

基于 RRT 搜索算法的六自由度 机械臂避障路径规划*

代彦辉, 梁艳阳, 谢 钢

(西南科技大学, 信息工程学院, 四川 绵阳 621010)

摘 要:针对三维空间下的机器人路径规划问题,提出了一种改进的快速随机树算法。该算法以机器人的出发点作为随机树的根节点,以基准向量为基准选择节点的扩展方向,逐渐增加叶子节点,降低了节点的随机性,提高了扩展速度。通过运动学反解快速规划出一条全局优化路径,对于反解失败的点,做了规划替代,保证了路径规划的平滑性,降低了机器人工作过程中的抖动,从而得到能够抵达目标位姿的安全路径。

关键词:机器人;路径规划;运动学正反解;快速随机搜索树

中图分类号:TP241.2 文献标识码:A 文章编号:1003-7241(2012)10-0031-07

The Mechanical Arm of Six- Dof Obstacle Avoidance Path Planning Based on Rapid-Exploring Random Trees

DAI Yan-hui, LIANG Yan-yang, XIE Gang

(College of Information Science & Engineering Southwest university of science and technology, Mianyang 621010 China)

Abstract: A new improved rapid-exploring random tree algorithm for path planning of robot is proposed to plan an optimal path in complex environment. The starting point as root node of the random tree, and the standard vector as benchmark to choose direction, by extending, gradually increase the leaf nodes. The algorithm Improve the speed of expansion. Through the inverse kinematics rapid planning out a global optimization path, to solve the point of failure, instead of a reasonable method, makes the path smoothness, ensures mechanical arm steady at work, and finds a security path to reach target pose.

Key words: robot; path planning; inverse and forward kinematics; rapidly-exploring random tree

1 引言

路径规划是机器人学要研究的重要课题,主要解决在障碍物环境下的机器人自主寻找一条从给定起点到终点的路径。传统的路径规划规划算法有栅格法,人工势场法,遗传算法,蚁群算法优化算法,启发式搜索法等。三维空间中,栅格法的栅格划分对路径规划影响很大,复杂度明显提高,以上方法需要建立模型在二维空间比较实用,对于三维空间的路径规划则难度加大,不适合解决多自由度机器人在三维环境中的路径规划。

基于快速扩展的随机树的路径规划算法,通过对状态空间中的采样点进行碰撞检测,避免了三维空间建模建立难度大的问题。该方法的优点是能够快速有效地搜索高维空间,在笛卡尔空间中,通过对未知空间的随机采样,使路径导向无障碍区域。从而能够有效地规划出一条从起始点到目标点的路径。

由于 RRT 算法采样的随机性,可能使得新生成的节点远离目标节点。因此,本文在确定随机节点时,通过选择基准向量,确保采用多点采样选取,确保事件的概率完备性。通过赌轮法决定朝着那一个随机节点的方向发展。本文采用通用型六自由度鉴定机械臂为研究对象,通过 matlab 可视化界面研究对象。

* 基金项目:四川省教育厅青年基金(编号10zd1135)

收稿日期:2012-05-17

2 六自由度机械臂的运动学正反解

文中采用通用型六自由度机械臂机械结构如图1所示,通常情况下,为了方便进行运动学的求解,采用D-H法就可以确定第*i*-1坐标系和第*i*坐标系之间变换矩阵,模型如图2所示, $x_0-y_0-z_0$ 为基坐标系, $x_t-z_t-y_t$ 为机械臂末端执行器坐标系。依据基坐标变换矩阵到各个关节的坐标关系,可以得到执行器的位姿矩阵

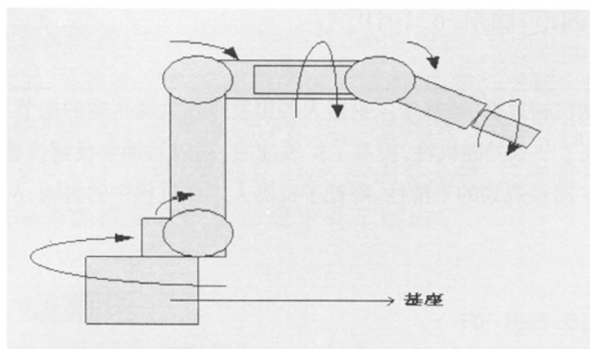


图1 六自由度机械臂机械结构

$$T = T_1^0(\theta_1)T_2^1(\theta_2)T_3^2(\theta_3)T_4^3(\theta_4)T_5^4(\theta_5)T_6^5(\theta_6) \quad (1)$$

根据机械臂的物理参数以及图2所示的坐标定义,可得到D-H参数如表1所示。依据(1)式和表1的参数,可以依次得到各个关节在基坐标下的位姿矩阵。运动学正解算法比较简单也比较成熟,本文不在阐述。

2.1 运动学反解

依据DH原理,依次右乘第*i*个关节角的坐标系变换矩阵,依次类推,可得到末端执行器坐标系在世界坐标系中的坐标变换矩阵为由 ${}^B T = {}^B T_0 T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T$

根据图2和表1,机械臂末端执行器的初始位置姿态为 $P_0 = [a_1 + d_4 + a_6 \quad 0 \quad a_2 + a_3]$, 其中 ${}^B T$ 、 ${}^B T_0$ 和 ${}^6 T$ 已知,从而可得到

$${}^0 T = {}^0 T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T = ({}^B T)^{-1} * {}^B T * ({}^6 T)^{-1} = \begin{bmatrix} n_x & o_x \\ n_y & o_y \\ n_z & o_z \\ 0 & 0 \end{bmatrix} \quad (2)$$

根据式(2)有

表1 六自由度的D-H参数表

| 连杆序号 <i>i</i> | a_{i-1} <i>mm</i> | α_{i-1} ($^\circ$) | θ_i ($^\circ$) | d_i <i>mm</i> | 关节变量范围 ($^\circ$) |
|---------------|------------------------|--------------------------------|----------------------------|--------------------|------------------------|
| 1 | $a_0 = 0$ | $\alpha_0 = 0$ | $\theta_1(0)$ | $d_1 = 0$ | -170—170 |
| 2 | $a_1 = 85$ | $\alpha_1 = -90$ | $\theta_2(-90)$ | $d_2 = 0$ | -92—135 |
| 3 | $a_2 = 380$ | $\alpha_2 = 0$ | $\theta_3(0)$ | $d_3 = 0$ | -129—116 |
| 4 | $a_3 = 100$ | $\alpha_3 = -90$ | $\theta_4(0)$ | $d_4 = 425$ | -160—160 |
| 5 | $a_4 = 0$ | $\alpha_4 = 90$ | $\theta_5(0)$ | $d_5 = 0$ | -120—120 |
| 6 | $a_5 = 0$ | $\alpha_5 = -90$ | $\theta_6(0)$ | $d_6 = 0$ | -360—360 |
| <i>t</i> | $a_6 = 85$ | 0 | 0 | 0 | 0 |

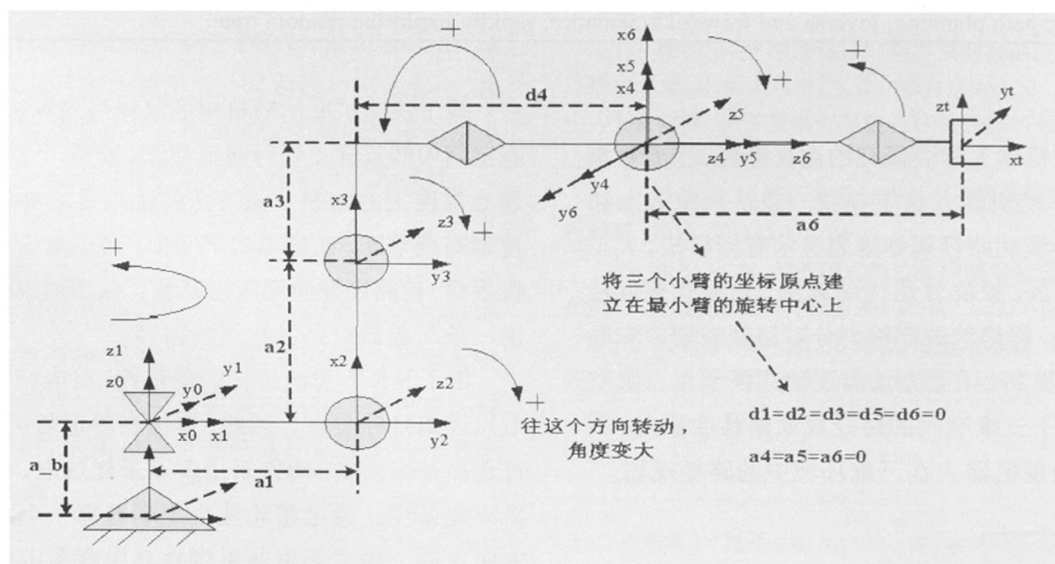


图2 采用D-H表示的六自由度机器人的关节坐标

$$\begin{aligned} {}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 &\Rightarrow {}^2T_4 {}^3T_5 {}^4T_6 = ({}^1T)^{-1} * ({}^0T)^{-1} * {}^6T * ({}^5T)^{-1} \\ &= {}^2T_4 {}^3T_5 {}^4T_6 = {}^2T_5 T = \begin{bmatrix} {}^2R & [{}^2P_x & {}^2P_y & {}^2P_z]^T \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R' & [P'_x & P'_y & P'_z]^T \\ 0 & 1 \end{bmatrix} \end{aligned}$$

其中,

$$\begin{aligned} \frac{2}{5}P_x &= \cos(\theta_3) * a_3 - \sin(\theta_3) * d_4 + a_2 \\ \frac{2}{5}P_y &= \sin(\theta_3) * a_3 + \cos(\theta_3) * d_4 \\ \frac{2}{5}P_z &= 0 \\ P'_x &= \cos(\theta_2) * \cos(\theta_1) * p_x + \cos(\theta_2) * \sin(\theta_1) * p_y \\ &\quad - \sin(\theta_2) * p_z - \cos(\theta_2) * a_1 \\ P'_y &= -\sin(\theta_2) * \cos(\theta_1) * p_x - \sin(\theta_2) * \sin(\theta_1) * p_y \\ &\quad - \cos(\theta_2) * p_z + \sin(\theta_2) * a_1 \\ P'_z &= -\sin(\theta_1) * p_x + \cos(\theta_1) * p_y \end{aligned}$$

$$P'_z = \frac{2}{5}P_z \Rightarrow -\sin(\theta_1) * p_x + \cos(\theta_1) * p_y = 0 \Rightarrow \theta_1 = \begin{cases} \arctan 2(p_y, p_x) \\ \arctan 2(-p_y, -p_x) \end{cases}$$

$$\begin{aligned} \text{由 } P'_x = \frac{2}{5}P_x &\Rightarrow \cos(\theta_3) * a_3 - \sin(\theta_3) * d_4 = \\ &\quad \cos(\theta_2) * \cos(\theta_1) * p_x + \cos(\theta_2) * \sin(\theta_1) * p_y \\ &\quad - \sin(\theta_2) * p_z - \cos(\theta_2) * a_1 - a_2 \end{aligned} \quad (3)$$

$$\begin{aligned} \text{由 } P'_y = \frac{2}{5}P_y &\Rightarrow \sin(\theta_3) * a_3 + \cos(\theta_3) * d_4 = \\ &\quad -\sin(\theta_2) * \cos(\theta_1) * p_x - \sin(\theta_2) * \sin(\theta_1) * p_y \\ &\quad - \cos(\theta_2) * p_z + \sin(\theta_2) * a_1 \end{aligned} \quad (4)$$

$$\text{令 } u = \cos(\theta_1) * p_x + \sin(\theta_1) * p_y - a_1, v = p_z,$$

$$h = \frac{u^2 + v^2 - a_2^2 - a_3^2 - d_4^2}{2a_2} \quad \text{联立(3)和(4)式}$$

$$\text{可得到 } \theta_2 = a \tan 2(-v, u) \pm a \tan 2(\sqrt{u^2 + v^2 - m^2}, m)$$

$$\theta_3 = a \tan 2(-d_4, a_3) \pm a \tan 2(\sqrt{a_3^2 + d_4^2 - h^2}, h)$$

$$\text{由 } {}^0T = {}^0T_3 {}^3T_6 T \Rightarrow {}^3T = ({}^0T)^{-1} ({}^6T), \text{ 得到}$$

$$\begin{bmatrix} \cos(\theta_4) * \cos(\theta_5) * \cos(\theta_6) - \sin(\theta_4) * \sin(\theta_6) \\ \sin(\theta_5) * \cos(\theta_6) \\ -\sin(\theta_4) * \cos(\theta_4) * \cos(\theta_6) - \cos(\theta_4) * \sin(\theta_6) \\ 0 \\ -\cos(\theta_4) * \cos(\theta_5) * \sin(\theta_6) - \sin(\theta_4) * \cos(\theta_6) \\ -\sin(\theta_5) * \sin(\theta_6) \\ \sin(\theta_4) * \cos(\theta_5) * \sin(\theta_6) - \cos(\theta_4) * \cos(\theta_6) \\ 0 \\ -\cos(\theta_4) * \sin(\theta_5) \\ \cos(\theta_5) \\ \sin(\theta_4) * \sin(\theta_5) \\ 0 \end{bmatrix} \quad (5)$$

$${}^3T = ({}^0T)^{-1} ({}^6T) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \quad (6)$$

式(6)中的元素是已知的, 其中,

$$\begin{aligned} r_{11} &= (c_1c_2c_3 - c_1s_2s_3) * n_x + (s_1c_2c_3 - s_1s_2s_3) * n_y - (s_2c_3 + c_2s_3) * n_z \\ r_{12} &= (c_1c_2c_3 - c_1s_2s_3) * o_x + (s_1c_2c_3 - s_1s_2s_3) * o_y - (s_2c_3 + c_2s_3) * o_z \\ r_{13} &= (c_1c_2c_3 - c_1s_2s_3) * a_x + (s_1c_2c_3 - s_1s_2s_3) * a_y - (s_2c_3 + c_2s_3) * a_z \\ r_{21} &= (-c_1c_2s_3 - c_1s_2c_3) * n_x - (s_1c_2s_3 + s_1s_2c_3) * n_y + (s_2s_3 - c_2c_3) * n_z \\ r_{22} &= (-c_1c_2s_3 - c_1s_2c_3) * o_x - (s_1c_2s_3 + s_1s_2c_3) * o_y + (s_2s_3 - c_2c_3) * o_z \\ r_{23} &= (-c_1c_2s_3 - c_1s_2c_3) * a_x - (s_1c_2s_3 + s_1s_2c_3) * a_y + (s_2s_3 - c_2c_3) * a_z \\ r_{33} &= -s_1p_x + c_1p_y \end{aligned}$$

根据式(5)和(6), 可计算出剩余的三个角度值。

$$\theta_5 = a \tan 2(\pm \sqrt{1 - r_{23}^2}, r_{23})$$

$$\text{当 } \sin(\theta_5) \neq 0 \quad \theta_4 = a \tan 2(r_{33} * \sin(\theta_5), -r_{13} * \sin(\theta_5))$$

$$\theta_6 = a \tan 2(-r_{22} * \sin(\theta_5), r_{21} * \sin(\theta_5))$$

当 $\sin(\theta_5) = 0$ 机器人处于奇异位型, 连杆 5 和连杆

6 成一直线。 $\theta_4 + \theta_6 = a \tan 2(-r_{12}, r_{11})$ 实际使用时,

在保持 $\theta_4 + \theta_6$ 不变的前提下, 令 θ_4 与 θ_6 朝相反方向转动相同大小的角度, 从而避免奇异位置。

3 基于 RRT 的避障路径搜索

3.1 长方体包络碰撞检测法

为了解决机械臂整体与障碍之间的碰撞问题, 本文提出了长方体包络检测算法。机械臂为链式结构。为研究方便, 我们将其骨架化, 从而转化为空间中的直线段。

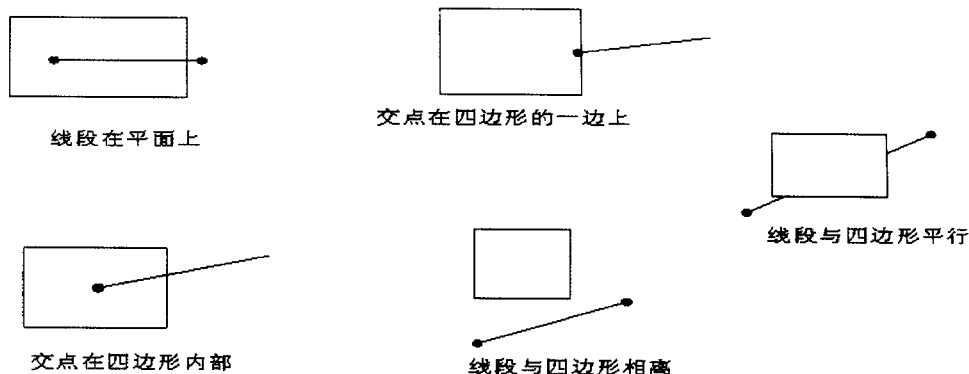


图3 空间线面关系的五种情况

通过控制搜索步长来补偿骨架化后机械臂模型带来的失真。障碍物一般具有不规则的几何形状,长方体包络更接近实际的障碍物模型,因此提高了精度,这样一来我们把问题转化为空间线段与空间四边形相交的问题,空间线面的位置关系有以下五种,交点在面内,交点在某条边上,线段在面内部,线段与面平行,线段与面相离。如图3所示。

给定空间平面上的一点 P_0 和法向量 N , 满足 $N \cdot (P - P_0) = 0$ 的点 P 的集合平面的方程一般可以表示为

$$Ax + By + Cz + D = 0 \quad (7)$$

其中 A, B , 和 C 为法向量 N 的 x, y, z 分量, $D = -N \cdot P_0$ 。

设 $P(t) = Q + tV$ 表示包含点 Q 并沿着 V 方向延伸的直线, 对于法向量为 N 并且距原点带符号距离为 D 的平面, 可以通过解下面的方程来确定直线与平面的交点, $N \cdot P(t) + D = 0$

$$\begin{aligned} &\text{用 } Q + tV \text{ 代替 } P(t), \text{ 可以得到} \\ &N \cdot Q + (N \cdot V)t + D = 0 \end{aligned} \quad (8)$$

$$\text{解关于 } t \text{ 的方程, 可得 } t = \frac{-(N \cdot Q + D)}{N \cdot V} \quad (9)$$

如果 $N \cdot V = 0$, 则说明线段与平面平行。则需要判断线段的点是否在平面上。如果 $N \cdot Q + D = 0$, 则线段在平面上。如何判断直线与平面的交点就是线段与四边形的交点。必须满足以下两个条件

(I) 交点必须在线段上

设空间线段的两个端点 $pt1(x_1, y_1, z_1)$ 和 $pt2(x_2, y_2, z_2)$ 交点 $point(x, y, z)$, 如果 $\min(x_1, x_2) \leq x \leq \max(x_1, x_2)$ 且 $\min(y_1, y_2) \leq y \leq \max(y_1, y_2)$ 且 $\min(z_1, z_2) \leq z \leq \max(z_1, z_2)$, 那末交点必定在线段上。

(II) 交点必须在四边形内部

把四边形沿对角线分成两个三角形, 假设点 P 位于三角形内, 当沿着 $ABCA$ 的方向在三条边上移动时, 点 P 始终位于边 AB, BC 和 CA 的右侧。利用这一特点, 当选定线段 AB 时, 点 C 位于 AB 的右侧, 同理选定 BC 时, 点 A 位于 BC 的右侧, 最后选定 CA 时, 点 B 位于 CA 的右侧, 当选择某一条边时, 只需检测 P 与该边所对的点在同一侧即可。通过向量叉积来实现, 连接 PA , 将 PA 和 AB 做叉积, 再将 CA 和 AB 做叉积, 如果两个叉积的结果的方向一致, 则两个点在同一侧。

3.2 RRT算法规划路径

基本 RRT 算法描述: C 是机械臂的可行笛卡尔空间, 且 $C \in R^3$ 。可用空间中的点 (x, y, z) 表示机器人空间位置坐标。快速随机树的构建过程如图4所示。 T_k 为一个拥有 k 个节点的机器人工作空间中随机树。 x_{init} 为初始状态, x_{goal} 为目标状态, 即检定台的表位。 x_{rand} 为在 C 空间随机选取的状态点。遍历随机树 T_k , 找到 T_k 上离 x_{rand} 距离最近的叶子节点 x_{near} , 令 $p, q \in C$, $dist(p, q)$ 代表空间的两个位姿的几何距离。如果 $L \leq dist(x_{near}, x_{rand})$ 则在 x_{near} 与 x_{rand} 的连线上求 x_{new} , 且 $dist(x_{near}, x_{new}) = L$ 。 L 为搜索步长, 如果存在 x_{new} , 则随机树增加一个新的节点, 否则重新选择一组 x_{rand} 作为备选节点, 重复以上过程; 直到达到目标点位置。扩展公式如下(11)所示:

$$x_{tmp} = x_{new} + L \cdot \frac{(x_{rand} - x_{near})}{\|x_{rand} - x_{near}\|} \quad (11)$$

但在 RRT 算法扩展树生成时, 新增的叶子节点应该距离目标节点越来越近, 由于扩展节点的方向是随机产生的, 因而会引入局部极小问题, 我们通过判断 x_{tmp} 与 x_{new} 之间的距离与 x_{tmp} 与其他叶子节点的距离确

定是否增加节点。如果前者较小则,增加该节点,否则取消增加。

3.2.1 算法改进

由于 x_{rand} 选取的随机性,从而导致从树的根节点 x_{init} 到叶子节点 x_{goal} 规划的路径有时接近目标路径,有时远离目标,随机性很大。下图为基本 RRT 随机树的扩展过程。

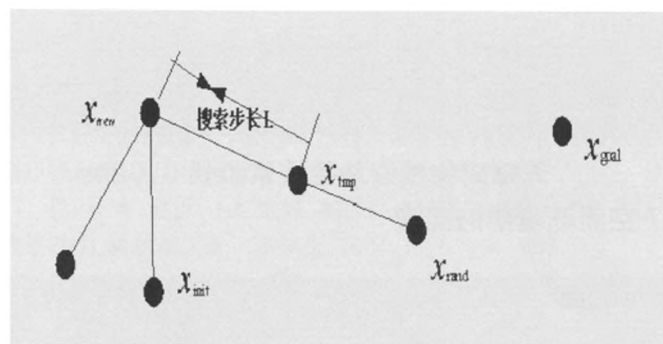


图4 RRT 随机树扩展

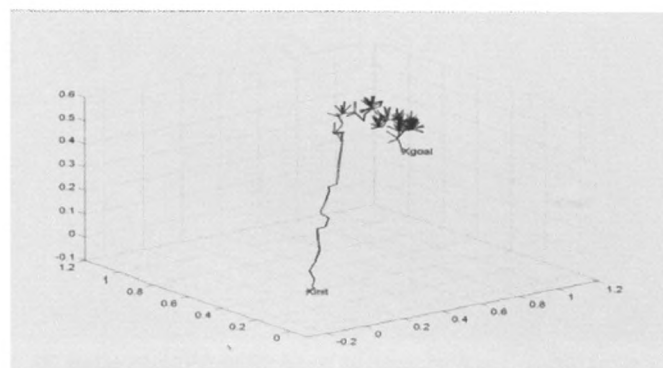


图5 基本 RRT 扩展树

针对 RRT 扩展过程中的这个特点,为了减少路径规划的随机性,使扩展树朝向最优目标点方向生长,由于随机树是按固定步长扩展节点,因此扩展的方向决定了路径规划性能。我们通过引入方向向量使得随机节点不至于偏离理想中的最优路径。即随机节点与目标节点的方向向量决定了路径的长短。

方向向量以随机树的当前父节点为基准点, x_{goal} 为目标节点确定方向向量 $direct = point(x_{goal}) - point(x_{near})$ 为基准向量,随机选取 20 个随机节点,然后计算 x_{rand} 与 x_{near} 确定的方向向量与基准向量之间的夹角。然后从小到大排序,选择前四个作为备选节点。通过赌轮法选择这四个随机节点扩展路径。详细过程参见步骤 3.3 中介绍。通过算法改进改进使得 RRT 能够有效地缩短规划时间,并且能够减少随机数的节点数,从而提高

了效率。

3.2.2 路径平滑

RRT 搜索之后的曲线折段,机械臂在运行中会发生抖动,我们采用三次 Hermite 曲线平滑处理,确保机械臂的速度连续性。Hermite 曲线是由 r_0, r_1, r_2, r_3 四个控制点组成的曲线,其中中间过渡点为 r_1 和 r_2 。它以曲线的两个端点 r_0, r_3 端点处的切向矢量 R_0, R_1 为边界条件的三次参数曲线,其中起点切线 $R_0 = r_1 - r_0$, 终点切线 $R_1 = r_3 - r_2$ 。

插值描述:1. 当起点为 $r_0 = p_0$ 时, 终点 $r_3 = p_2$, 中间过渡点 $r_1 = r_2 = p_1$, 记 $m = p_0 - p_2$, $n = p_5 - p_2$, $q = (m + n) / 2$, $f = cross(m, n)$, 则 $p_1 - p_2 = cross(q, f)$, $\|p_1 - p_2\| = 0.5 * \|m\|$; $p_3 - p_2 = -(p_1 - p_2)$;

2. 当 $r_0 \neq p_0$ 时, 若 $r_0 = p_2$, 则中间过渡点 $r_1 = p_3$, $r_2 = p_4$, $r_3 = p_5$;

规划曲线如(12)式所示:

$$p(t) = (2t^3 - 3t^2 + 1)p_0 + (t^3 - 2t^2 + t)2R_0 + (t^3 - t^2)3R_1 + (-2t^3 + 3t^2)p_3 \quad (12)$$

3.3 机械臂路径规划算法

Step1 初始化机械臂的起点 x_{init} 和目标点 x_{goal} , 在三维空间内建立障碍物模型。

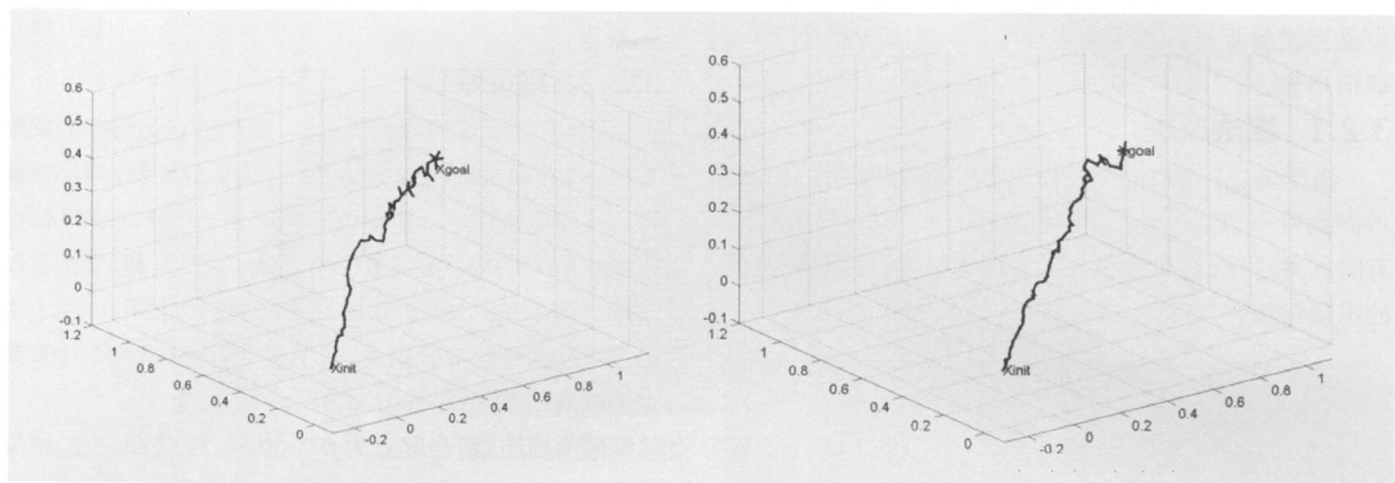
Step2 通过 $extendTree(point)$ 函数扩展叶子节点。

(1) 判断新增节点与目标节点的距离。如果到达目标节点即 $(dist(x_{new}, x_{goal}) \leq L)$, 则随机树扩展结束, 退出循环。

(2) 如果未达到目标节点则扩展节点, 遍历随机树, 找到随机树距离目标最近的节点 x_{new} 。选择随机节点 x_{rand} 扩展叶子节点, 为了减小随机性, 我们选择 20 个空间点作为备选随机节 x_{rand} , 并计算这 20 个随机节点 x_{goal} 与目标节点之间的距离 $dist(x_{rand}, x_{goal})$, 并保存到数组 $array[20]$, 并通过轮盘赌算法选取其中的四个随机数 x_{rand} , 按照欧式距离将其从小到大排序。

(3) 进入 4 次循环, 如果新节点 x_{new} 与随机节点 x_{rand} 的距离小于扩展步长 L , 本次不扩展, 继续选择 x_{rand} , 当 4 次循环结束时, 未找到随机节点 x_{rand} , 转(1)。判断新节点与其父节点组成的线段是否与障碍物发生碰撞, 如果没有碰撞则按(6)式扩展节点, 转(1)。当循环结束时, 认为找到扩展的叶子节点, 转(1)。

Step3 平滑搜索到的节点, 即用折线段(最小二乘法求得)逼近搜索路径, 并将其保存在 $node$ 数组中, 通



无障碍物搜索路径搜索步长 0.03m

无障碍物搜索路径搜索步长 0.02m

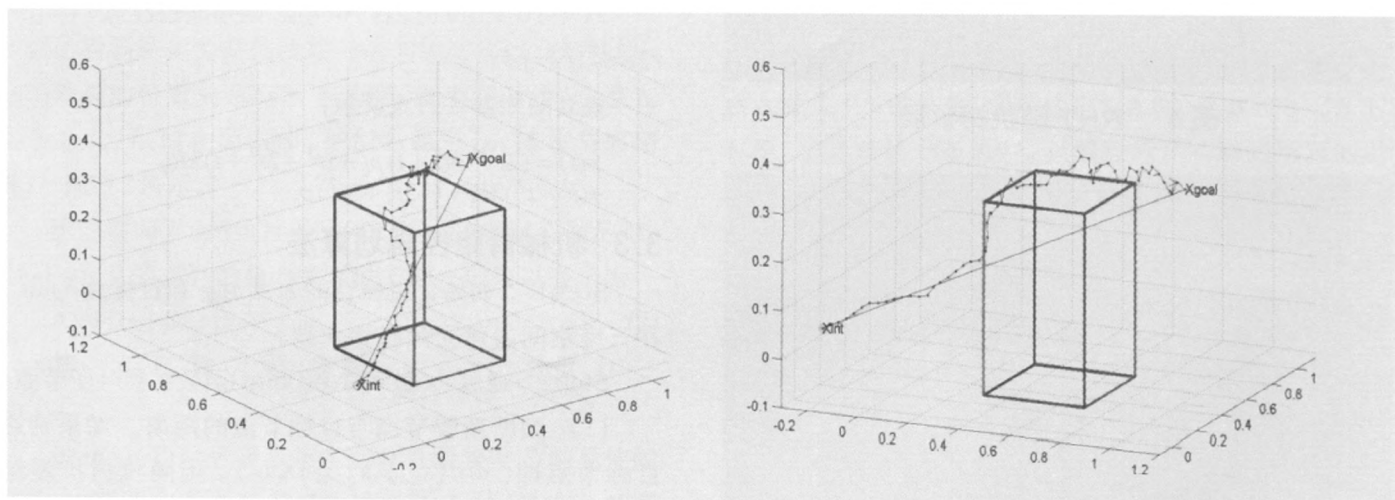
图6 $1.5 \times 1.3 \times 0.7$ 空间环境中的实验

图7 有障碍物的路径规划图

图8 从另一侧看到障碍物的路径规划图

过 Hermite 插值平滑曲线。

Step4 机械臂运动学反解, 执行器的空间某点位姿矩阵 $\text{pos_goal} = [2 * (\text{rand}(1, 3) - 0.5), [\text{new_node}(:, i)]]$, 代入 $\text{backKineticsXPcore}$ 函数进行运动学反解, 如果反解失败, 则在该点相邻的两点为对称轴, 找到其对称点坐标, 继续反解。

Step5 反解之后我们得到一系列 $[q_1, q_2, q_3, q_4, q_5, q_6]$ 关节角的集合, 然后通过运动学正解。得到六个关节的空间坐标, 再次进行碰撞检测。

Step6 如果反解失败, 则在该位姿点的相邻的两点的六个角之间随机取值, 至到得到安全的六个关节的角度。

4 仿真结果分析

图5所示的仿真实验是在 $1.5 \times 1.3 \times 0.7 / (\text{m}^3)$ 的空间下, 无障碍物, 取工作空间中的点 $(-0.2, 0.022, 0.05)$ 为起点, 点 $(0.9, 0.7, 0.3)$ 为目标点, 图6左边的 RRT 的搜索步长为 0.03, 图6右边的 RRT 的搜索步长为 0.05。图5左右两边的仿真图展现了一棵 RRT 的生长过程。

图5所示仿真实验是在 $1.5 \times 1.3 \times 0.7$ 的空间环境中, 在无障碍物时机械臂沿着直线的方向行进, 这也说明该算法接近最短路径。

下面为设障碍的仿真效果图, 障碍物位于起始位置与目标位置之间, x_{init} 与 x_{goal} 的连线刚好穿越障碍物, 实验环境与无障碍时相同, 障碍物是一长方体, 仿真结果如图7和图8所示。

5 结束语

由以上规划算法仿真图可以看出,RRT 算法具有随机性,基本 RRT 算法搜索的路径距离理想路径偏差较大,而且搜索速度较慢。改进后的算法减少了路径规划时间,修正了偏离目标点路径规划的方向。仿真结果表明:在无障碍物基本上是沿着直线,在有障碍时随机扩展,能够顺利避开障碍物,并且仍然以较近的路径到达目标位置。而且降低了运动学正反解的难度,提高了规划效率。

参考文献:

- [1] 王滨,金明河,谢宗武,刘宏.基于启发式的快速随机树路径规划算法[J].机械制造,2007-12,45(520):2-3.
- [2] 王华,赵臣,王红宝等.基于快速扫描随机树方法的路径规划器[J].哈尔滨工业大学学报,2004,36(7):964-965.
- [3] 郝利波,侯媛彬.基于一种改进RRT算法的足球机器人路径规划[J].西安科技大学学报,2011-01,31(1):81-82.
- [4] 美 Eric Lengyel 著,詹海生,李广鑫译,王鸿利译.3D

(上接第11页)

5 结束语

通过分析粒子群算法容易陷入局部极值的原因与学习因子在迭代过程中的作用,本文在简化粒子群算法的基础上,通过引入正弦函数动态调整学习因子,使得算法能够快速摆脱局部极值,提高了收敛速度。通过对4个测试函数进行试验,结果表明改进后的算法在搜优精度,收敛速度上都有显著的提高。然而如何选取以及有效调整算法中的某些参数算法还有待进一步的研究。

参考文献:

- [1] KENNEDY J, EBERHART R C. Particle Swarm Optimization[C]//Proc of IEEE Int'l Conf on Neural Networks, 1995:1942-1948.
- [2] QIN Y Q, SUN D B, LI M. Path planning for mobile robot using the particle swarm optimization with mutation operator[C]. Perth, Australia: Proc of Int Conf on Machine Learning and Cybernetics, 2004:2473-2478.
- [3] GUDISE V G, VENAYAGAMOORTHY G K. Comparison of particle swarm optimization and back propagation as training algorithms for neural networks[C]//Proc of IEEE Swarm Intelligence Symposium. 2003:110-117.
- [4] 丁坚勇,陶文伟,张文涛,基于模拟退火PSO的电力系统无功优化[J].武汉大学学报工学版,2008,41(2):94-98.
- [5] 胡建秀,曾建潮.微粒群算法中惯性权重的调整策略[J].计算机工程,2007,33(11):193-195.
- [6] EBERHART R C, KENNEDY J. A new optimizer

游戏与计算机图形学中的数学方法[M],清华大学出版社,2004.4.

[5] STEVEN M. LAVALLE, JAMES J. KUFFNER, Jr. Rapidly-Exploring Random Trees: Progress and Prospects [J]. Algorithmic and Computational Robotics. 2000:3-4.

[6] ZHAXF. optimal Pose Trajectory Planning for Robot Manipulators[J]. Mechanism and Machine Theory, 2002:1063-1086.

[7] 王红娟. 三维实体建模及布尔运算造型技术[D]. 山东济南: 山东科技大学 2007.

[8] 庄慧忠,杜树新,吴铁军,机器人路径规划及相关算法与仿真研究[J]. 科技通报,2004-5,20(3):211-212.

[9] 美 Steve Rabin 著. 庄越挺,吴飞译. 人工智能游戏编程真言[M]. 清华大学出版社,2005.1:79-81.

作者简介: 代彦辉(1985-),男,在读硕士,研究方向: 机器人视觉与智能算法。

using particle swarm theory[A]. Proceedings of the 6th International Symposium on Micro Machine and Human Science[C], Piscataway NJ: IEEE Service Center, 1995:39-43.

[7] 胡旺,李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报,2007(4):861-868.

[8] SHI Y, EBERHART R. A Modified Particle Swarm Optimizer[C]. IEEE Int. Conf. On Evolutionary Computation, Piscataway, NJ, IEEE Service Center, 1998.69-73.

[9] CLERC M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization [C]//In: Proc. of the ICEC. Washington: [s.n.], 1999:1951-1957.

[10] SHI Y, EBERHART R C. Empirical Study of Particle Swarm Optimization[A], proceeding of Congress on Evolutionary Computation[C]. Washing D C, 1999, 1945-1950.

作者简介: 任伟建(1963-)女,博士研究生,教授,研究方向: 复杂系统的建模与控制。