路径规划。局部路径规划算法主要包括人工势场法^[2]、动态窗口法^[3]和神经网络法^[4]等, 其中人工势场法因其实时性好、算法简单等优点而受到广泛关注。由于人工势场法很容易落入局部最小值, 并且在障碍物处震荡, 限制了它的实际应用价值。全局路径规划主要分为基于搜索和基于随机采样的 2 个方向。基于搜索的路径规划算法是根据启发函数中备用节点的集合中来寻找下一个节点, 不同的启发函数也就有不同的搜索类算法, 典型算法包括 Dijkstra 算法^[5]、A*算法^[6]和跳点搜索

(jump point search, JPS) 算法^[7]等。Dijkstra 算法是第一个规划搜索算法, 它主要利用贪心思想来寻找图中的最短路径。1968 年, Hart 通过对 Dijkstra 算法添加一个启发式函数, 创建了 A*算法, 从而加速了搜索的速度。JPS 算法建立在 A*算法模型的基础上, 优化了 A*算法中搜索后续节点的操作。JPS 算法相比 A*算法是一种更为高效的搜索算法。基于随机采样的算法通过在状态空间中进行随机采样, 不断扩展并最终覆盖整个状态空间, 从而获得一条路径的算法。典型算法包括快速随机搜索树 (RRT) 算法^[8]和概率路线图算法 (PRM) 等。基于采样的算法可以避免配置空间的离散化, 从而对高维空间具有良好的可扩展性。RRT 算法因其高效单查询机制和易于扩展应用得到了广泛使用, 但 RRT 算法得到的路径并非最优且由于是随机规划, 规划的效率并不是很高。当前, 有很多研究成果针对 RRT 算法的不足提出改进算法。Karaman 等^[9]提出渐进最优性

收稿日期: 2024-12-03; 修回日期: 2025-01-17

第一作者: 党涛(1998—), 男, 内蒙古人, 硕士。

(最短距离)的 RRT*算法,该算法在 RRT 基础上添加了随机几何图与枝减优化理论;文献[10]提出删除 RRT 中低性能节点来规划高维机器人路径;文献[11]通过引入局部优化算法来提高效率,并且提高路径的平滑性;文献[12]提出双向随机树来规划高维度路径由起点和终点分别生成一个自己的随机树,直到二者相遇即为所需路径,虽然提高了速度,但不能得到最优路径;文献[13]提出目标偏置的思想,即将 RRT 算法与人工势场法相结合,缩短了路径长度和节点数,但在通过狭窄的渠道时仍存在问题。

以上文献均对 RRT 算法提出了一定的优化,但仍然存在搜索效率低、搜索路径非最优,且没有考虑到拐点处转向是否满足机器人运动学约束等问题,算法的实用性仍需提高。笔者综合考虑 RRT 算法的优缺点和移动机器人在实际环境中转弯段行进稳定性限制,针对 RRT 算法以上缺点,利用 Dijkstra 算法和贝塞尔曲线对 RRT 算法进行优化,使得机器人能够得到更优路径且更符合机器人运动学约束,从而平稳到达终点。

1 理论基础

传统 RRT 算法是一种基于采样随机性算法^[14]。如图 1 所示。

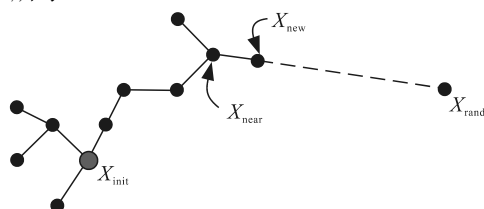


图 1 RRT 算法原理

从起点进行扩展,在空间中通过随机函数采样得到随机点,利用节点扩展策略得到新节点,再经碰撞检测判断是否将新节点加入随机树中,最后成功找到路径。RRT 算法可以直接应用于非完整约束系统的规划,不需要进行路径转换,它的算法复杂度较小,适用于高维多自由度的系统。

RRT 算法的基本步骤为:

- 1) 采样空间中起点 X_{init} , 并且以 X_{init} 作为随机搜索树的根节点;
- 2) 在采样空间中由随机算法生成一个随机采样点 X_{rand} ;
- 3) 判断生成的 X_{rand} 是否落在障碍物中,若在障碍物中,舍弃该点并重复步骤 2);
- 4) 遍历随机搜索树找到距离 X_{rand} 最近的节点 X_{near} ;

5) 由节点 X_{near} 沿着节点 X_{rand} 方向按固定扩展步长生成新的节点 X_{new} 作为新的树节点。如果在扩展过程中遇到扩展点落在障碍物的情况则重复上述过程。直到目标点变为树中节点或者超过迭代次数搜索结束。由目标点回溯到起点便可得到路径。

2 改进 RRT 算法

2.1 Dijkstra 算法

Dijkstra 算法是典型的最短路径搜索算法,用于计算一个节点到其他节点的最短路径。Dijkstra 算法采用的是一种贪心策略^[15],主要特点是以起点为中心向外层层扩展,直到扩展到终点为止。它的主要思想为:1) 声明一个起始点,由起始点开始计算距离代价;2) 引入 2 个合集 S 和 T ,其中 S 用来保存起点到已经找到的各个顶点之间的最短距离, T 用来保存未找到路径的顶点合集;3) 起初 S 中只有起点, T 中保存除起点外的其他顶点,从 T 中选择距离起点最近的点加入集合 S ,并更新 T 中点的距离代价。重复上述操作完成遍历 T 中的顶点。

2.2 改进 RRT 算法

由于 RRT 算法的采样过程具有随机性^[16],算法存在一些缺陷:1) 不稳定性,在相同的地图环境、起始点的条件下,RRT 算法所生成的路径是不同的,生成的不均匀路径不能直接由机器人执行使用。2) 不定向性,在路径扩展过程中,随机节点分布在环境地图上的各个位置,路径不能够快速地收敛到目标节点,这使得算法的搜索时间大大增加。为了提高 RRT 算法的搜索路径效率和平稳性,利用 Dijkstra 算法和贝塞尔曲线对 RRT 算法进行改进,如图 2 所示。

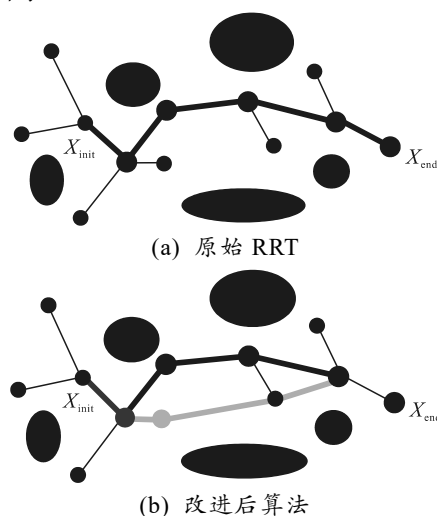


图 2 改进后算法原理

基于 2 种算法的优缺点，提出一种融合算法，通过 2 种算法的优势互补，寻得一条可行的、距离更短的路径。利用 RRT 算法寻得一条初始可行的路径，对于通过 RRT 算法所得的初始路径上利用 Dijkstra 算法对初始路径进行优化。由 RRT 算法特性可知初始所得的路径并非最优，将 RRT 算法随机树扩展的过程中不在初始路径中的节点通过 Dijkstra 算法对这些未被规划到初始路径的结点进行重新评估利用。通过 Dijkstra 的贪心策略，对于扩展节点进行距离代价评估(由 2 点之间的距离作为评估代价)，以代价更小的节点且没有在障碍物内的节点代替初始路径中的节点，以此来利用 Dijkstra 对路径进行更新。如 2 图(a)所示，其中加粗的黑色路径为 RRT 算法规划的初始路径。在初始路径的基础上，经过 Dijkstra 算法对路径进行重新规划，对于路径周围扩展节点进行重新规划得到新的路径，如图 2(b)所示，加粗的灰色路径即为更新后的路径。

利用 Dijkstra 算法对所得路径以及路径周围没有加入树中的节点进行更新，将扩展树周围扩展出的未加入树的点个数小于等于 50 h，加入到 Dijkstra 算法 T 集合中，上一个父节点作为起点在集合 S 中保存，由 Dijkstra 算法获得路径周围代价最低的点，将新得到的点代替原父节点，加入到随机搜索树中，继续扩展。以此来得到距离代价更低的路径。改进后算法总体流程如图 3 所示。

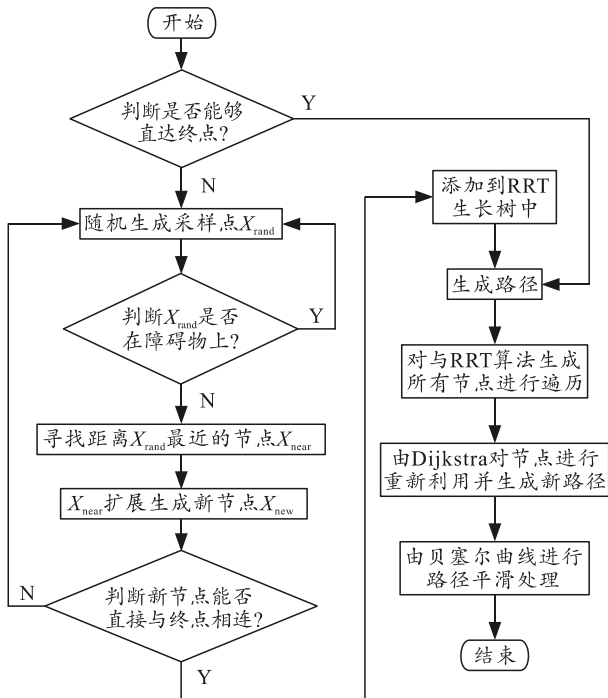


图 3 改进后算法流程

2.3 利用贝塞尔曲线对路径进行优化

贝塞尔曲线^[17]是一种通过连接向量来表示曲线的方法，即画曲线前先通过向量绘制一个多边形，代表该曲线的趋势和走向。利用改进后 RRT 算法所得的路径具有一定的不平滑性使得机器人的能量消耗变大，其中的某些拐点可能并不符合机器人的动力学约束，使得机器人的转弯效率低下，无法直接使用于机器人的实际行走路径。而贝塞尔曲线由于其灵活性强易于计算等优点，在路径规划中得到广泛应用。N 阶贝塞尔曲线的表达式如下：

$$B(\tau) = \sum_{i=0}^n P_i B_{i,n}(\tau) = \sum_{i=0}^n P_i \frac{n!}{(n-i)!i!} (1-\tau)^{n-i} \tau^i, \tau \in [0, 1]. \quad (1)$$

式中： P_i 为 $(n+1)$ 个控制位置点； τ 为曲线自变量； n 为贝塞尔曲线的阶数； $B_{i,n}$ 为 n 次伯恩斯坦多项式，并且其满足：

$$B_{i,n}(u) = C_n^i u^i (1-u)^{n-i} = \frac{n!}{(n-i)!i!} u^i (1-u)^{n-i}, i = 0, 1, 2, \dots, n. \quad (2)$$

为保证机器人能够较为平稳的进行转弯运动，采用三阶贝塞尔曲线对于运动路径中的拐点进行平滑性处理，其中 3 次贝塞尔曲线的表达式为：

$$P(u) = P_0(1-u)^3 + 3P_1(1-u)^2u + 3P_2(1-u)u^2 + u^3P_3. \quad (3)$$

式中 $P_i = (x_0, y_0) \sim (x_3, y_3)$ 为 4 个控制点的坐标。

如图 4 所示，由 4 个控制点确定的平面三阶贝塞尔曲线。

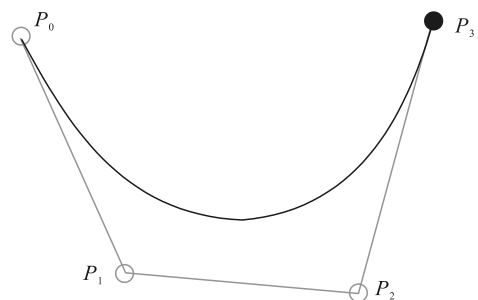


图 4 三阶贝塞尔

曲线经过第 1 个控制点以及第 4 个控制点，即 $P(0)=P_0, P(1)=P_3$ ；曲线在端点处的切向量为 $P'(0)=3(P_1-P_0), P'(1)=3(P_3-P_2)$ 。曲线在任意一点的曲率为：

$$k(u) = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'^2(u) + y'^2(u))^{3/2}}. \quad (4)$$

为了满足机器人运动学的约束，需要对贝塞尔曲线的曲率进行约束。假设机器人在拐点处半径为

R , 那么确定贝塞尔曲线的曲率最大值为 $1/R$; 曲率的最小值可取 0 , 那么曲率的有界条件则可以表示为 $0 < k < 1/R$. 通过曲率的公式可知: 曲率仅在 $x'(u)=y'(u)=0$ 时不连续, 如果假设条件成立, 在 u 之后曲线将会退化成一个点, 那么曲线将无法到达控制点 P_3 , 这与贝塞尔曲线的定义相互矛盾, 可以确定三阶贝塞尔曲线的轨迹是处处连续的。

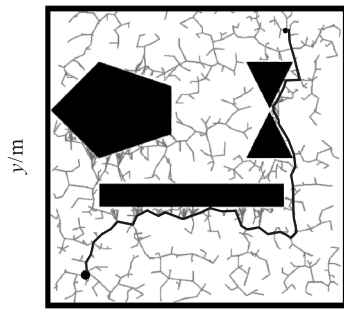
针对规划路径在拐点处不平滑的问题, 在避障的前提下选择利用三阶贝塞尔曲线对规划轨迹的拐点处进行平滑处理, 通过仿真实验可以验证优化后的路径转弯更加平滑, 使得机器人消耗降低且更符合机器人的运动约束。

3 仿真实验及分析

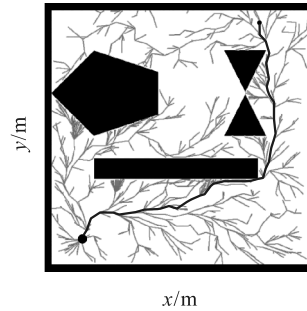
为了验证本文中算法的可行性, 分别对 RRT 算法、RRT*算法和改进后 RRT 算法进行仿真验证。分别在简单障碍物环境和复杂障碍物环境下对 3 种算法进行对比实验验证。本文中实验地图是以实验室内的障碍物为基础所建立。

3.1 简单障碍物环境实验

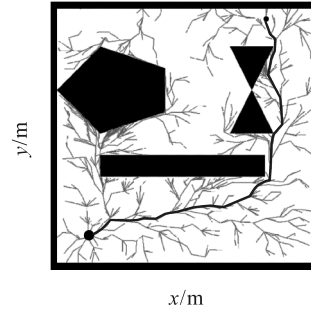
简单障碍物环境地图大小为 $x=y=600\text{ m}$, 规划的起点为 $(60, 70)$, 终点为 $(560, 480)$, 扩展步长为 10 m . 当采样节点数大于 $10\ 000$ 时停止搜索。图 5 为简单障碍物环境地图中的仿真实验。



(a) RRT 算法路径规划



(b) RRT*算法路径规划



(c) 改进后 RRT 算法路径规划

图 5 简单障碍物环境仿真结果对比

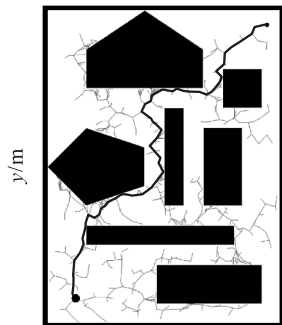
图 5 中下方大的黑色点表示路径起点, 上方小黑色点表示路径终点。经过对比得到, 改进后算法和原算法相比得到的路径更优, 路径代价更低且生成的路径更加平滑。表 1 中记录了每个算法 50 次试验的平均数据。

表 1 简单障碍物环境仿真实验数据

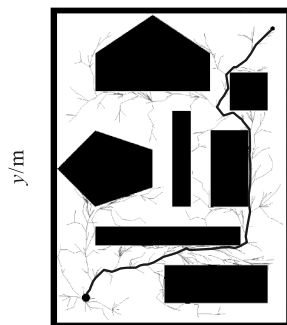
算法	节点数	路径长度 l/m	搜索时间 t/s
RRT	1 006	1 146.7	1.34
RRT*	928	1 089.6	1.16
改进 RRT	776	828.5	0.96

3.2 复杂障碍物环境实验

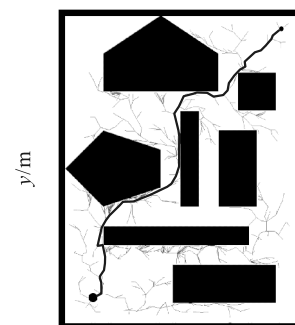
复杂障碍物地图大小为 $x=600\text{ m}$, $y=800\text{ m}$, 规划的起点为 $(70, 60)$, 终点为 $(560, 760)$. 当采样节点数大于 $15\ 000$ 时停止搜索, 图 6 中依次对比 3 种算法在复杂环境地图中的仿真结果。表 2 为每个算法 50 次实验的平均数据。



(a) RRT 算法路径规划



(b) RRT*算法路径规划



(c) 改进后 RRT 算法路径规划

图 6 复杂障碍物环境仿真结果对比

表 2 复杂障碍物环境仿真实验数据

算法	节点数	路径长度 l/m	搜索时间 t/s
RRT	1 389	1 520.6	2.35
RRT*	1 209	1 326.4	2.04
改进 RRT	980	976.7	1.97

根据表 1 和 2 中的数据可以看出：在简单障碍物环境中，改进后算法与 RRT 算法相比，路径节点数减少了 22%，路径距离长度缩短了 27.89%；与 RRT*算法相比路径节点数减少了 16.4%，路径长度缩短了 23.97%。在复杂环境地图中，改进后算法与 RRT 算法相比，路径节点数减少了 29.5%，路径距离长度缩短了 35.79%；与 RRT*算法相比路径节点数减少了 18.9%，路径长度缩短了 26.39%。搜索时间并没有很大差距。可以看出：在改进后算法所获得的路径相比原算法在距离上有了很大提升，相比原算法可以得到距离更短、代价更低的路径。

3.3 贝塞尔曲线对路径优化对比实验

在得到初始运动轨迹后，利用贝塞尔曲线对所得轨迹拐点处进行优化。图 7 为优化前后轨迹对比实验结果。

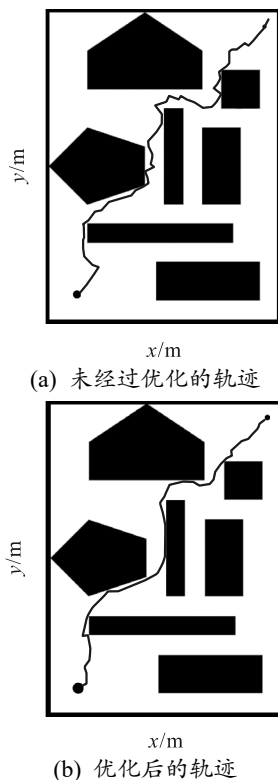


图 7 贝塞尔曲线优化后对比

通过对路径拐点处的对比，可以看出：经过贝塞尔曲线优化后的路径，在拐点处的转弯更加平滑，提高了路径在拐点处的质量，更加有利于机器人在实践中进行应用，同时也更加符合机器人运动学

约束。

4 结论

笔者提出一种改进 RRT 路径规划算法。由仿真结果可知，改进 RRT 算法能够在保证生成最优路径的同时，得到更加平滑的运动轨迹，更有利于在机器人实践中应用。

参考文献：

- [1] 买智强, 辛舟, 张雪琪. 基于改进传统 RRT 算法的机器人路径规划研究[J]. 机械制造与自动化, 2022, 51(6): 177-179.
- [2] 胡铮, 徐斌. 改进人工势场法的轨迹规划[J]. 电光与控制, 2023, 30(3): 38-41, 53.
- [3] 辛鹏, 马希青. 融合改进 RRT 和动态窗口法的路径规划算法[J]. 机床与液压, 2022, 50(23): 20-24.
- [4] 陈秋莲, 郑以君, 蒋环宇. 基于神经网络改进粒子群算法的动态路径规划[J]. 华中科技大学学报(自然科学版), 2021, 49(2): 51-55.
- [5] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1: 269-271.
- [6] 郭翰卿, 付丽霞, 张勇, 等. 安全性 A*算法融合动态窗口法的路径规划[J]. 兵工自动化, 2022, 41(12): 41-48.
- [7] 周熙栋, 张辉, 陈波. 非结构化场景下基于改进 JPS 算法的移动机器人路径规划[J]. 控制与决策, 2024, 39(2): 474-482.
- [8] 杜明博, 梅涛, 陈佳佳, 等. 复杂环境下基于 RRT 的智能车辆运动规划算法[J]. 机器人, 2015, 37(4): 443-450.
- [9] KARAMAN S, FRAZZOLI E. Incremental sampling-based algorithms for optimal motion planning[C]//Science and Systems VI. Robotics: Science and Systems Foundation, 2010: 267-274.
- [10] ADIYATOV O, VAROL H A. Rapidly-exploring random tree based memory efficient motion planning[C]//2013 IEEE International Conference on Mechatronics and Automation. Takamatsu, Japan: IEEE, 2013: 354-359.
- [11] CHOUDHURY S, GAMMELL J D, BARFOOT T D, et al. Regionally accelerated batch informed trees(RABIT*): a framework to integrate local information into optimal path planning[C]//2016 IEEE International Conference on Robotics and Automation. Stockholm, Sweden: IEEE, 2016: 4207-4214.
- [12] LAVALLE S M, KUFFER J J. Randomized kinodynamic planning[C]//Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). Detroit, MI, USA: IEEE, 1999: 473-439.