

复杂环境下六自由度机械臂路径规划的 Biased-RRT 修正算法

陈志勇, 黄泽麟, 曾德财, 于潇雁

(福州大学机械工程及自动化学院, 福建 福州 350108)

摘要:为解决复杂环境下六自由度机械臂的路径规划问题,提出一种基于采样规则目标导向设计、父节点重选的 Biased-RRT 修正算法.该算法在原目标偏置策略的基础上对随机采样点的选取规则进行重新设定,引导算法搜索树在尽可能向目标区域扩展的同时有效避开复杂障碍物.在节点扩展方面,依据新节点距离目标点的远近采用变步长扩展方式,即在距离远时选用大步长,加快搜索树扩展;进入目标区域后选用小步长,防止节点扩展陷入局部死循环.在路径优化方面,本算法通过引入基于路径代价最小的重选父节点操作及多余路径节点剔除操作,使规划出的路径相对优化.最后,利用 3 次样条插值技术为机械臂各关节规划出一条光滑、连续且无障的运动曲线.仿真结果表明,本算法可有效缩短路径规划时间、减少路径长度,较好地完成了复杂环境下六自由度机械臂的预期路径规划任务.

关键词:机械臂;路径规划;Biased-RRT 修正算法;父节点重选;样条插值

中图分类号:TP241

文献标识码:A

Biased-RRT modified algorithm for path planning of a 6-DOF manipulator in complex environment

CHEN Zhiyong, HUANG Zelin, ZENG Decai, YU Xiaoyan

(School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, Fujian 350108, China)

Abstract: To solve the path planning problem of a 6-DOF manipulator in complex environment, a Biased-RRT modified algorithm based on the target-oriented design of the sampling rule and the reselection of parent node was proposed. On the basis of the original target bias strategy, the selection rules of random sampling points were reset in the algorithm to guide the search tree to expand to the target area as much as possible while effectively avoiding complex obstacles. In terms of node expansion, the variable step length expansion method was adopted according to the distance between the new node and the target point, that is, when the distance was long, a large step length was selected to speed up the expansion of the search tree. A small step length was selected after entering the target area to prevent the node expansion from falling into local death cycle. In terms of path optimization, the planned path was relatively optimized in the proposed algorithm by reselecting the parent node based on the minimum path cost and the removal of redundant path nodes. Finally, the cubic spline interpolation technique was used to plan a smooth, continuous and obstacle-free motion curve for each joint of the manipulator. The simulation results show that the proposed algorithm can effectively shorten the path planning time, reduce the path length, and better complete the expected path planning task of the 6-DOF manipulator in complex environment.

Keywords: manipulator; path planning; Biased-RRT modified algorithm; parent reselection; spline interpolation

0 引言

机械臂路径规划是指机械臂在满足系统运动学约束的前提下,在所有关节运动范围内规划出一条机

收稿日期:2021-10-10

通信作者:陈志勇(1984-),副教授,主要从事机器人动力学与控制研究,chenzhiyong0203@foxmail.com

基金项目:福建省自然科学基金资助项目(2020J01450)

<http://xbzrb.fzu.edu.cn>

机械臂自身不发生碰撞且不与外界障碍物碰撞的运动路径. 目前机械臂的避障路径规划算法主要有: A* 搜索算法^[1-3]、人工势场法^[4-5]、快速扩展随机树(rapidly exploring random tree, RRT)算法、智能算法^[6-8]等.

A* 搜索算法采用启发函数来确定路径搜索方向, 如贾庆轩等^[9]把障碍物模型映射到机械臂关节空间, 运用 A* 算法完成路径规划, 不过该算法在面对多自由度机械臂避障路径问题时, 计算效率较低, 难以满足实时性要求. 人工势场法^[10]引入势场思想, 建立目标引力和障碍物斥力, 促使探索的路径远离障碍物并接近目标, 从而完成路径规划, 不过该算法对于障碍物较多的高维空间工况, 其计算复杂度骤增且容易陷入局部极小值. 马培羽等^[11]将人工势场法运用到机械臂上, 人为添加虚拟势场, 推动其算法向目标位置靠近, 从而离开局部最优值, 但对于高维空间工况, 依旧存在计算复杂的问题, 难以处理多障碍物环境. 此外, 包括蚁群算法、遗传算法、粒子群算法等在内的各类智能算法虽然各有优点, 但均面临着算法计算效率不高、实时性差等突出问题. RRT 算法^[12]采用随机采样的方式扩展树状结构来探索空间, 只对扩展的节点进行碰撞检测, 避免障碍物的空间描述, 因此对于高维空间也有良好的效果, 被越来越多地应用于机械臂路径规划中^[13-16]. 但随机采样的扩展方式更倾向于探索整个空间, 计算效率不稳定. 近年来, 许多学者尝试从不同方面对 RRT 算法进行改进. 如文献^[17]引入目标偏置思想, 提出 Biased-RRT 算法, 在随机采样的基础上有一定小概率采样到目标区域, 引导路径搜索向目标区域进行, 从而提高算法效率, 但算法并未对路径进行优化, 所得路径并非最优. 文献^[18]提出结合路径长度和路径安全性建立路径代价函数来约束随机采样点, 并用贪婪策略来弥补 RRT 算法局部扩展缓慢的缺点, 但随着节点扩展, 路径代价函数的计算量明显增多, 算法规划时间无法保证. Karaman 等^[19]提出了 RRT* 算法, 在 RRT 算法中添加重新布线和代价函数, 通过不断地更新路径, 可以找到一条代价较小或者最小的路径, 但需要耗费的计算资源大幅度提升.

为此, 本研究提出一种六自由度机械臂避障路径规划的 Biased-RRT 修正算法. 该算法通过对传统目标偏置策略中的随机点采样规则的目标导向设计, 来确保搜索树尽可能快地向目标点扩展, 且避免了算法陷入局部极小值问题; 借鉴 RRT* 算法中的父节点重选操作及多余路径节点的剔除操作, 来获得路径代价最小的路径点及其简化路径. 再者, 采用 3 次样条插值技术为机械臂各关节规划出一条光滑、连续且无障的运动路径曲线. 系统数值仿真及 3D 模拟仿真结果, 均证实所提 Biased-RRT 修正算法的有效性.

1 六自由度机械臂路径规划算法设计

1.1 问题描述

要实现复杂环境下六自由度机械臂的避障路径规划, 实际上是要在满足六自由度机械臂运动学约束的前提下, 在其运动起点到运动终点之间规划出一条复杂环境下无碰撞、合理的可行路径, 并在此基础上尽可能做到加快规划时间、优化路径质量, 让所提路径规划算法能够更好地适用于六自由度机械臂的实时控制. 为此, 本节提出一种复杂环境下六自由度机械臂路径规划的 Biased-RRT 修正算法, 并分别从机械臂逆运动学分析, Biased-RRT 修正算法的采样规则设计、新节点扩展、路径优化的父节点重选, 以及算法整体流程等方面进行详细阐述.

1.2 六自由度机械臂的逆运动学分析

不失一般性, 研究如图 1 所示的六自由度机械臂, 该机械臂各关节均为旋转关节. 由于本研究讨论的 Biased-RRT 修正算法拟在关节空间下对系统进行路径规划, 故在获取机械臂末端位置信息后, 需将笛卡尔空间下的机械臂末端位置转换成机械臂关节空间下的 6 个关节转角位置.

设机械臂末端坐标系 $O_e - x_e y_e z_e$ 在机械臂基坐标系 $O - xyz$ 的位置为 $\mathbf{P} = (x, y, z)^T$, 用 $z - y - z$ 欧拉角表示的姿态为 $\mathbf{Q} = (\gamma, \beta, \alpha)^T$, 即以基坐标系 $O - xyz$ 为起始坐标系, 先绕基坐标系 $O - xyz$ 的 z 轴旋转 α 角, 获得坐标系 $O' - x' y' z'$; 再绕旋转后的坐标系 $O' - x' y' z'$ 的 y' 轴旋转 β 角, 获得坐标系

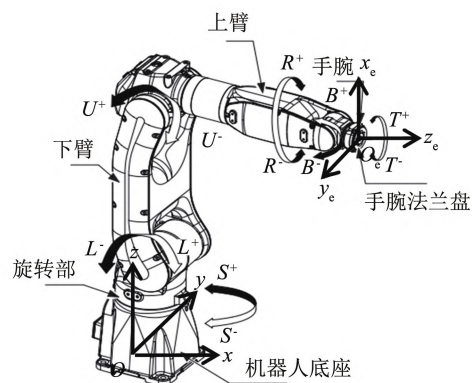


图1 六自由度机械臂
Fig.1 6-DOF manipulator

$O'' - x''y''z''$; 最后绕坐标系 $O'' - x''y''z''$ 的 x'' 轴旋转 γ 角, 最终确定出机械臂末端操作器的连体坐标系 $O_e - x_e y_e z_e$. 于是, 机械臂末端坐标系 $O_e - x_e y_e z_e$ 相对于基坐标系 $O - xyz$ 的齐次变换矩阵^[20]可表示为:

$${}^0_6T = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} = {}^0_1T(\theta_1) {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T(\theta_5) {}^5_6T(\theta_6) \quad (1)$$

其中: \mathbf{R} 为以 α 、 β 、 γ 表示的机械臂末端坐标系 $O_e - x_e y_e z_e$ 相对于机械臂基坐标系 $O - xyz$ 的旋转变换矩阵; $\mathbf{0}$ 为零向量; ${}^{i-1}_iT(\theta_i)$ ($i = 1, 2, \dots, 6$) 为机械臂连杆 i 坐标系相对于连杆 $i - 1$ 坐标系的齐次变换矩阵; θ_i 表示机械臂连杆 i 相对于连杆 $i - 1$ 的关节转角.

利用反变换法解耦式(1), 可依次算出机械臂各关节角 θ_i ($i = 1, 2, \dots, 6$). 为便于后续路径规划算法的设计, 令符号 $\mathbf{q}_* = (\theta_1^*, \theta_2^*, \dots, \theta_6^*)^T$ 表示算法搜索树 T 在关节空间下扩展的节点, 其中 θ_i^* ($i = 1, 2, \dots, 6$) 表示为在节点位置 $*$ 处机械臂第 i ($i = 1, 2, \dots, 6$) 个关节的关节角度值.

1.3 采样规则的目标导向设计

传统 Biased-RRT 算法使随机采样有小概率采样到目标点 \mathbf{q}_{goal} , 但为了让搜索树尽可能向目标点 \mathbf{q}_{goal} 扩展、提高搜索效率, 本修正算法拟对采样规则进行目标导向设计. 该设计涵盖 3 个部分: 1) 若算法采样时满足目标偏置条件, 即目标概率 P_{goal} 小于随机数 rand 时, 则将目标点 \mathbf{q}_{goal} 直接作为采样点; 2) 若算法采样时不满足目标偏置条件, 即目标概率 P_{goal} 大于随机数 rand 时, 则在整个工作空间随机采样 10 次, 依据曼哈顿距离, 从中挑选出距离目标点 \mathbf{q}_{goal} 最近的采样点 \mathbf{q}_{min} ; 3) 若搜索树扩展中出现被障碍物阻挡, 导致无法继续扩展的情况, 则算法在搜索树扩展被障碍物阻挡时, 在工作空间中进行单次随机采样, 增加扩展方向多样性, 继而让搜索树可以向其他方向扩展, 以便顺利离开障碍物区域. 本设计利用目标位置为搜索树扩展增加引导信息, 减少搜索树对无效区域的扩展, 将扩展范围缩小至目标点附近, 从而提高了算法的搜索效率.

于是, 基于目标导向设计后的采样规则为

$$\mathbf{q}_{\text{rand}} = \begin{cases} \mathbf{q}_{\text{goal}} & (P_{\text{goal}} < \text{rand}) \\ \mathbf{q}_{\text{min}} & (P_{\text{goal}} \geq \text{rand}) \\ \mathbf{q}_{\text{rand}} & (\text{collision}) \end{cases} \quad (2)$$

其中: P_{goal} 为预设的目标偏置概率; rand 为在 $[0, 1]$ 内生成的随机数; \mathbf{q}_{min} 表示距离目标点 \mathbf{q}_{goal} 最近的随机采样点; \mathbf{q}_{rand} 为选取的随机采样点; collision 表示搜索树扩展时遇到障碍物的情况.

1.4 基于变步长的新节点扩展方式

鉴于随机采样是在系统关节空间下进行的, 为便于后续规划算法的设计及路径代价度量, 将机械臂从节点 \mathbf{q}_A 运动至节点 \mathbf{q}_B 的曼哈顿距离

$$\text{Len}(\mathbf{q}_A, \mathbf{q}_B) = \sum_{i=1}^6 |\mathbf{q}_B(i) - \mathbf{q}_A(i)| \quad (3)$$

定义为节点 \mathbf{q}_A 与 \mathbf{q}_B 之间的运动路径代价.

图 2 为 Biased-RRT 修正算法所采用的扩展步长选取策略示意图. 一般来说, 传统 RRT 算法采样时常以固定步长来扩展搜索树的新节点 \mathbf{q}_{new} , 大步长虽可加快搜索速度, 但若步长选取过大, 有可能会使搜索树临近目标点时, 新节点 \mathbf{q}_{new} 在目标点 \mathbf{q}_{goal} 附近来回振荡, 容易陷入局部死循环, 无法准确扩展到目标点 \mathbf{q}_{goal} ; 但若步长选取过小, 则搜索树扩展缓慢、搜索效率低下. 为此, Biased-RRT 修正算法在搜索树扩展初期, 选取较大的步长 st_1 , 加快搜索树的探索; 为了解决较大步长的缺陷, 以 R_0 为半径, 目标点 \mathbf{q}_{goal} 为中心生成目标区域, 在新节点 \mathbf{q}_{new} 生成之前, 需要在搜索树上搜寻距离随机采样点 \mathbf{q}_{rand} 路径代价最小的节点 $\mathbf{q}_{\text{nearest}}$, 之后判断节点 $\mathbf{q}_{\text{nearest}}$ 与目标点 \mathbf{q}_{goal} 的曼哈顿距离 $\text{Len}(\mathbf{q}_{\text{nearest}}, \mathbf{q}_{\text{goal}})$ 是否小

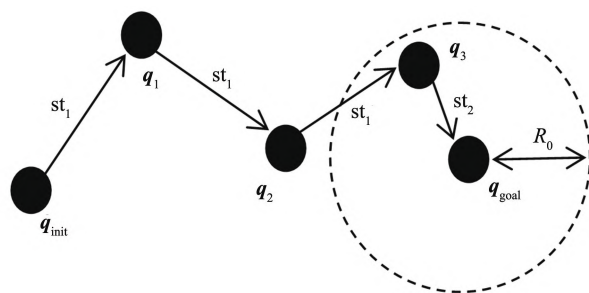


图 2 步长选取策略

Fig.2 Strategy of the step selection

于阈值 R_0 ? 若 $q_{nearest}$ 在目标区域内, 则将步长调整为较小的步长 st_2 .

扩展步长选取策略为

$$st = \begin{cases} st_1 & (\text{Len}(q_{nearest}, q_{goal}) > R_0) \\ st_2 & (\text{Len}(q_{nearest}, q_{goal}) \leq R_0) \end{cases} \quad (4)$$

其中: st 表示算法新节点扩展所选取的步长, st_1 为大步长, st_2 为小步长; R_0 表示目标区域半径.

1.5 基于路径代价最小的父节点重选

传统 Biased-RRT 算法在整个空间中搜索较为盲目, 所得到的路径质量往往较差. RRT* 算法虽可通过在路径节点的扩展中依据路径代价来重选父节点和重新规划路径, 使算法具有渐进最优, 但由于需要重新规划路径, 该算法的实时性受到较大影响. 为此, 本研究仅将 RRT* 算法中的重选父节点操作融入 Biased-RRT 修正算法中, 以实现路径优化. 具体做法如下:

在搜索树生成新节点 q_{new} 后, 将以 q_{new} 为圆心, 以 r 为半径, r 随着搜索树 T 扩展选取不同的数值, 具体设置为 $r = k_{rrg_} \times \log(\text{cardDbl})$; $k_{rrg_} = e + e/\text{dimDbl}$, $\text{cardDbl} = n_0 + 1$; n_0 表示当前搜索树 T 扩展的总节点数, e 是自然对数的底, dimDbl 是关节空间维度. 然后寻找当前搜索树上距离 q_{new} 小于 r 的所有节点, 记 $q_{nears}[i_0]$ 为满足条件的第 i_0 个节点, 并计算从起始点 q_{init} 途经 $q_{nears}[i_0]$ 并运动到 q_{new} 的总路径代价, 选取总代价最小且与 q_{new} 之间的运动不发生碰撞的节点替换掉 q_{new} 原本的父节点 $q_{nearest}$, 删除原本的 q_{new} 和 $q_{nearest}$ 的路径. 在选取出路径代价最小的父节点以后, 搜索树进入下一轮采样, 继续扩展. 重选父节点的简单演示过程如图 3 和图 4 所示, 其中带有圆圈数字表示搜索树扩展的各节点, 相邻两节点连线上的数值则表示为两点之间的运动路径代价值.

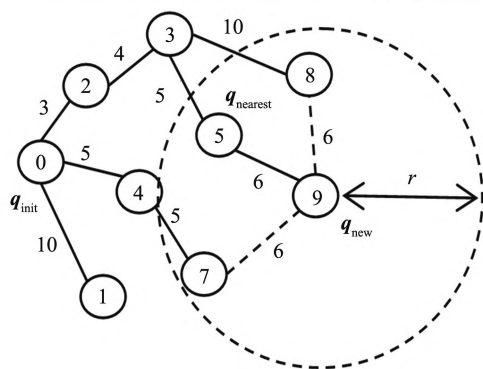


图3 初始扩展节点

Fig.3 Initial expansion nodes

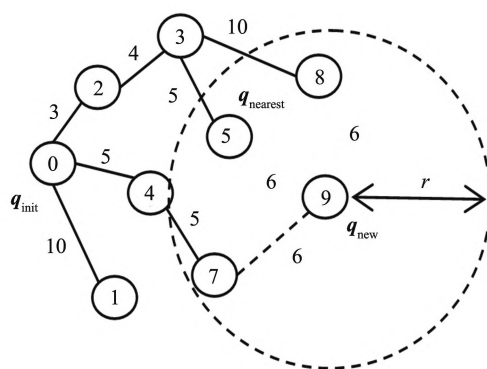


图4 重选父节点

Fig.4 Reselection of parent nodes

于是, 从起始点 q_{init} 途经 $q_{nears}[i_0]$ 并运动到 q_{new} 的路径代价可表示为

$$q_{new}[i_0].cost = q_{nears}[i_0].cost + \text{Len}(q_{nears}[i_0], q_{new}) \quad (5)$$

其中: $q_{nears}[i_0].cost = \sum_{j_0=0}^{j_0-2} \text{Len}(q_{j_0}, q_{j_0+1})$ 表示机械臂从起始点 q_{init} 运动到 $q_{nears}[i_0]$ 的路径代价; j_0 为从起始点 q_{init} 运动至 $q_{nears}[i_0]$ 所经过的总节点数; $q_{*}.cost$ 表示从起始点 q_{init} 运动至当前节点 q_{*} 的路径代价.

1.6 路径简化

为简化路径点数量、缩短路径长度, 本节拟对 Biased-RRT 修正算法规划出来的路径进行多余节点、拐点的剔除处理. 图 5 为路径简化的简单示意图, 其中黑色方块表示障碍物, 虚线表示初期规划出的路径, 黑色表示简化后的路径(即原路径 $q_{init} \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_{goal}$ 被简化为 $q_{init} \rightarrow q_1 \rightarrow q_4 \rightarrow q_{goal}$). 将 Biased-RRT 修正算法规划出来的所有离散节点的集合记为点集 $\{N\}$, 从起始点 q_{init} 开始, 依次

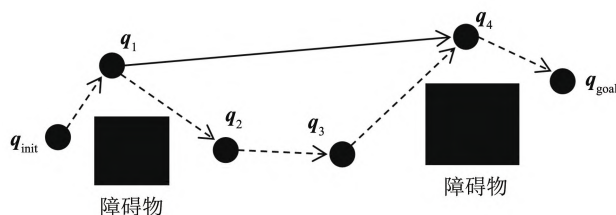


图5 路径简化示意图

Fig.5 Schematic diagram of the simplified path

遍历点集 $\{N\}$ 中的节点 q_L . 若点集 $\{N\}$ 有节点 q_L 与起始点 q_{init} 之间的连线, 没有碰撞发生且可以直达, 则直接连接该节点 q_L 和起始点 q_{init} , 删除 q_{init} 和 q_L 间的路径点, 同时将 q_{init} 和 q_L 保存至新的点集 $\{P_0\}$ 中; 若发生碰撞, 起始点 q_{init} 则与该节点的父节点 q_{L-1} 连接, 将 q_{init} 和 q_{L-1} 保存至点集 $\{P_0\}$ 中, 然后从 q_{L-1} 开始继续往后遍历节点. 重复以上过程, 直至遍历到目标点 q_{goal} , 最终所得点集 $\{P_0\}$ 即为简化后的新路径点的集合.

1.7 路径规划的实现步骤

路径规划的实现具体可分为以下 12 个步骤.

步骤 1 设 q_{init} 为初始起点, q_{goal} 为目标终点; 以终点位置 q_{goal} 为圆心、 R_0 为半径, 划定目标区域 C , 节点初始扩展大步长为 st_1 .

步骤 2 将 q_{init} 作为搜索树 T 的树根, 当搜索树 T 扩展到 q_{goal} 时, 结束搜索树 T 扩展.

步骤 3 当随机数 $rand$ 小于预先设定的目标偏置概率 P_{goal} 时, 进入步骤 4; 反之, 算法跳到步骤 5.

步骤 4 将随机采样点 q_{rand} 直接设定为终点 q_{goal} , 进入步骤 6.

步骤 5 在工作空间中随机采样 10 次, 选取与终点 q_{goal} 距离最小的点作为 q_{rand} .

步骤 6 在搜索树 T 上搜寻一个距离 q_{rand} 最小的节点 $q_{nearest}$.

步骤 7 计算 $q_{nearest}$ 与 q_{goal} 之间的距离 $Len(q_{nearest}, q_{goal})$, 若 $Len(q_{nearest}, q_{goal}) \leq R_0$, 则 $q_{nearest}$ 进入目标区域 C 范围, 步长选用小步长 st_2 ; 若 $Len(q_{nearest}, q_{goal}) > R_0$, 则仍选用初始的大步长 st_1 .

步骤 8 若 $Len(q_{nearest}, q_{rand}) \geq st$, 依据步长在 $q_{nearest}$ 和 q_{rand} 之间的连线上扩展新的节点 q_{new} ; 若 $Len(q_{nearest}, q_{rand}) < st$, 算法返回执行步骤 3.

步骤 9 在 q_{new} 附近半径 r 的范围内, 寻找 q_{new} 的所有邻近节点 $q_{nears}[i_0]$, 作为用来替换 q_{new} 的父节点 $q_{nearest}$ 的备选; 判断 q_{new} 和邻近节点 $q_{nears}[i_0]$ 之间是否有障碍物? 若有, 则删除这个邻近节点; 若没有, 计算从起始点 q_{init} 途经 $q_{nears}[i_0]$ 并运动到 q_{new} 的路径代价, 选取代价最小的邻近节点作为 q_{new} 新的父节点.

步骤 10 循环以上步骤, 直至搜索树 T 扩展至终点 q_{goal} .

步骤 11 从 q_{goal} 开始, 依据节点扩展顺序, 找到从 q_{init} 到 q_{goal} 的所有路径点, 得出规划的路径.

步骤 12 从起点 q_{init} 依次遍历路径点, 删除多余节点, 简化路径.

2 关节路径曲线的生成

Biased-RRT 修正算法是在机械臂的关节空间中规划, 因此所规划出的路径节点结构为 $q = (\theta_1, \theta_2, \dots, \theta_6)^T$. 利用 3 次样条插值技术对路径点进行插值, 以便为机械臂各关节分别生成一条光滑、连续的路径曲线. 下面将对机械臂第 $i(i = 1, 2, \dots, 6)$ 个关节路径曲线生成进行分析.

设 Biased-RRT 修正算法规划出的路径点有 $n + 1$ 个, 则各关节在关节空间下各有 $n + 1$ 个路径节点; 若定义各关节运行到各自规划的关节路径点所需时间为 $t_k(k = 0, 1, \dots, n)$, 则第 $i(i = 1, 2, \dots, 6)$ 个关节的所有路径点可表示为: $(t_0, \theta_{i0}), (t_1, \theta_{i1}), \dots, (t_{n-1}, \theta_{i(n-1)}), (t_n, \theta_{in})$, 其中: $\theta_{ik}(i = 1, 2, \dots, 6; k = 0, 1, \dots, n)$ 表示机械臂第 i 个关节在时间 t_k 下的实际位置.

对机械臂第 $i(i = 1, 2, \dots, 6)$ 个关节两相邻路径点 (t_j, θ_{ij}) 和 $(t_{j+1}, \theta_{i(j+1)})$ 进行如下 3 次样条插值, 即

$$S_{ij}(t) = a_{ij} + b_{ij}(t - t_j) + c_{ij}(t - t_j)^2 + d_{ij}(t - t_j)^3 \quad (6)$$

$$S'_{ij}(t) = b_{ij} + 2c_{ij}(t - t_j) + 3d_{ij}(t - t_j)^2 \quad (7)$$

$$S''_{ij}(t) = 2c_{ij} + 6d_{ij}(t - t_j) \quad (8)$$

其中: $S_{ij}(t)$ 表示第 $i(i = 1, 2, \dots, 6)$ 个关节第 $j(j = 0, 1, \dots, n - 1)$ 段的 3 次样条曲线; $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ 为第 i 个关节第 j 段样条的待求多项式系数; $S'_{ij}(t)$ 和 $S''_{ij}(t)$ 分别表示 $S_{ij}(t)$ 的一阶及二阶导数.

为了保证生成轨迹的光滑、连续性及各关节运行的平稳性, 前述各段插值曲线需要满足边界条件 ($S_{ij}(t_j) = \theta_{ij}, S_{ij}(t_{j+1}) = S_{i(j+1)}(t_{j+1}) = \theta_{i(j+1)}$)、连续性条件 ($S'_{ij}(t_{j+1}) = S'_{i(j+1)}(t_{j+1}), S''_{ij}(t_{j+1}) = S''_{i(j+1)}(t_{j+1})$) 及关节运动速度起止条件 ($S'_{i0}(t_0) = 0, S'_{i(n-1)}(t_n) = 0$). 将上述条件代入式 (6) ~ (8) 中, 经一系列整理后可得 a_{ij}, b_{ij}, c_{ij} 和 d_{ij} ^[21]. 最后, 机械臂第 $i(i = 1, 2, \dots, 6)$ 个关节的总路径曲线 θ_i 将由前述分段曲线 $S_{i0}, S_{i1}, \dots, S_{i(n-1)}$ 组成.

3 仿真实验与结果分析

3.1 仿真实验概述

本研究首先对传统 RRT 算法、传统 Biased-RRT 算法及 Biased-RRT 修正算法进行二维仿真规划实验,通过不同算法在平均扩展节点数目、路径质量、规划时间上的结果对比,来证实 Biased-RRT 修正算法的有效性。其次,在 ROS 平台上将 Biased-RRT 修正算法应用于复杂环境下六自由度机械臂的路径规划实验,以进一步检验 Biased-RRT 修正算法的可行性。

3.2 改进算法的有效性验证实验

不失一般性,将系统位置节点 q_s 简化为二维向量,先行实施改进算法在有复杂障碍物的二维环境下的路径规划实验。仿真环境所采用的地图大小为 $800 \text{ px} \times 800 \text{ px}$,且设置有大小不等、形状不同的障碍物,仿真要求算法在起始点 $q_{\text{init}} = (10, 10)^T$ 到目标点 $q_{\text{goal}} = (720, 750)^T$ 之间快速规划出一条无障、便捷的运动路径。仿真先后采用传统 RRT 算法、Biased-RRT 算法及 Biased-RRT 修正算法对系统路径分别进行规划。仿真时,传统 RRT 算法和 Biased-RRT 算法均采用定步长 40, Biased-RRT 修正算法采用变步长方式,即初始大步长 st_1 设置为 50,目标区域半径 R_0 为 20,进入目标区域后的小步长 st_2 设置为 10;传统 Biased-RRT 算法和 Biased-RRT 修正算法的目标偏置概率均设置为 0.1。图 6~8 分别描述了传统 RRT 算法、Biased-RRT 算法及 Biased-RRT 修正算法所生成的搜索树及其规划出来的路径。在各图中,红色点表示起始点,绿色点表示目标点,空心蓝点为搜索树扩展的节点;蓝色实线表示算法初步规划出的路径,图 8 中红色实线表示经过 Biased-RRT 修正算法简化后的路径。

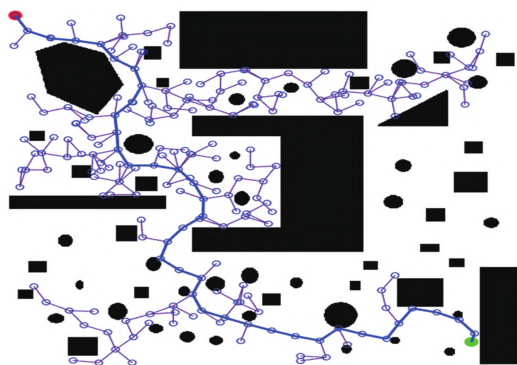


图 6 传统 RRT 算法仿真结果

Fig.6 Simulation results of traditional RRT algorithm

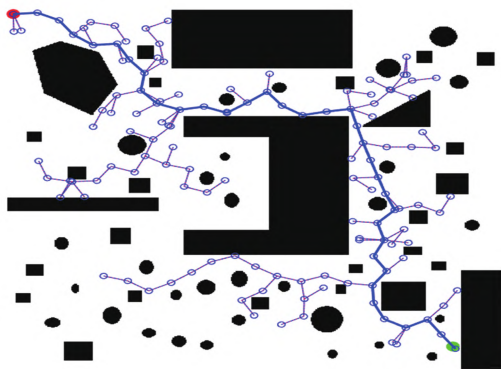


图 7 传统 Biased-RRT 算法仿真结果

Fig.7 Simulation results of traditional Biased-RRT algorithm

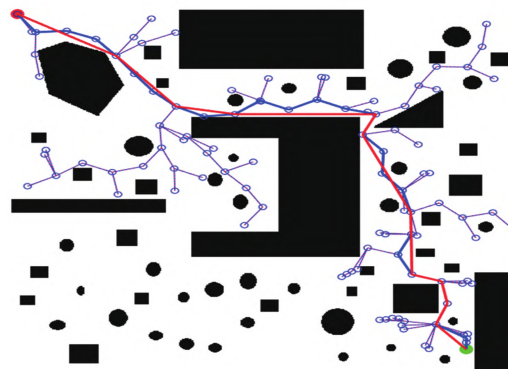


图 8 Biased-RRT 修正算法仿真结果

Fig.8 Simulation results of Biased-RRT correction algorithm

对于路径规划问题,需要考虑算法对计算资源的消耗,以便算法可以应用于更多的环境中。传统 RRT 算法、Biased-RRT 算法及 Biased-RRT 修正算法都是以树结构探索空间,采用节点的方式记录路径点位,依据扩展顺序来生成路径。3 种算法在相同环境下算法运行 30 次后的平均扩展节点数如表 1 所示。表 1 中 Biased-RRT 修正算法采用路径简化前的路径节点数目来对比。算法搜索树中记录的节点越多,需要的

表 1 平均扩展节点数对比

Tab.1 Comparison of the average number of expanded nodes

算法	扩展节点数	路径节点数目	未使用节点占比/%
RRT	345	40	88.41
Biased-RRT	167	39	76.65
Biased-RRT 修正算法	92	32	65.21

计算资源越多, 路径所使用的节点数目越少, 意味着算法计算资源有效使用率越低, 因此可用多余的节点占比来衡量算法的额外开销。

从表 1 可知, 在同等环境下, 传统 RRT 算法是对整个空间随机探索, 即使是对 RRT 算法改进后的 Biased-RRT 也是小概率地向目标区域扩展, 从而导致算法产生许多额外开销。相比之下, 所提 Biased-RRT 修正算法搜索树节点数目较少, 节点利用率较高, 可有效减少算法的额外开销。这是因为 Biased-RRT 修正算法并不在整个空间随机采样, 而是利用目标位置为指引, 通过多次采样来选取靠近目标点的采样点, 减少算法对不必要空间的探索, 让搜索树生成趋向性明显, 避免对无效方向的扩展生成。其次, 相比固定步长扩展而言, 变步长的扩展方式让搜索树可以用更少的节点和扩展次数来快速通过前期区域, 从而减少了搜索树节点数。

传统 RRT 算法和 Biased-RRT 算法都不具备优化路径的能力, 算法规划出的路径冗余节点多, 路径质量往往不是最优的, 会增加机械臂的运行时间, 因此需要对路径质量优化。一般以平均路径节点数和平均路径代价来考量算法规划出的路径质量。3 种算法在同等环境下所规划的路径质量如表 2 所示, 表 2 中 Biased-RRT 修正算法以路径简化后的节点数进行对比。

表 2 路径质量对比

Tab.2 Comparison of path quality

算法	平均路径 节点数	平均路径 代价
RRT	40	1 539
Biased-RRT	39	1 501
Biased-RRT 修正算法	11	1 375

由表 2 可知, Biased-RRT 修正算法对路径有一定的优化能力, Biased-RRT 修正算法平均节点数较之传统 RRT 和 Biased-RRT 算法减少约 72.5%, 路径缩短约 10.65%。传统 RRT 算法和 Biased-RRT 算法在搜索树扩展时, 会向无效区域扩展, 所得的路径并非是最优路径。为了尽可能优化路径质量, Biased-RRT 修正算法引入 RRT* 的重选父节点操作, 在每次扩展节点的同时, 不依赖扩展的先后顺序确定父节点, 而是拣选出路径代价最小的邻近节点作为父节点, 修正节点扩展路径。最后对规划出的路径简化处理, 剔除冗余节点和拐点, 减少不必要的曲折路径, 从而减少了路径节点和代价。

在机械臂的实际应用中, 对路径规划时间有着较高的要求, 算法规划时间越短, 机械臂运行效率越高, 更能满足机械臂应用要求。

从表 3 可得, Biased-RRT 修正算法相比 RRT 算法提升近 74.71%, 比之 Biased-RRT 算法, 提高约 44.67%, 表明 Biased-RRT 修正算法搜索树扩展目的性明显, 算法的实时性更高。这是因为 Biased-RRT 修正算法里基于采样规则的目标导向设计充分利用了目标点作为引导信息, 可确保搜索树扩展方向不会过多偏离目标点, 在一定程度上避免了搜索树无效扩展, 从而加快算法搜索效率。

表 3 平均规划时间对比

Tab.3 Comparison of average planning time

算法	\bar{t}/s
RRT	38.505
Biased-RRT	17.597
Biased-RRT 修正算法	9.735

3.3 基于 ROS 实验平台的机器人仿真实验验证

为验证所提 Biased-RRT 修正算法在六自由度机械臂路径规划上可行性, 在 ROS 平台上, 选用 MOTO-MAN-GP7 型六自由度机械臂作为仿真机械臂, 机械臂的 D-H 参数如表 4 所示。

表 4 MOTOMAN-GP7 的 D-H 参数

Tab.4 D-H parameters of MOTOMAN-GP7

关节 i	$\alpha_{i-1}/(^{\circ})$	a_{i-1}/mm	d_i/mm	$\theta_i/(^{\circ})$	关节可动范围/ $(^{\circ})$
1	0	0	330	0	-170~+170
2	-90	40	0	-90	-70~+190
3	180	445	0	0	-116~+255
4	-90	40	-440	0	-190~+190
5	90	0	0	-90	-135~+135
6	-90	0	-80	0	-360~+360

基于 ROS 平台对该机械臂进行三维建模,并生成如图 9(a)所示的存在有复杂障碍物的仿真环境.图中,仿真空间尺寸设置为 $2\text{ m} \times 2\text{ m} \times 2\text{ m}$;障碍物选取为圆柱体,图中数字表示障碍物的序号;以机械臂基座中心构建坐标系,障碍物 1 的位置为 $(0\text{ m}, -0.4\text{ m}, 0.3\text{ m})$ 、高度 0.6 m 、半径 0.2 m ;障碍物 2 的位置为 $(0.5\text{ m}, 0.5\text{ m}, 0.2\text{ m})$ 、高度 0.4 m 、半径 0.2 m ;障碍物 3 的位置为 $(0.8\text{ m}, 0\text{ m}, 0.35\text{ m})$ 、高度 0.3 m 、半径 0.1 m ;障碍物 4 的位置为 $(0.8\text{ m}, 0\text{ m}, 0.1\text{ m})$ 、高度 0.2 m 、半径 0.2 m ;障碍物 5 的位置为 $(0.8\text{ m}, -0.5\text{ m}, 0.1\text{ m})$ 、高度 0.2 m 、半径 0.2 m ;机械臂末端起始位姿为 $\mathbf{P}_1 = (0.55\text{ m}, 0\text{ m}, 0.81\text{ m})^T$ 、 $\mathbf{Q}_1 = (\pi/2(\text{rad}), -\pi/2(\text{rad}), \pi/2(\text{rad}))^T$;所提改进 Biased-RRT 修正算法规划机械臂从末端起始位姿运动到第 1 个指定位姿 $\mathbf{P}_2 = (0.8\text{ m}, -0.3\text{ m}, 0.35\text{ m})^T$ 、 $\mathbf{Q}_2 = (\pi(\text{rad}), 0(\text{rad}), \pi(\text{rad}))^T$,紧接着再运动到第 2 个指定位姿 $\mathbf{P}_3 = (0.8\text{ m}, 0.3\text{ m}, 0.4\text{ m})^T$ 、 $\mathbf{Q}_3 = (\pi(\text{rad}), 0(\text{rad}), \pi(\text{rad}))^T$,最后返回到起始位姿的具体路径;仿真中,选用 ROS 平台中的 KDL 逆运动学求解器将机械臂末端位姿转化为机械臂各关节位置,并利用 3 次样条插值技术生成机械臂各关节路径曲线,进而引导机械臂仿真模型进行避障运动.仿真实验中算法的目标偏置概率设为 0.05,搜索树扩展的大步长 st_1 设置为 7.614,小步长 st_2 设置为 3.807,目标区域半径 R_0 设置为 7.614.

图 9 为机械臂整个仿真运动过程示意图,图中的橙色机械臂表示起始状态,蓝色机械臂表示运动状态;图 9(b)表示以机械臂基座为参考坐标系,机械臂末端运动至第一个位置运动的过程;图 9(c)表示机械臂运动至第二个指定位置;图 9(d)表示机械臂返回起始位置的过程.

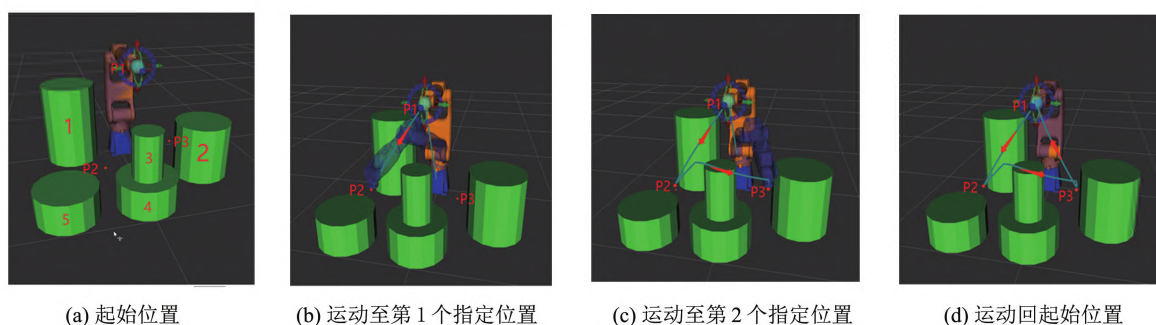


图 9 机械臂运动过程

Fig.9 Motion of the manipulator

从上述仿真结果可以观察到:所提 Biased-RRT 修正算法可使六自由度机械臂在复杂障碍物环境中有效地避开障碍物,依次到达指定位置,并完成预期运动.图 10 和图 11 分别给出机械臂从起始位置平稳运动到第一个指定位置时,机械臂各关节角的运动及其速度变化曲线.

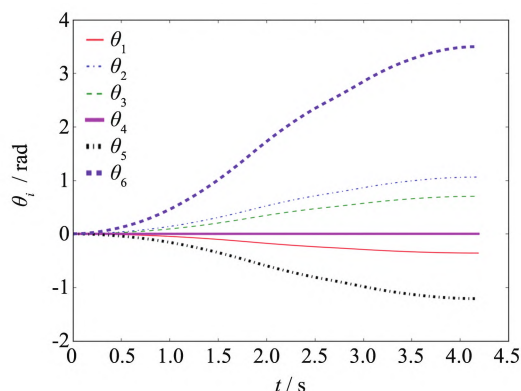


图 10 机械臂各关节运动曲线

Fig.10 Motion curves of the manipulator's joints

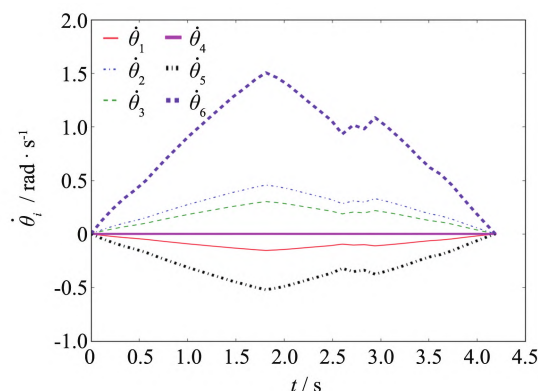


图 11 机械臂各关节速度曲线

Fig.11 Velocity curves of the manipulator's joints

4 结语

设计一种六自由度机械臂路径规划的 Biased-RRT 修正算法. 相比于传统 RRT、Biased-RRT 算法, 所提 Biased-RRT 修正算法利用目标偏置、多次随机采样取“最近”和单次随机采样相结合的多重采样方式来生成节点, 以减少搜索时长、提高搜索效率; 其间, 所采用的变步长节点扩展方式将有助于修正算法生成的搜索树快速、稳定地搜索到目标节点. 此外, 还通过引入基于路径代价最小的父节点重选、多余路径节点剔除操作, 来优化规划出来的路径, 并结合 3 次样条插值技术来生成机械臂各关节光滑、连续的路径曲线. 二维及三维情形下的仿真结果表明: 所提 Biased-RRT 修正算法在机械臂路径规划上是可行、有效的.

参考文献:

- [1] YUAN Q, HAN C S. Research on robot path planning based on smooth a algorithm for different grid scale obstacle environment[J]. Journal of Computational & Theoretical Nanoscience, 2016, 13(8): 5312-5321.
- [2] TANG B J, HIROTA K R, WU X D, *et al.* Path planning based on improved hybrid A* algorithm[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2021, 25(1): 64-72.
- [3] XIONG X Y, MIN H T, YU Y B, *et al.* Application improvement of A* algorithm in intelligent vehicle trajectory planning[J]. Mathematical Biosciences and Engineering, 2021, 18(1): 1-21.
- [4] ROSTAMI S M H, SANGAIAH A K, WANG J, *et al.* Obstacle avoidance of mobile robots using modified artificial potential field algorithm[J]. EURASIP Journal on Wireless Communications and Networking, 2019, 2019(1): 1-19.
- [5] BADAWY A. Dual-well potential field function for articulated manipulator trajectory planning[J]. Alexandria Engineering Journal, 2016, 55(2): 1235-1241.
- [6] 胡永仕, 张阳. 基于遗传模糊算法的智能车辆避障路径规划研究[J]. 福州大学学报(自然科学版), 2015, 43(2): 219-224.
- [7] LIU J H, YANG J G, LIU H P, *et al.* An improved ant colony algorithm for robot path planning[J]. Soft Computing, 2017, 21(19): 5829-5839.
- [8] 李锋源, 许艳萍, 王武. 多策略蚁群算法求解机器人路径规划[J]. 福州大学学报(自然科学版), 2011, 39(3): 385-391.
- [9] 贾庆轩, 陈钢, 孙汉旭, 等. 基于 A* 算法的空间机械臂避障路径规划[J]. 机械工程学报, 2010, 46(13): 109-115.
- [10] KHATIB O. Real-time obstacle avoidance system for manipulators and mobile robots[J]. International Journal of Robotics Research, 1986, 5(1): 90-98.
- [11] 马培羽. 六自由度机械臂避障路径规划研究[D]. 广州: 华南理工大学, 2010.
- [12] LAVALLE S M. Rapidly-exploring random trees: a new tool for path planning[J]. Algorithmic & Computational Robotics New Directions, 1999, 1(1): 293-308.
- [13] 杨英浩, 刘晋浩, 郑一力, 等. 林木联合采育机械臂避障路径规划[J]. 林业科学, 2021, 57(2): 179-192.
- [14] ZHANG H J, WANG Y K, ZHENG J, *et al.* Path planning of industrial robot based on improved RRT algorithm in complex environments[J]. IEEE Access, 2018(6): 53296-53306.
- [15] 余敏, 罗建军, 王明明, 等. 一种改进 RRT* 结合四次样条的协调路径规划方法[J]. 力学学报, 2020, 52(4): 1024-1034.
- [16] 李洋, 徐达. 基于引力自适应步长 RRT 的双臂机器人协同路径规划[J]. 机器人, 2020, 42(5): 606-616.
- [17] LAVALLE S M. Randomized kinodynamic planning[J]. The International Journal of Robotics Research, 2001, 20(5): 378-400.
- [18] 周芳, 朱齐丹, 赵国良. 基于改进快速搜索随机树法的机械手路径优化[J]. 机械工程学报, 2011, 47(11): 30-35.
- [19] KARAMAN S. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.
- [20] 熊有伦, 丁汉, 刘恩沧, 等. 机器人学[M]. 北京: 机械工业出版社, 1993: 24-40.
- [21] 赵金龙, 晁永生, 袁逸萍. 基于 A* 算法和三次样条的工业机械臂路径平滑性研究[J]. 机械设计与研究, 2019, 35(1): 61-69.

(责任编辑: 沈芸)