

[研究·设计]

DOI:10.3969/j.issn.1005-2895.2022.01.007

基于自适应目标偏置系数的机械臂路径规划算法

吴文迎, 蔡锦达, 高朋帅

(上海理工大学 出版印刷与艺术设计学院, 上海 200093)

摘要:针对快速搜索随机树(RRT)算法因其随机特性导致的运行时间过长和计算结果往往不是最优解的问题,在RRT算法及其改进算法的基础上,课题组提出了一种基于自适应目标偏置系数的机械臂路径规划算法,并将该算法应用到医疗设备无影灯的使用中。首先,用D-H参数法对6自由度机械臂进行运动学描述和正运动学求解,得到末端的运动状态和位姿;然后,综合RRT算法及其改进算法的优点拓展新节点和重置父节点,并引入自适应目标偏置系数,再对路径作剪枝后处理。对比实验结果显示:该方法在保证算法搜索效率的基础上,有效地提高了路径质量。基于ROS的仿真平台验证了该方法具有实用性和可行性。

关键词:机械臂;路径规划;快速搜索随机树算法;D-H参数法;自适应目标偏置;剪枝后处理

中图分类号:TP241;TH165 **文献标志码:**A **文章编号:**1005-2895(2022)01-0034-09

Path Planning Algorithm for Manipulator Based on Adaptive Target Biasing Coefficient

WU Wenying, CAI Jinda, GAO Pengshuai

(College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In order to solve the problems that RRT algorithm has long running time and the calculation results are often not the optimal solution due to its random characteristics, a path planning algorithm for manipulator based on adaptive target bias coefficient was proposed based on RRT algorithm and its improved algorithm, and the algorithm was applied to the shadowless lamp of medical equipment. Firstly, the D-H parameter method was used to describe the kinematics and solve the forward kinematics of the 6-DOF manipulator, and the motion state and pose of the terminal were obtained. Then, the new node was expanded and the parent node was reset based on the advantages of RRT algorithm and its improved algorithm, the adaptive target bias coefficient was introduced and the path was pruned after processing. The comparative experimental results show that this method can effectively improve the path quality while ensuring the search efficiency of the algorithm. The practicability and feasibility of the proposed method are verified by a simulation platform based on ROS.

Keywords: manipulator; path planning; RRT (Rapid-Exploration Random Tree) algorithm; D-H parameter method; adaptive target offset; post-pruning

机械臂的路径规划问题一直是人工智能领域研究的热点。路径规划的目的是使机械臂在一定约束条件下,从起始状态到目标状态,在三维空间中搜索无碰撞路径进行操作^[1]。在对机械臂做路径规划的同时,要满足3个原则:①可行性,即路径规划必须能够实际用于真实机械臂,而不是“纸上谈兵”;②最优解,避障问

题是指在有障碍物的环境中,从起点到目标点规划一条不发生碰撞的最优路径^[2];③防碰撞,机械臂投入使用后维修检测都将会耗费人力财力,因此在做路径规划时应最大程度地避免机械臂运动中与障碍物的碰撞。

为了找出能够同时满足以上3个原则的路径规划

收稿日期:2021-06-17;修回日期:2021-11-25

第一作者简介:吴文迎(1998),女,江苏南京人,硕士研究生,主要研究方向为嵌入式技术。E-mail:594357820@qq.com

方法, LaValle^[3] 提出了快速搜索随机树 (rapid-exploration random tree, RRT) 算法, 该算法可以对高维空间快速搜索, 通过在随机采样点向空白区域搜索避开模拟障碍物从而高效地解决复杂高维空间的路径规划问题。由于 RRT 算法的实际应用效率较高, 越来越多的路径规划研究基于该算法, 于是涌现出了大量 RRT 的改进算法。Kuffner 等^[4] 提出了基于双向扩展平衡的连结型双树 RRT-connect 算法, 该算法在 RRT 算法的基础上同时发展 2 棵随机树, 分别从起始点和目标点生长, 并运用贪婪策略将 2 棵树连接起来, 以减少搜索时间。Sertac Karaman 等^[5] 将基于随机采样的路径规划算法与随机几何理论相结合, 提出了一种新的快速寻优随机图算法 (rapidly exploring random graph, RRG), 并将该算法的树型版本称为 RRT* 算法。

课题组在 RRT 算法的基础上, 综合 RRT-connect 和 RRT* 算法的优势, 提出了一种基于自适应目标偏置系数的机械臂路径规划算法, 该算法将自适应偏置系数添加到节点的拓展过程中, 从而减少了随机采样点的数目, 减少了算法运行时间。在路径树生成后, 加入了剪枝算法来进一步优化路径树。与 RRT 算法及其已有的改进算法相比, 该算法可以有效地提高搜索效率和路径质量。

1 机械臂的运动学分析

课题组将工作末端为无影灯的 6 自由度机械臂作为研究平台。建立 6 自由度机械臂 D-H 参数连杆坐标系如图 1 所示, 然后根据坐标系计算各关节的 D-H 参数。

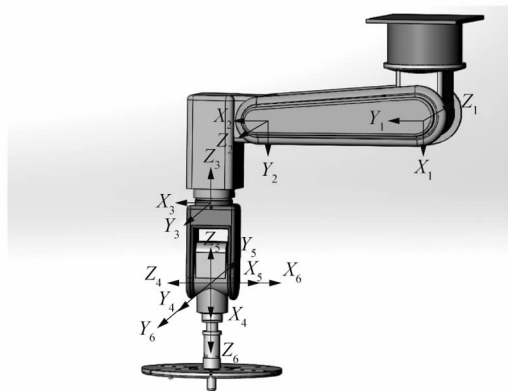


图 1 D-H 参数连杆坐标系

Figure 1 D-H parametric linkage coordinate system

由 D-H 参数坐标系得出各个关节的 D-H 参数如表 1 所示。

表 1 机械臂的 D-H 参数表

Table 1 D-H coordinate parameters of manipulator

关节 i	连杆转角 $\theta_i / (^\circ)$	连杆偏距 d_i / m	连杆偏距 a_i / m	扭角 $\alpha_i / (^\circ)$
1	θ_1	d_1	0	90
2	θ_2	0	a_2	0
3	θ_3	0	a_3	90
4	θ_4	d_4	0	90
5	θ_5	0	0	90
6	θ_6	0	0	0

由于课题组对机械臂所做的路径规划基于给定的初始点及目标点在坐标系中的关节坐标, 因此不对机械臂做逆运动学求解。

2 路径规划方法

RRT 算法及其改进算法是一种基于随机采样的查询步进式算法, 在高维空间和复杂约束下的路径规划应用中具备良好的使用性能^[6]。

2.1 基础 RRT 算法

RRT 算法即快速搜索随机树算法, 原理是随机发展出一棵路径树 T 。RRT 算法主要步骤如流程图 2 所示。首先初始化参数, 起始点 S 、目标点 G , 扩展步长 S_s 、偏差 b 等。

路径规划问题是对所有 $\tau \in [0, 1]$, 寻找一条从初始配置 $\sigma(0) = S$ 开始到达目标区域 $\sigma(1) \in G$ 且满足 $\sigma(\tau) \in x_{\text{rand}}$ 的无碰撞路径 $\sigma: [0, 1] \rightarrow x_{\text{rand}}$ ^[7]。如图 3 所示, 这棵路径树从起始点 S 开始生长, 在地图空间中随机采样取点, 寻找这棵路径树中与随机采样点 X_{rand} 最接近的点 X_{near} , 以 X_{near} 为初始点沿着 $X_{\text{near}} - X_{\text{rand}}$ 方向延长步长 Δ 得到新点 X_{new} , 且此点需要与随机采样点不碰撞障碍地连接起来, 于是对点 X_{new} 做碰撞检测, 即检验 X_{new} 和 X_{rand} 2 点之间的路径是否存在障碍。通过碰撞检测的新点加入路径树中。搜索过程中不断地在路径树中加入随机采样点, 生成随机路径树, 直到探索到目标点 G , 于是生成一条从起始点到目标点的唯一路径。

RRT 算法伪代码。

$T. \text{add}(S)$

for $i = 1$ to N_{set}

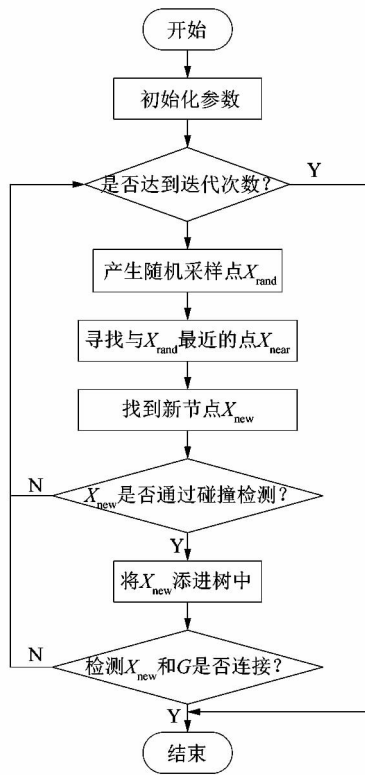


图2 RRT 算法流程图

Figure 2 RRT algorithm flowchart

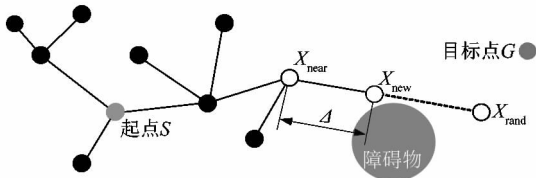


图3 RRT 算法路径规划的过程

Figure 3 RRT algorithm path planning process

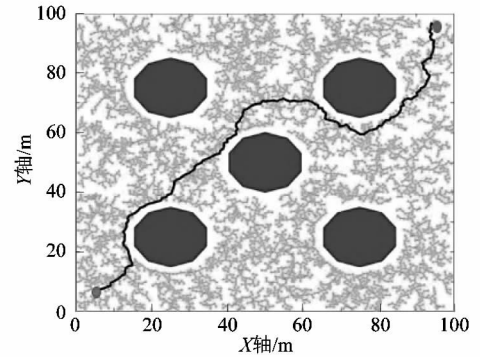
```

 $x_{rand} \leftarrow \text{Sample}()$ 
 $x_{near} \leftarrow \text{Nearest}(T, x_{rand})$ 
 $x_{new} \leftarrow \text{Extend}(x_{near}, x_{rand}, \text{StepSize})$ 
if  $\text{Collectionfree}(x_{new}, \text{world})$ 
     $T.add(x_{new})$ 
else continue
if  $\|x_{new} - G\| \leq \text{bri}$ 
    return  $T$ , flag = ture
else flag = false
    return  $T$ , flag

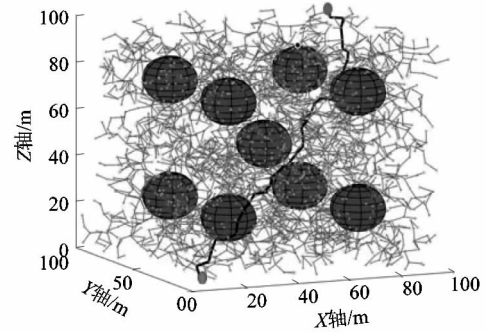
```

其中 $\text{Sample}()$ 函数在六维空间中完成随机点的采

样。 $\text{Extend}()$ 函数完成节点的扩展, $\text{Collectionfree}()$ 函数完成采样过程中的碰撞检测。



(a)二维障碍空间



(b)三维障碍空间

图4 RRT 算法路径规划结果

Figure 4 RRT algorithm path planning results

图4所示为RRT算法在二维障碍空间和三维障碍空间中的路径规划结果。RRT算法适用于多障碍物情况下6自由度机械臂的运动规划,但该算法虽然降低了路径规划的计算成本,却因为采样点的随机性导致收敛速度慢,路径生成时间长,且未对生成的路径做后处理,因此不是路径规划的最优解。

2.2 RRT-connect 算法

RRT算法从起始点发展一棵路径树到目标点,由于采样点的选择具有随机性,因此该算法的收敛速度较慢,路径生成效率不高。RRT-connect算法在RRT算法的基础上提出了双树的构想,即从起始点区域和目标点区域同时搜索状态空间通过随机采样点来各自发展一棵路径树。如图5所示,在每一次迭代中,发展一棵树的新点后,将此点作为另一棵树的目标点进行生长,2棵树不断地往对方的方向交替扩展,直到2棵树规划出的路径支点相连。不同于RRT算法在生成样本并插入到路径树中时扩展最大扩展长度,在没有

障碍物碰撞的情况下,路径树继续向目标方向扩展,因此可以更快地规划路径^[8]。

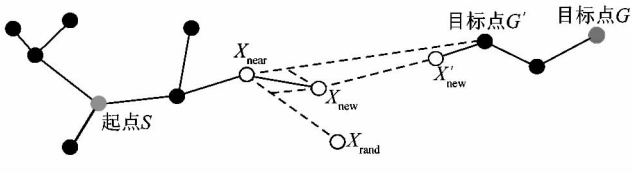


图5 RRT-connect 算法路径规划的过程

Figure 5 RRT-connect algorithm path planning process

RRT-connect 算法伪代码。

```

 $T_a T_b$ 
for  $i = 1$  to  $N_{set}$ 
 $x_{rand} \leftarrow \text{Sample}()$ 
 $x_{near} \leftarrow \text{Nearest}(T_a, x_{rand})$ 
 $x_{new} \leftarrow \text{Extend}(x_{near}, x_{rand}, \text{StepSize})$ 
 $T_a.add(x_{new})$ 
if  $\text{Getit}(x_{new})$ 
    flag = ture
    return( $T_a, T_b, \text{flag}$ )
else
    swap( $T_a, T_b$ )
flag = false
return(flag)

```

RRT-connect 算法路径规划结果如图6所示。

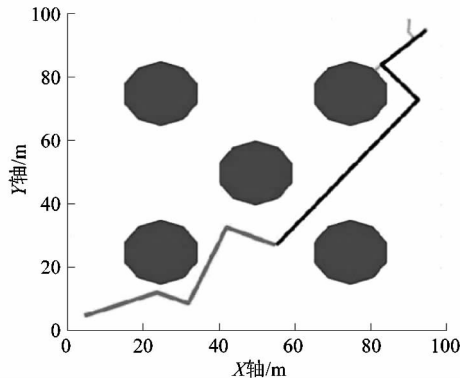


图6 应用贪婪策略的 RRT-connect 算法

Figure 6 RRT-Connect algorithm with greedy strategy

RRT-connect 算法在处理节点时,只要一棵路径树上生成的新节点与另一棵路径树的距离比指定的阈值

小,则连接2点,这容易导致路径的连接处转角过大,甚至出现 180° 的转角^[9]。较 RRT 算法而言,RRT-connect 算法提高了搜索速度和搜索效率,2 棵树的快速生长带有启发性且能够有效避免陷入局部最优,路径树生长的搜索过程更加贪婪和明确。

2.3 RRT* 算法

RRT 算法中,选取的随机采样点总是和距离该点最近的点连接,而不是最优的点。RRT* 算法针对这一问题在 RRT 算法的基础上做出改进,在随机采样点加入路径树后,RRT* 算法没有直接将最近邻节点作为其父节点,而是以该点周围半径为 r 的区域为目标区域寻找一个使其代价值最小的节点作为父节点^[10],即重置父节点操作。如图7所示,标准是加入路径树后能够使得该点到起点的距离更短。如果重置后的父节点可以减少路径代价,则将新的节点加入路径树连接起来,摒弃原来的连接线,即重布线操作。重布线操作对每一个新点的父节点选择增加一个优化,减少每一代的路径距离^[11]。

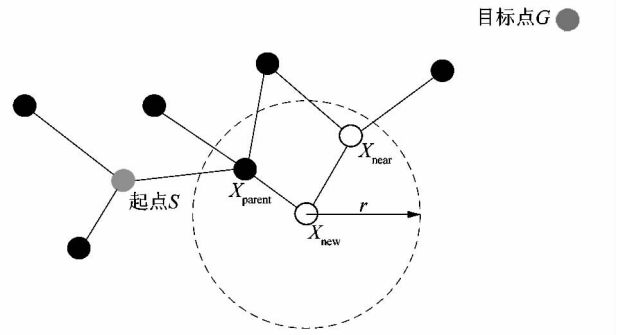


图7 重置父节点操作示意图

Figure 7 Resetting parent node operation schematic

RRT* 算法伪代码。

```

 $T$ 
for  $i = 1$  to  $N_{set}$ 
 $x_{rand} \leftarrow \text{Sample}()$ 
 $x_{near} \leftarrow \text{Nearest}(T, x_{rand})$ 
 $x_{new} \leftarrow \text{Extend}(x_{near}, x_{rand}, \text{StepSize})$ 
 $T \leftarrow x_{new}$ 
 $x_{near} \leftarrow \text{Findnear}(x_{new}, T, r)$ 
Chooseparent( $x_{new}, x_{near}$ )
Rewrite( $x_{new}, x_{near}$ )

```

```
if Getit( $x_{\text{new}}$ )
```

```
    flag = true
```

```
else
```

```
    flag = false
```

```
return  $T$ , flag
```

RRT* 需要大量的节点回溯来判断约束条件, 从而影响收敛速度, 难以应用于具有刚性实时性要求的系统^[12], 在通道大且障碍物多的地图环境中效率较低。

2.4 改进 RRT 算法

综合 RRT-connect 算法和 RRT* 算法的优点, 课题组提出了一种基于自适应目标偏置系数的改进 RRT 算法, 改进后的 RRT 算法基于随机采样的方式, 应用了双树的构想, 不应用贪婪算法。在起始点区域和目标点区域各自生长一棵路径树, 将对方作为交替生长的方向, 当双树连接后即终止对地图空间的搜索, 将 2 棵路径树整合成为 1 颗从起始点到目标点的路径树, 最后对合成的路径树做剪枝后处理。

2.4.1 自适应偏置算法

RRT 算法具有随机特性, 因此路径树的生长缺乏导向性, 在路径树的生长过程中设置 1 个偏置概率参数可以使得随机采样点更偏向于目标点^[13]。

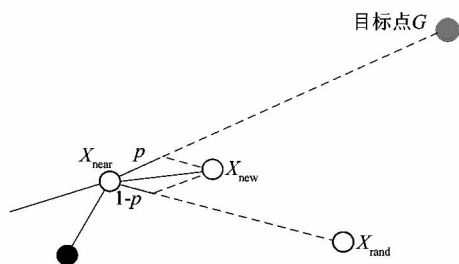


图8 应用目标偏置系数的扩展方法

Figure 8 Extended method of applying target bias coefficient

如图8所示, 设置的目标偏置系数 P 决定了新节点 X_{new} 偏向随机采样点 X_{rand} 还是目标点 G , 当 P 的值越大说明新节点 X_{new} 越偏向目标点 G , 当 P 的值越小说明新节点 X_{new} 越偏向随机采样点 X_{rand} 。且有

$$X_{\text{near}} = f(P) = X_{\text{near}} + \left[P \times \frac{G - X_{\text{near}}}{|G - X_{\text{near}}|} + (1 - P) \times \right.$$

$$\left. \frac{X_{\text{rand}} - X_{\text{near}}}{|X_{\text{rand}} - X_{\text{near}}|} \right]。 \quad (1)$$

式中 $0 < P < 1$ 。

如图9所示, 应用目标偏置系数可以使得对地图空间的探索更具目的性, 但实验发现当 P 较小时, 路径树的搜索更具随机性, 但同时在地面空间的搜索时间更长; 当 P 较大时, 路径树的生长更具目的性, 但随机性减小, 难以处理遇到较大障碍物时的情况, 搜索失败率提高。因此在路径树生长过程中引入自适应偏置系数应用于新节点的选择, 根据 X_{near} 与离它最近的障碍物的直线距离 L_{en} 来设置偏置系数 P 。若 L_{en} 较大, 则增大 P 的值, 使得路径树能够更加快速往对方的方向逼近; 若 L_{en} 较小, 则先使得路径树迅速规划路线逃离障碍物, 再增大 P 的值, 迅速往目标方向逼近。引入自适应偏置系数的目标偏置算法路径规划结果如图10所示。

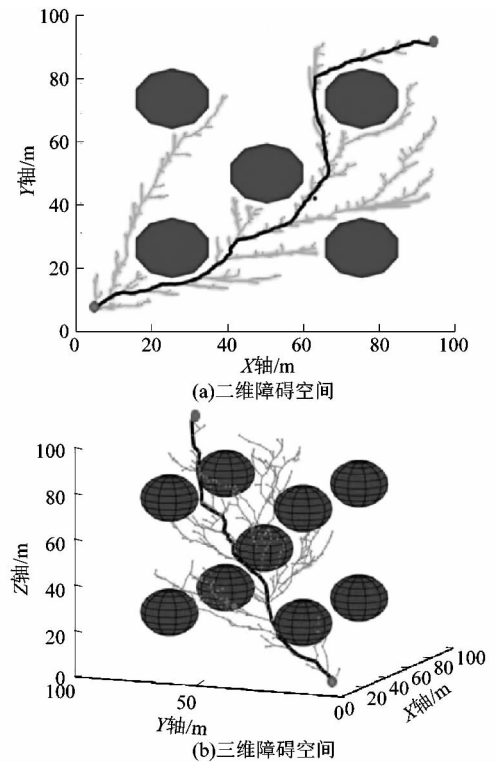
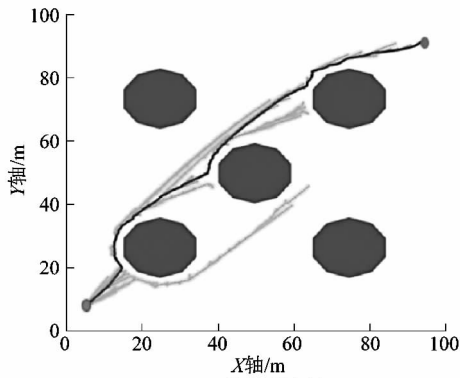


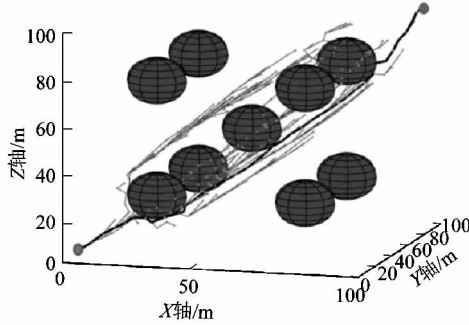
图9 目标偏置算法障碍空间中搜索

Figure 9 Search in obstacle space of target bias algorithm

自适应偏置函数 $\text{GoalExtend}(x_{\text{rand}}, x_{\text{near}}, G)$ 使得路径树向另一棵树的方向生长, 得到其伪代码。



(a) 二维障碍空间



(b) 三维障碍空间

图 10 引入自适应偏置系数的目标偏置算法

Figure 10 Target bias algorithm with adaptive bias coefficient

```

 $x_{rand}, x_{near}, G$ 
 $d \leftarrow \text{dist}(x_{near}, \text{world})$ 
 $P \leftarrow \text{Bias}(d)$ 
 $x_{new} \leftarrow f(P)$ 
return  $x_{new}$ 

```

2.4.2 重置父节点

RRT* 算法利用重置父节点和节点重构 2 个方法来降低路径树生长成本,创造最优的路径收敛效果,但在每一次的迭代中都存在重置父节点和节点重构会较大地影响路径树的生长速度。且局部采样的步骤提前会使得搜索路径树的随机性降低,甚至有可能降低生成的路径树质量。改进的 RRT 算法为路径树中生成的新节点扩大重置父节点的范围并重置父节点,不做节点重构。改进的 RRT 算法中重置父节点范围的过程如下^[14]:

1) 生成节点 x_{new} , 并找出 x_{near} 的点集合;

2) 遍历 x_{near} 点集合中的节点,将这些节点的 n 级父节点加入 x_{near} 的点集合中;

3) 合并 x_{near} 与 x_{near}' 的点集合,作为 x_{new} 重选父节点的范围。

计算并保存生成的新节点到起始点的代价函数 F_{min} ,将生成的新节点 x_{new} 作为中心,遍历路径树上和 x_{new} 的距离小于 r 且满足函数 $\text{Collectionfree}()$ 的节点,将这些节点作为 x_{new} 的父节点,把它们与 x_{new} 的距离 H_n 计算保存下来,计算 x_{new} 的代价函数 $F_n = G_n + H_n$,将 F_n 的最小值与 F_{min} 作比较,若最小值小于 F_{min} ,则重置父节点,重新选用 F_n 的最小值所对应的节点为 x_{new} 的父节点。

2.4.3 改进的 RRT 算法伪代码

得到改进的 RRT 算法伪代码。

```

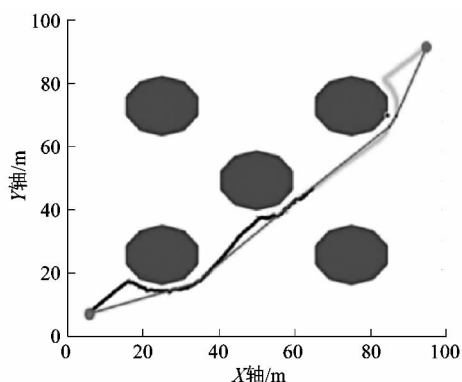
 $nT, T_a, T_b$ 
for  $i = 1$  to  $N_{set}$ 
 $x_{rand} \leftarrow \text{Sample}()$ 
 $x_{near} \leftarrow \text{Nearest}(T_a, x_{rand})$ 
 $x_{new} \leftarrow \text{goalExtend}(x_{near}, x_{rand}, T_b, \text{StepSize})$ 
 $T_a.add(x_{new})$ 
 $x_{near} \leftarrow \text{Findnear}(x_{new}, T_a, r)$ 
Chooseparent( $x_{new}, x_{near}$ )
flag = false
if Getit( $x_{new}$ )
    flag = true
else
    swap( $T_a, T_b$ )
if flag = true
     $T = T_a + T_b$ 
     $nT = \text{Trim}(T)$ 
    return( $nT, \text{flag}$ )
return(flag)

```

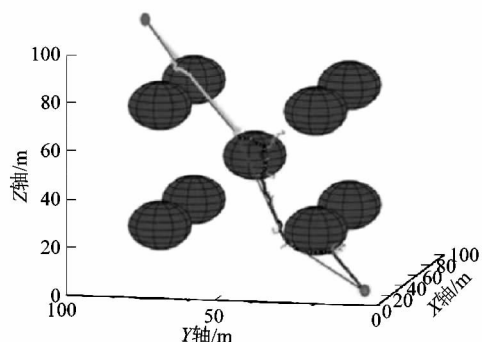
图 11 所示为改进后的 RRT 算法分别在二维障碍空间和三维障碍空间中搜索路径的结果,可以看出改进 RRT 算法路径曲率较小,长度较短,质量较高。

3 路径后处理

改进后的 RRT 算法虽然能够有效提高地图空间中的 RRT 搜索效率,但因其搜索能力更强,也会导致路径的扩展方向变化较大,优化后路径仍有许多冗余点。在有障碍物的复杂地图空间中,冗余点过多会导



(a)二维障碍空间



(b)三维障碍空间

图 11 改进的 RRT 算法搜索路径的结果

Figure 11 Improved RRT algorithm for searching path results

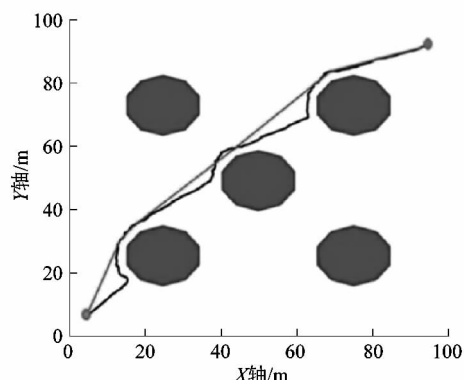
致无法有效跟踪机械臂的运动^[15]。因此对得到的路径做剪枝后处理是必要操作。

剪枝处理为路径树剔除冗余点,将剩下的关键拐点连接起来生成更优的路径树。本研究的剪枝处理算法 Trim(T)应用了贪婪策略,路径树是由位姿点集 nT 形成的一条连接起始点和目标点的路径。首先将路径树生长过程中生成的节点全部考虑,然后检查相邻节点的连接线是否接触地图空间中布置的障碍物。只要选取的 2 端节点的连接线和障碍物没有交叉,则删除 2 端节点之间的其他节点,2 端节点可以直接连接作为一个新的路径,直到发生碰撞,将碰撞点的父节点作为新的起点,然后碰撞点再次被持久化,重复以上的步骤直到达到目标点^[16]。如图 12 所示,经过剪枝处理后的路径在机械臂的实际应用中更为合理,减少了冗余点和路径的长度。

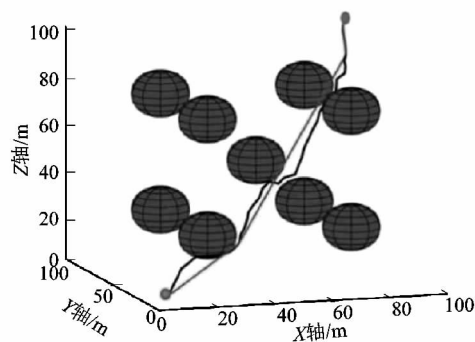
4 实验结果与分析

4.1 对比实验

基于对 6 自由度机械臂运动过程中速度和稳定性



(a)二维障碍空间



(b)三维障碍空间

图 12 剪枝处理前后得到的路径

Figure 12 Path obtained before and after pruning

的要求,利用 MATLAB 仿真软件对课题组提出的改进 RRT 算法进行仿真,并将改进算法中的关键参数与不同算法产生的效果进行比较^[17],来判定算法的质量。

在二维的确定障碍空间内进行对比实验,将搜索时间作为判断路径搜索速度的指标;将路径长度和最大曲率作为判断路径质量的指标。考虑到其他因素对实验的影响,取各算法运行 100 次的平均值作为结果参数,设置搜索步长 $S_s = 0.5$ 。由于 RRT* 算法具有渐进最优的特性,因此实验中发现 RRT* 算法搜索 20 000 次后生成的路径基本一致,课题组将 RRT* 算法搜索 20 000 次后生成的路径长度为最短路径。实验采用 Intel Xeon(至强) W-2255@3.70 GHz 的 10 核处理器,实验结果如表 2 所示。

对比实验结果显示,RRT 算法搜索速度相对较慢,路径质量也较差;RRT-connect 算法因为应用了贪婪算法,因此每一次迭代都会生成多步搜索,路径规划的单次时间最短,搜索速度最快,但 RRT-connect 算法生成的路径最大曲率相对另外 2 种 RRT 改进算法要

大得多,且生成的路径长度最大,超过 RRT* 算法生成的路径长度 37.5%,路径质量较差;RRT* 算法产生的路径长度最小,最大曲率也仅仅略高于本研究中改进的 RRT 算法,路径质量较好,但单次搜索时间远高于其他 3 种算法,搜索速度太慢。

表 2 实验结果对比

Table 2 Comparison of experimental results

算法	实验次数	单次搜索 时间/s	迭代次数	路径 长度/m	最大 曲率/%
RRT*	1	218.63	20 000	133.39	0.127
基本 RRT	100	4.01	11 310	168.64	7.190
RRT-connect	100	0.14	45	183.47	5.676
改进 RRT	100	0.21	1 776	136.80	0.041

改进的 RRT 算法仅在单次运行时间上略长于 RRT-connect 算法,但该算法生成的路径最大曲率相对较小,路径长度略高于 RRT* 算法,超过 RRT* 算法规划的路径长度的 2.6%,实验结果显示改进的 RRT 算法在保证搜索效率的基础上,大幅提高了生成的路径质量。

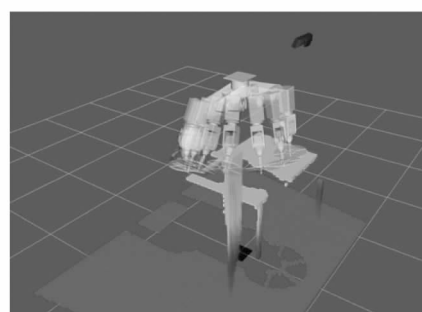
4.2 基于 ROS 的仿真模拟

在 ROS 平台的 Rviz 仿真软件中模拟手术室的工作环境^[18],将基于自适应目标偏置系数的 RRT 算法应用在 6 自由度机械臂上,通过机械臂避障规划与控制实验来验证算法的有效性。图 13 所示为 6 自由度机械臂在 2 种不同的环境中做路径规划的仿真过程^[19]。

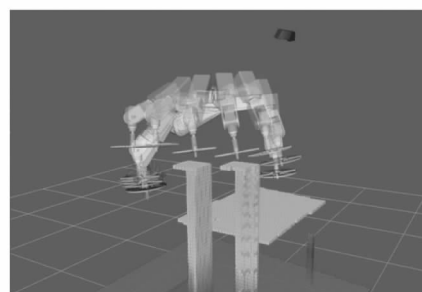
图 14 所示为 6 自由度机械臂分别在这 2 种模拟环境中工作时各个关节的角度随时间变化的曲线。通过曲线可以看出不同环境中关节的运动存在差异,但各关节变化的曲线都较为平滑,展现了 6 自由度机械臂模拟工作时稳定性较好,且冗余的路径长度被有效减少。

5 结语

课题组提出了一种基于自适应目标偏置系数的机械臂路径规划算法,利用 MATLAB 对该算法进行仿真实验,将实验结果与已有的 RRT 算法及其改进算法的路径规划结果作对比分析,并在基于 ROS 的仿真平台进行验证。实验结果表明:应用自适应目标偏置系数的改进 RRT 算法兼顾搜索效率和路径质量,且应用贪



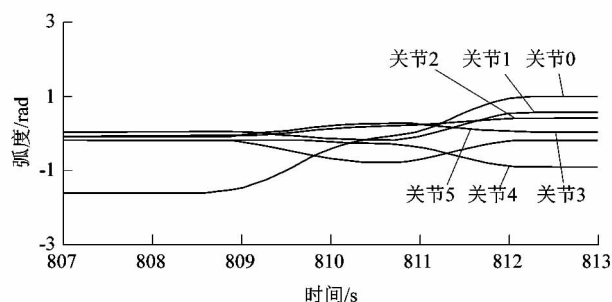
(a) 障碍空间 I



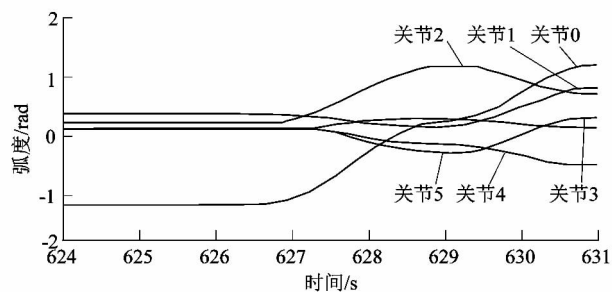
(b) 障碍空间 II

图 13 机械臂在变化环境中的路径规划

Figure 13 Path planning of manipulator in changing environment



(a) 障碍空间 I



(b) 障碍空间 II

图 14 仿真过程中各关节弧度变化曲线

Figure 14 Curve of each joint change during simulation

婪策略的剪枝后处理算法有效减少了冗余节点和路径长度,该算法有较好的避障性能和寻优能力,符合在实际应用中对机械臂工作稳定性的要求。课题组改进的RRT算法模拟仿真的地图环境中只有静态障碍物,没有考虑到有动态障碍物的地图环境。接下来的研究应围绕如何将应用场景扩大到添加了动态障碍物的地图环境中对算法作进一步改进,提高算法的实用性。

参考文献:

- [1] YUAN C R, LIU G F, ZHANG W Q. 6-DOF industrial manipulator motion planning based on RRT-connect algorithm [M]. Singapore: Springer, 2020: 93 – 101.
- [2] JIANG A, YAO X, ZHOU J. Research on path planning of real-time obstacle avoidance of mechanical arm based on genetic algorithm [J]. Journal of Engineering, 2018 (16): 1579 – 1586.
- [3] LAVALLE S M. Rapidly-exploring random trees: a new tool for path planning [D]. Ames: Iowa State University, 1998: 293 – 308.
- [4] KUFFNER J J, LAVALLE S M. RRT-connect: an efficient approach to single-query path planning [C]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA). San Francisco: IEEE, 2000: 995 – 1001.
- [5] KARAMAN S, FRAZZOLI E. Incremental sampling-based algorithms for optimal motion planning [J]. Robotics: Science and Systems, 2011, 104(2): 267 – 274.
- [6] 马慧丽, 鲁照权, 王寿庭. 基于改进 RRT 算法的机械臂路径规划研究 [J]. 机械设计与研究, 2020, 36(4): 42 – 46.
- [7] LI Y J, WEI W, GAO Y, et al. PQ-RRT*: an improved path planning algorithm for mobile robots [J]. Expert Systems with Applications, 2020, 152: 113425.
- [8] KANG J G, LIM D W, CHOI Y S, et al. Improved RRT-connect algorithm based on triangular inequality for robot path planning [J]. Sensors, 2021, 21(2): 333.
- [9] 黄壹凡, 胡立坤, 薛文超. 改进 RRT-Connect 算法的移动机器人路径规划 [J/OL]. 计算机工程: 1-9 [2021-06-14]. <https://doi.org/10.19678/j.issn.1000-3428.0060192>.
- [10] 谭建豪, 潘豹, 王耀南, 等. 基于改进 RRT* FN 算法的机器人路径规划 [J/OL]. 控制与决策: 1-7 [2021-06-14]. <https://doi.org/10.13195/j.kzyjc.2019.1713>.
- [11] LIANG Y, MU H, CHEN D, et al. PR-RRT*: motion planning of 6-DOF robotic arm based on improved RRT algorithm [C]//Proceedings of 10th IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). Xi'an: IEEE, 2020: 417 – 422.
- [12] JIANG L Q, WANG S T, MENG J, et al. Improved double-tree RRT* algorithm for efficient path planning of mobile robots [C]//2020 IEEE Region 10 Conference (Tencon). Osaka, Japan: IEEE, 2020: 206 – 211.
- [13] 蔡文涛, 邓屹, 张静, 等. 基于改进 RRT 算法的机械臂路径规划 [J]. 传感器与微系统, 2019, 38(5): 121 – 124.
- [14] 李兆强, 张时雨. 基于快速 RRT 算法的三维路径规划算法研究 [J/OL]. 系统仿真学报: 1-10 [2021-06-14]. <http://kns.cnki.net/kcms/detail/11.3092.v.20210202.0858.002.html>.
- [15] 李金良, 舒翰儒, 刘德建, 等. 基于改进 RRT 路径规划算法 [J]. 组合机床与自动化加工技术, 2021(2): 22 – 24.
- [16] LIU B, FENG W Z, LI T T, et al. A variable-step RRT* path planning algorithm for quadrotors in below-canopy [J]. IEEE Access, 2020, 8: 62980 – 62989.
- [17] SHAO L, LIU H Z, CHEN C, et al. Path planning for mobile robots based on improved RRT algorithm [C]//2020 IEEE International Conference on Mechatronics and Automation (ICMA). Beijing: IEEE, 2020: 1240 – 1244.
- [18] 李春光, 李宁健, 徐宜瑞, 等. 基于 RRT 的 ROS 全局路径规划 [J]. 电子技术与软件工程, 2019(5): 95 – 96.
- [19] 高朋帅. 基于六自由度机器人的智能手术无影灯路径规划与仿真 [D]. 上海: 上海理工大学, 2021: 62.

(上接第 33 页)

- [7] 刘润晨. 混联腿足式四足步行机器人的运动学建模与步态规划研究 [D]. 济南: 山东大学, 2020: 8 – 10.
- [8] 汪永明, 马腾飞, 韦强. 闭链式四足机器人对角小跑步态规划与仿真分析 [J]. 机电工程, 2020, 37(5): 553 – 558.
- [9] 黄真, 赵永生, 赵铁石. 高等空间机构学 [M]. 北京: 高等教育出版社, 2006: 118 – 130.
- [10] 李亮, 李思凡, 王海芳. Delta 机器人分拣平台的机构尺度综合与分析 [J]. 中国工程机械学报, 2021, 19(2): 158 – 163.
- [11] 郭晖晖. 四足机器人步态规划与运动控制研究 [D]. 南京: 南京航空航天大学, 2017: 35 – 40.
- [12] 田晓杰. 小型四足机器人步态规划及生成方法研究 [D]. 济南: 山东大学, 2020: 28 – 30.
- [13] 何冬青, 马培荪. 四足机器人动态步行仿真及步行稳定性分析 [J]. 计算机仿真, 2005, 22(2): 146 – 149.