

基于离散采样的多模态四足机器人路径规划

孙帅帅¹, 冯春晓¹, 张 良²

(1. 中国科学技术大学 工程科学学院, 合肥 230026; 2. 安徽大学 电气工程与自动化学院, 合肥 230601)

摘 要: 针对双向快速扩展随机树(RRT-Connect)在多模态四足机器人路径规划中存在不必要的跳跃和行走部分路径的地形起伏程度大及转向角度变化大的问题, 提出了一种基于离散采样的解决方案。预处理路径去除不必要的跳跃, 离散采样并动态规划获得粗解, 使用B样条曲线拟合并二次规划得到最终路径。仿真结果表明, 本文方法规划出的路径使得机器人对质心高度的调节平均减少了31.4%, 途径地形的起伏程度减小13.4%, 地形倾斜角度变化降低11.4%, 转向角度变化减小62.7%, 验证了本文方法的有效性。

关键词: 人工智能; 四足机器人; 多模态; 路径规划; 离散采样; 二次规划

中图分类号: TP242 **文献标志码:** A **文章编号:** 1671-5497(2024)04-1120-09

DOI: 10.13229/j.cnki.jdxbgxb.20240155

Path planning for multimodal quadruped robots based on discrete sampling

SUN Shuai-shuai¹, FENG Chun-xiao¹, ZHANG Liang²

(1. College of Engineering Science, University of Science and Technology of China, Hefei 230026, China; 2. College of Electrical Engineering and Automation, Anhui University, Hefei 230601, China)

Abstract: Aiming at the challenges of unnecessary leap and significant undulations terrains with large steering angles in path planning of multimodal quadruped robots by Rapidly Exploring Random Tree algorithm, a path planning algorithm solution based on discrete sampling is proposed. The path is preprocessed to remove unnecessary leap and a solution set is obtained by discrete sampling and dynamic programming method. B-spline curves are used to define spline segments and quadratic programming method is used to optimize the final path. The simulation results show that paths planned by the proposed method exhibit an average reduction of 31.4% in the adjustment of robot's center of mass height, a 13.4% decrease in undulation of terrain, an 11.4% reduction in terrain slope angle and a 62.7% reduction in steering angle, which affirm the effectiveness of the proposed method.

Key words: artificial intelligence; quadruped robot; multimodal; path planning; discrete sampling; quadratic programming

收稿日期: 2024-02-07.

基金项目: 国家自然科学基金项目(52105081, 52005474, 62303002); 国家自然科学基金优秀青年科学基金项目(海外)(GG2090007004); 中国科学技术大学启动基金项目(KY2090000067).

作者简介: 孙帅帅(1989-), 男, 教授, 博士. 研究方向: 智能自适应机器人和振动控制. E-mail: sssun@ustc.edu.cn

0 引言

四足机器人因其四肢结构的良好稳定性和足式结构的灵活性,可在行进中实现多种运动模式切换^[1],如行走^[2]、爬行^[3]和跳跃^[4],在非结构化的室外复杂地形中表现出了优秀的适应能力^[5]。运动模式的多样性和控制系统的复杂性需要搭配高效稳定的路径规划算法,现有的四足机器人路径规划通常将问题简化为二维路径规划问题,没有考虑地形起伏和倾斜程度对运动稳定性的影响,且没有合理利用多模态四足机器人的运动功能。

全局路径规划算法主要分为基于搜索的算法和基于采样的算法^[6],基于采样的算法如快速扩展随机树^[7](Rapidly-exploring random trees, RRT)和概率路线图算法^[8](Probabilistic roadmap method, PRM),在复杂环境中多自由度机器人路径规划问题上的计算成本更低。高程栅格地图^[9,10]与二维栅格地图相比,带有高度和倾斜信息,能够更好地反馈地形^[11],因此,基于采样的算法与高程地图结合是目前四足机器人路径规划问题常见的解决方案,相关学者以二者为基础,在不同的技术方向上对该问题进行了研究。Wermelinger等^[12]对高程地图进行处理后得到标记有可通过性的地图,使用RRT*^[13]实现全局路径规划。该方法考虑了地形信息和四足机器人的行走能力,但没有考虑使用跳跃的方式越过障碍。Norby等^[14]提出基于动力学的RRT-Connect^[15]快速规划算法(KD-RRT-Connect)。该方法将四足机器人一次行走和跳跃组合为动作基元,通过执行一次动作基元实现搜索树节点的延伸,以此实现行走和跳跃模式相结合的规划,但该方法的行走阶段缺少对地形的综合考量,如地形的起伏和倾斜程度,并且不必要的跳跃会增加能耗和不稳定性。Wellhausen等^[16]使用概率路线图(PRM*),该方法考虑了地形可通过性和复杂地形的落足问题,充分发挥了四足机器人的行走能力,但使用该方法的前提是四足机器人的足端必须与地面存在接触,因此,无法对跳跃进行规划。

针对以上工作无法很好地解决多模态四足机器人使用多种运动模式组合越障和行走路线安全稳定的问题,本文基于KD-RRT-Connect算法,提出了针对多模态四足机器人的路径规划解决方案,预处理原始路径并离散采样获取周围地形信息,采用动态规划的方法获得新路线的粗解,最后

通过B样条曲线拟合和二次规划进行平滑优化得到最终路径,并通过仿真验证了算法的有效性。

1 KD-RRT-Connect问题描述

基于RRT-Connect算法合成运动是一种快速高效的方法^[17],Norby等^[14]以RRT-Connect为基础提出了适用于可跳跃的四足机器人的快速规划算法KD-RRT-Connect,该方法将一次短暂的行走和跳跃组合成动作基元,并简化使用行走和跳跃的初始加速度表达,如下所示:

$$\mathbf{a} = [\mathbf{a}_D^T \quad \mathbf{a}_O^T \quad t_s \quad t_f]^T \quad (1)$$

式中: $\mathbf{a}_D$ 为动作基元行走阶段的初始加速度; $\mathbf{a}_O$ 为动作基元跳跃阶段起跳时刻的加速度; t_s 和 t_f 分别为行走阶段和跳跃阶段的时间。

动作基元的加速度在满足 s_{rand} 处地形摩擦约束和最大地面反馈力限制的条件下随机生成,执行动作基元的过程作为随机树的节点延伸过程,如果该过程中的行走阶段,四足机器人未出现足端全部悬空和整个过程中身体未与地形发生碰撞,就成功在随机树中添加节点。对于与另一棵树的启发式延伸节点的过程,则使用跳跃时间为0的动作基元连接,即该过程为仅行走。初始树生成节点的过程如图1所示。

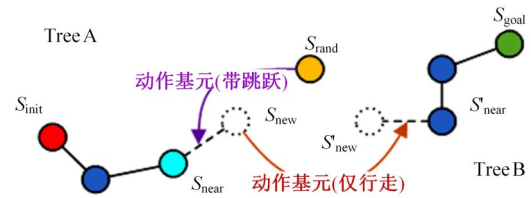


图1 KD-RRT-Connect生成节点

Fig. 1 Generating nodes for KD-RRT-Connect

在两棵树相连后,使用仅行走的动作基元尝试在长距离连接节点,连接时仍然需要进行地形碰撞检测,如果成功使用仅行走的方式完成连接,就剔除中间的树节点,对于尝试在长距离检测中连接的2个节点,使用式(2)(3)计算该行走过程的初始和末端加速度为:

$$\mathbf{a}_D = -\frac{6(s_D - s_O) + 2t_s(2\mathbf{v}_D + \mathbf{v}_O)}{t_s^2} \quad (2)$$

$$\mathbf{a}_O = \frac{6(s_D - s_O) + 2t_s(\mathbf{v}_D + 2\mathbf{v}_O)}{t_s^2} \quad (3)$$

式中: s_D 和 s_O 分别为起始节点和终点节点的位置; $\mathbf{v}_D$ 和 $\mathbf{v}_O$ 分别为起始节点和终点节点的速度; t_s 为该行阶段的持续时间。

遍历完成后再插值得到最终路径,由该过程可知,KD-RRT-Connect在延伸节点和最终插值生成路径时,仅考虑了四足机器人是否在行走时出现悬空和身体与地形发生碰撞,没有考虑实际路线的地形起伏和倾斜程度等信息,并且最终路径会存在不必要的跳跃,增加了四足机器人的能耗和完成任务的风险。

2 改进算法设计

本文改进算法设计如下:首先,对给定KD-RRT-Connect生成的原始路径 O 进行优化预处理,将不必要的跳跃重新规划为行走,得到路径 P ;其次,对路径 P 进行离散空间采样,从有效的采样点集合 Q 中获取总代价最小的粗解 R ;再次,对该集合进行三次B样条曲线拟合并样条插值,获取均匀点集 T ;最后,将平滑优化问题转化为二次规划问题并使用OSQP(Operator splitting quadratic programming)求解器进行求解,得到优化后的最终路径 U 。算法流程如图2所示。

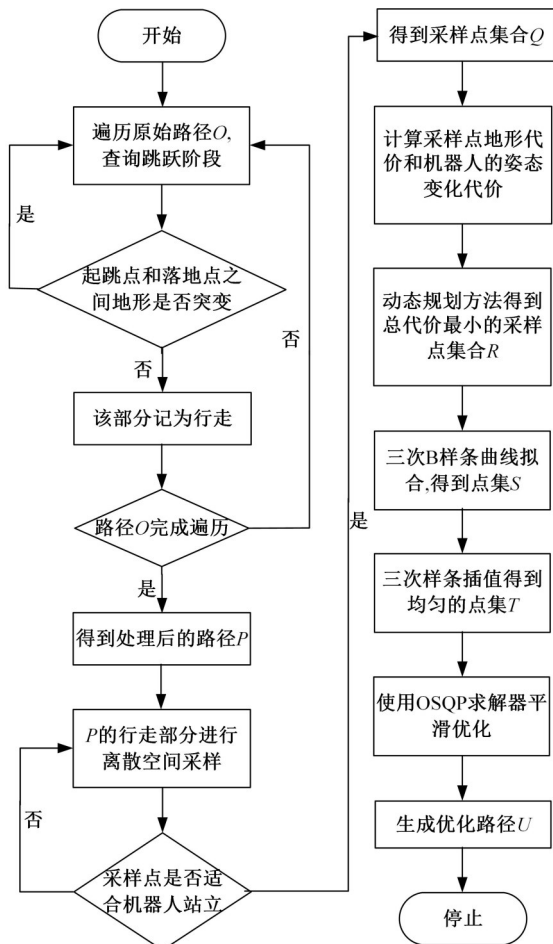


图2 算法流程图

Fig. 2 Flow chart of Proposed algorithm

2.1 路径预处理

KD-RRT-Connect规划的路径 O 包含了行走和跳跃2种运动模式。由于RRT类算法自身的随机特性^[18],路径 O 中的跳跃部分可能发生在没有突变的地形上,四足机器人在该段地形上可以选择更为可靠的行走方式通过,因此,本文的优化方法首先需要对路径 O 进行优化预处理,通过行走的方式替换不必要的跳跃动作,以增强机器人行走路线的连续性和可靠性。路径预处理的设计如下:查找原始路径 O 中的跳跃阶段,对起跳点和落地点之间对地形高度是否突变进行检测,当地形高度差均小于设定阈值则说明该段路径对应的地形高度无较大突变,四足机器人可以采用行走模式;对于必须使用跳跃才能通过的地形,则保留跳跃部分,然后继续完成后续路径情况的查询,遍历后得到路径 P 。上述过程在算法1中以伪代码表示。

算法1:路径预处理

输入: O 从KD-RRT-Connect得到的原始路径

输出: P 路径预处理后的结果

```

1  P = setFlightToStance(O, 0, n-1)
2  for i = 0 to n-1 do
3      if isTakeoffState(O, i) then
4          j = getLandIndex(O, i)
5          for k = i+1 to j do
6              m = getGroundHeight(O, k-1)
7              n = getGroundHeight(O, k)
8              if |m-n| > StanceLimit then
9                  setStanceToFlight(P, O, i, j)
10             end
11         end
12     end
13 end
  
```

2.2 离散空间采样与动态规划

通过路径预处理操作得到的路径 P ,为机器人提供了一条可到达目标点的路线,相比于原始路径 O 去除了不必要的跳跃过程,但与原始路径 O 相同的是,预处理得到的路径 P 也没有考虑路径周围一定区域内的地形。为了在此区域内找到一条地形起伏较小、比 P 更平滑的路线,同时保留KD-RRT-Connect基于概率采样算法的优点,对 P 周围一定范围内的地形进行离散采样,通过采样点处的地形有效性估计机器人的活动空间,由于限制了采样区域的大小,因此,路径 P 对目标点的可达性能够确保此范围内的路线变化不会改

变新路径的有解性。离散采样的设计如下:从路径 P 的行走阶段中等间隔选取路径点 P_s , 对 P_s 中每一个路径点 p_i , 沿其法线方向分别在两侧等间隔选取地形上的采样点, 并通过机器人是否能在该地形上有效站立这一条件判断该点的地形有效性, 最终得到地形有效的采样点 q_i , 将全部有效采样点的集合记为 Q , 此集合近似描述了机器人的活动空间。对于跳跃阶段, 起跳点和落地点一定是路径 P 中某一行走阶段的终点或起点, 因此, 路径 P 中的跳跃过程不会影响改进算法在行走阶段的优化结果。为方便后续描述, 本文假设预处理得到的路径 P 只包含行走阶段。通过对一定区域地形的离散采样近似描述机器人活动空间的过程在算法 2 中以伪代码表示。

算法 2: 生成地形有效的采样点集合

输入: P 路径预处理后的结果

输出: Q 地形有效采样点集合

```

1    $P_s = \text{getRefSamples}(P, \delta d)$ 
2    $k = \text{getSize}(P_s)$ 
3    $\text{range} = \max\text{Lateral}/\delta l$ 
4   for  $i = 0$  to  $k$  do
5        $\text{heading} = \text{getHeading}(P_s, i)$ 
6       if  $i = 0$  then
7            $X_{\text{sample}} = \text{getSamplePos}(P_s, \text{heading}, i, 0)$ 
8            $Q \leftarrow Q \cup \{X_{\text{sample}}\}$ 
9           continue
10      end
11      for  $j = -\text{range}$  to  $\text{range}$  do
12           $X_{\text{sample}} = \text{getSamplePos}(P_s, \text{heading}, i, j)$ 
13          if  $\text{isValidState}(X_{\text{sample}})$  then
14               $Q \leftarrow Q \cup \{X_{\text{sample}}\}$ 
15          end
16      end
17  end
```

2.2.1 离散采样代价函数

在限定活动空间后, 新路线的生成过程可以通过从集合 Q 中的元素 q_i 中选取采样点的方式表现。尽管四足机器人的足式结构可以更好地适应地形变化, 但地形的倾斜程度仍然会影响机器人在行进时的稳定性, 为了适应倾斜地形, 四足机器人需要调整身体姿态和步态, 导致更多的能量消耗。与 KD-RRT-Connect 相同, 改进算法使用相同的模型简化方法, 而细化具体落足位置和步态的调整交给底层控制而不在全局规划器中考虑。对于全局路径规划, 机器人仍需要避免过于倾斜

和复杂的地形, 以协调整体任务难度, 但只选择平坦路线的策略相当于舍弃了足式结构对地形的灵活适应性, 因此, 优化路线的选择需要同时考虑地形倾斜的变化和机器人姿态的变化。

地形倾斜程度可通过倾斜角度描述。对采样点集合 Q 中每一个地形有效的采样点 $q_{(i,j)}$, 可以使用下式计算该处位置地形的倾斜角度:

$$\theta_{(i,j)} = \arccos\left(\frac{\mathbf{n}_{q_{(i,j)}} \cdot \mathbf{z}}{|\mathbf{n}_{q_{(i,j)}}| \cdot |\mathbf{z}|}\right) \quad (4)$$

式中: $\theta_{(i,j)}$ 为采样点 $q_{(i,j)}$ 处地形的倾斜角度; $\mathbf{n}$ 为对应采样点的地形单位法向量, 可以通过所处地图网格的梯度分量计算得到; $\mathbf{z}$ 为世界坐标系下 z 方向的单位向量。

机器人的姿态变化可以通过 2 个采样点之间的距离变化、高度变化和机器人在采样点位置的转向变化近似描述。改进算法将四足机器人在 2 个采样点上的状态转移过程以代价函数的形式表示, 状态转移包含机器人的姿态变化和采样点的地形倾斜变化, 机器人在某一采样点的代价递推定义如下:

$$Cost = Cost_{\text{trans}} + Cost_{\text{pre}} \quad (5)$$

$$Cost_{\text{trans}} = Cost_{\text{dis}} + Cost_{\text{dz}} + Cost_{\text{ang}} + Cost_{\text{slope}} + Cost_{\text{goal}} \quad (6)$$

式中: $Cost_{\text{trans}}$ 为机器人的状态转移代价; $Cost_{\text{pre}}$ 为前一采样点的代价, 起点处的代价为 0; $Cost_{\text{dis}}$ 、 $Cost_{\text{dz}}$ 、 $Cost_{\text{ang}}$ 、 $Cost_{\text{slope}}$ 和 $Cost_{\text{goal}}$ 分别为距离代价、地形高度变化代价、转向变化代价、地形倾斜变化产生的代价和终点位置代价。对于每一项代价, 可以通过以下公式计算得到:

$$Cost_{\text{dis}} = w_{\text{dis}} \delta_d \quad (7)$$

$$Cost_{\text{dz}} = w_{\text{dz}} \delta_z \quad (8)$$

$$Cost_{\text{ang}} = w_{\text{pre}} \Delta\alpha_{\text{pre}} + w_{\text{p}} \Delta\alpha_{\text{p}} \quad (9)$$

$$Cost_{\text{slope}} = w_{\text{slope}} \Delta\theta \quad (10)$$

$$Cost_{\text{goal}} = w_{\text{goal}} \Delta l / \delta_n \quad (11)$$

式中: w_{dis} 为距离代价权重; w_{dz} 为高度变化权重; w_{pre} 为前采样点的转向变化权重; w_{p} 为当前采样点转向变化权重; w_{slope} 为地形倾斜变化权重; w_{goal} 为终点位置权重; δ_d 为采样点的欧氏距离; δ_z 为采样点在地形上的高度差; $\Delta\alpha_{\text{pre}}$ 和 $\Delta\alpha_{\text{p}}$ 分别为采样点连线方向与前采样点和当前采样点方向的变化角度; $\Delta\theta$ 为采样点地形倾斜角度的变化; Δl 为采样点与终点的横向索引差值; δ_n 为当前采样点纵向索引与终点索引的差值。由式(5)~(11)可知,

通过设置递推代价函数,将查找最优路线的过程转化为机器人在起点和终点之间的离散空间搜索粗解的过程,该过程会综合考虑距离、地形高度、转向角度、地形倾斜程度和与终点位置关系的变化。

2.2.2 动态规划

采样点集合 Q 中存在多个不同的解,机器人通过已设置的代价函数从其中找到代价最小的最优路线。从采样点集合 Q 中查找粗解 R 的过程是一个经典的动态规划问题。动态规划是求解决策过程最优化的一种方法,通过将待求解问题拆分为多个相同的子问题,分别对子问题求解并保存已解决子问题的结果,从而提高算法效率,避免了大量重复计算。该过程在算法3中以伪代码表示。

算法3:动态规划计算粗解

输入: Q 地形有效的采样点集合

输出: R 最优路线的粗解

```

1   minCost ← ∞
2   for each  $q_i \in Q$  do
3       for each  $q_{(i,j)} \in q_i$  do
4           for each  $q_{(i-1,k)} \in q_{i-1}$  do
5               Cost = calTotalCost( $q_{(i-1,k)}$ ,  $q_{(i,j)}$ )
6               if Cost < minCost then
7                   minCost = Cost
8                    $q_{(i-1,k)} \leftarrow \text{Parent}(q_{(i,j)})$ 
9               end
10          end
11          if hasParent( $q_{(i,j)}$ ) then
12              setCost( $q_{(i,j)}$ , minCost)
13          end
14      end
15  end
16   $r = \text{getEndPoint}(q_{n-1})$ 
17   $R \leftarrow \text{getPreToBegin}(r)$ 

```

基于设置的代价函数,机器人通过动态规划搜索到了一条更适合的路线,解决了机器人在地形选择上产生的决策问题。

2.3 三次B样条曲线拟合

粗解 R 是从集合 Q 中选择的离散点,其离散性受初始采样间隔的制约,并且存在疏密不均的情况,这种离散的粗解无法直接用于机器人导航,因此,要对粗解 R 做进一步处理,得到连续的曲线,在路径规划问题中通常使用曲线拟合的方式将离散点处理成光滑的曲线。B样条曲线具有局部性、连续性和凸包性的优点,非常适合将离散点

拟合成曲线。 k 阶B样条曲线的定义如下所示:

$$P(u) = \sum_{i=0}^n P_i B_{i,k}(u) \quad (12)$$

式中: P_i 为控制点; $B_{i,k}(u)$ 为基函数,基函数参考 deBoor-Cox 递推定义^[19]。

本文使用三次B样条曲线。粗解 R 中的点即为给定控制点,使用 Steinbeck 的 Tinspline 曲线库^[20]得到B样条曲线的离散点,记该离散点集合为 S 。但曲线拟合无法完全解决原本粗解 R 中点的稀疏程度不同的问题,为了曲线上的点均匀分布,需要对集合 S 进行三次样条插值,通过固定的间隔对 S 中的点进行插值处理,得到均匀分布的离散点集合 T 。

2.4 基于二次规划的平滑优化

曲线拟合和样条插值目的是实现离散点拟合并均匀分布,拟合对粗解 R 的平滑优化主要体现在局部范围,并不能替代整体路径的平滑处理过程。在全局路径规划中,整体路径的平滑优化可以转化为二次规划问题并使用求解器解决。当目标函数为二次型并且约束条件是线性时,可以将求函数最小值的问题转化为二次规划问题。二次规划问题可以用式(13)(14)的形式描述:

$$\min \left(\frac{1}{2} x^T H x + g^T x \right) \quad (13)$$

$$l \leq A x \leq u \quad (14)$$

式中: x 为待求解的变量; H 为 Hessian 矩阵; g 为梯度矩阵; A 为约束矩阵; l 和 u 分别为约束条件的下界和上界。

四足机器人作为一种地面机器人,路径的平滑过程主要表现在行走时的偏航角变化,因此,优化变量为离散点 $t_i(x_i, y_i)$ 。全局路径规划平滑优化的代价函数可以用下式表示:

$$Cost = Cost_{\text{smooth}} + Cost_{\text{length}} + Cost_{\text{dev}} \quad (15)$$

式中: $Cost_{\text{smooth}}$ 为平滑度; $Cost_{\text{length}}$ 为长度; $Cost_{\text{dev}}$ 为相对参考点的偏移量。路径的平滑程度可以通过使用相邻点的位置关系表示:

$$Cost_{\text{smooth}} = w_{\text{smooth}} \sum_{i=1}^{n-2} \left((x_{i-1} + x_{i+1} - 2x_i)^2 + (y_{i-1} + y_{i+1} - 2y_i)^2 \right) \quad (16)$$

式中: w_{smooth} 为平滑度的权重系数。长度的优化目标是使优化后的点尽可能紧凑分布,代价定义如下所示:

$$Cost_{\text{length}} = w_{\text{length}} \sum_{i=1}^{n-1} \left((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \right) \quad (17)$$

式中: w_{length} 为长度的权重系数。如果相对参考点的偏移量过大,则会丢失原有的几何信息,并且与样条插值后的点集产生较大的偏离可能导致机器人与地形发生碰撞。对偏移量的代价定义如下所示:

$$Cost_{\text{dev}} = w_{\text{dev}} \sum_{i=0}^{n-1} \left((x_i - x_{\text{ref}_i})^2 + (y_i - y_{\text{ref}_i})^2 \right) \quad (18)$$

式中: w_{dev} 为相对偏移量的权重系数。相对参考点的偏移量的边界条件通过下式表示:

$$x_{\text{ref}} - x_l \leq x \leq x_{\text{ref}} + x_u \quad (19)$$

$$y_{\text{ref}} - y_l \leq y \leq y_{\text{ref}} + y_u \quad (20)$$

待优化的目标函数能够记为标准的二次型形式,约束条件为线性,本文使用 OSQP 求解器^[21]对其进行求解,得到优化后的路径点集合 T_s 。该集合包含了新路线的二维坐标,对于机器人身体竖直方向的维度,本文选择与 KD-RRT-Connect 中相似的处理方法,但尽可能使机器人对自身质心高度的调节较小并限定在一定范围内。由于地形高度与位置的关系由地形环境本身决定,无法通过等式表达,因此,对该维度单独进行处理,具体做法是先通过集合 T_s 计算在机器人质心相对地面的高度不变时,质心在世界坐标系下的高度。该高度等于某一处地形表面的高度与机器人初始质心高度的和,如下所示:

$$z_{\text{ref}} = H(x, y) + H_{\text{normal}} \quad (21)$$

式中: $H(x, y)$ 为 (x, y) 位置的地形高度; H_{normal} 为初始状态机器人质心距离地面的高度。优化目标是机器人自身对质心高度变化的调节,使用下式进行定义:

$$Cost_z = \sum_{i=1}^{n-1} (z_i - z_{i-1})^2 \quad (22)$$

相对参考点的偏移量可表示为:

$$z_{\text{ref}} - z_l \leq z \leq z_{\text{ref}} + z_u \quad (23)$$

优化目标函数为二次型形式,约束条件为线性,同样使用 OSQP 求解,并将求解后的结果与 T_s 组合成三维坐标,得到最终路径 U 。

3 实验与分析

为验证改进算法的有效性,本文基于 ROS noetic 版本,在相同地图、相同起点和终点的初始条件下,使用 KD-RRT-Connect 和本文的改进算

法进行仿真对比实验。仿真的初始地图如图 3 所示。

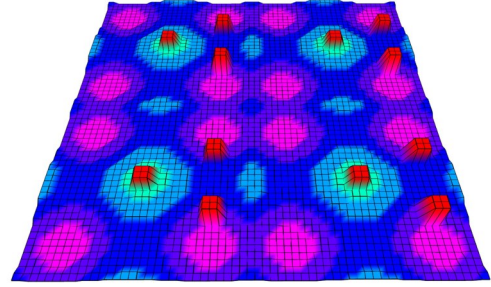


图 3 仿真地图环境

Fig. 3 Simulation environment

地图的环境参数如表 1 所示。由于 RRT 类算法具有随机性,本文为了保证规划结果的准确性和稳定性,对 2 种算法各运行 100 次。

表 1 环境参数

Table 1 Environment parameters

参数	数值/m
地图长度	12.4
地图宽度	12.4
地图网格分辨率	0.2
初始质心相对地面的高度	0.3

为了验证改进算法在不同复杂程度的地形上的效果,本文对初始地图的网格在高度上分别添加 0~5 cm 和 0~10 cm 的随机量,并分别在图 4(b) 和 (c) 2 幅更复杂的地图上运行 2 种算法各 100 次。

3.1 仿真结果分析

图 5 是 2 种算法在不同复杂程度地形中的部分运行结果。其中,黑色曲线代表原始算法 KD-RRT-Connect,白色曲线代表本文的改进算法。在(a-1)中,原始路径在一处凸起附近有较大的转向角度,而在改进算法中,路径绕过凸起并且转弯时的曲率较小,提高了机器人在行走阶段的稳定性。在(a-3)(b-1)(c-1)和(c-3)中,原始路径在经过 KD-RRT-Connect 算法长距离检测并剔除冗余路径点后仍然存在不平滑的部分,说明该处不平滑区域是不必要的跳跃模式导致,改进算法在判断地形没有较大突变后,采用行走的运动模式替换原有不必要的跳跃并最终优化为更平滑的路径,减少了多模态四足机器人在此部分的能耗并提高了执行任务的稳定性。其余结果则表现了改进算法对路径的平滑优化,并考虑了地形信息的新路线的选择,如(a-2)和(b-2),改进算法生成的路径选择沿浅坑的边缘行进而非进入浅坑,减

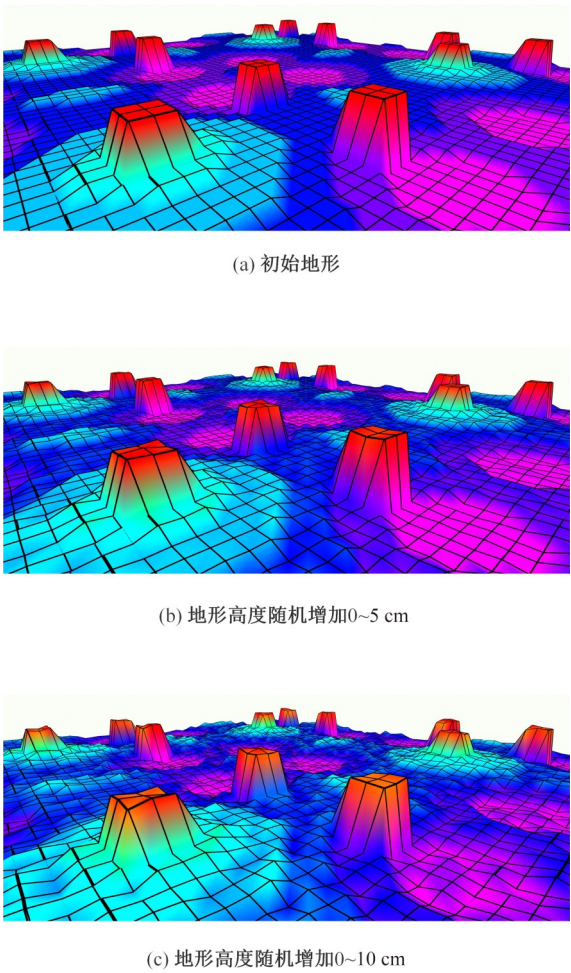


图 4 不同复杂程度的地形

Fig. 4 Terrain of different complexity

小了地形起伏程度的变化并缩短了路径长度。整体来看,改进算法生成的路径表现更平滑,转向角度更小,地形选择更平坦,提高了多模态机器人行进稳定性。

3.2 计算结果分析

本文以路径长度、机器人质心高度变化、地形起伏高度变化、地形倾斜变化和转向角度变化为评价指标,分别计算了 2 种算法在不同复杂程度

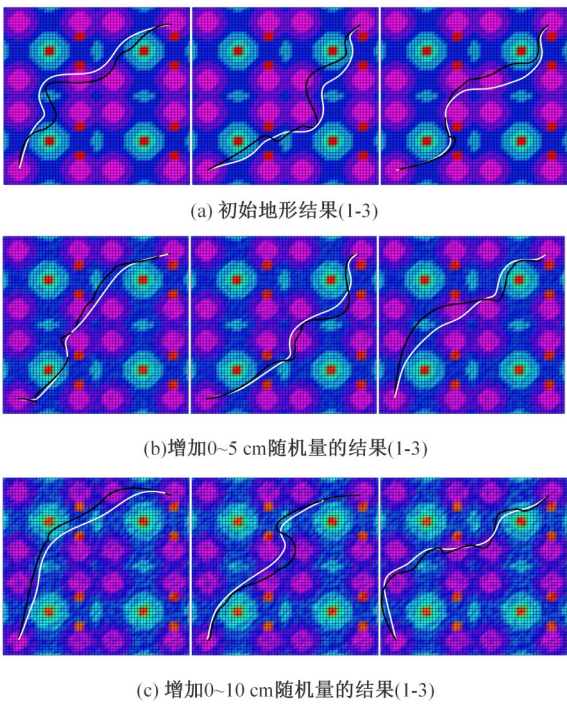


图 5 不同复杂程度的地形结果

Fig. 5 Results of different complexity terrain

的地形运行结果的平均值和方差,如表 2 和表 3 所示。

图 6 展示了 2 种算法在路径长度和地形起伏高度变化上的分布情况,从散点图中可以看出,尽管 RRT 类算法具有随机特性,散点分布不会特别集中,但从分布趋势可以看出,改进算法在这 2 个指标上都优于原始算法,平均路径长度更短,途径地形的高度变化也更小,有利于降低多模态四足机器人的能量消耗和提高行进的稳定性。

图 7 的(a)和(b)分别展示了 2 种算法在质心高度变化和地形起伏变化、地形倾斜变化和转向角度变化的对比结果,O 代表原始算法,U 代表改进算法。改进算法相比原始算法在平均路径长度上缩短了 5.8%,在质心变化这一指标上降低了 31.4%,显著减少了机器人在行进时对自身质心

表 2 结果的平均值

Table 2 Mean value of results

算法	路径长度	质心变化	地形起伏	地形倾斜	转向变化
KD-RRT-Connect	16.667	2.744	2.266	11.549	11.098
改进算法	15.703	1.882	1.962	10.238	4.141

表 3 结果的方差

Table 3 Variance of results

算法	路径长度	质心变化	地形起伏	地形倾斜	转向变化
KD-RRT-Connect	1.575	0.321	0.363	2.028	35.520
改进算法	0.896	0.231	0.341	2.268	2.269

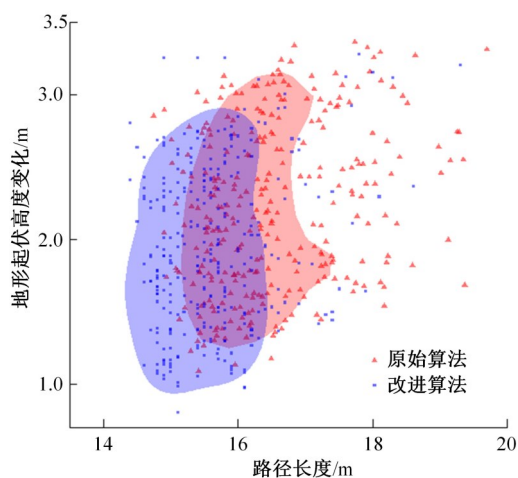
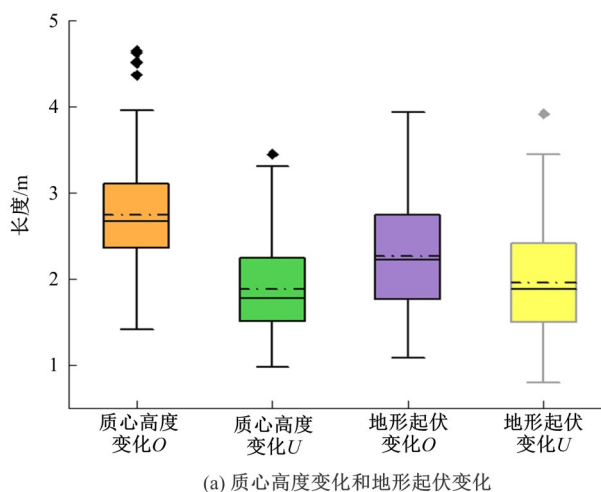
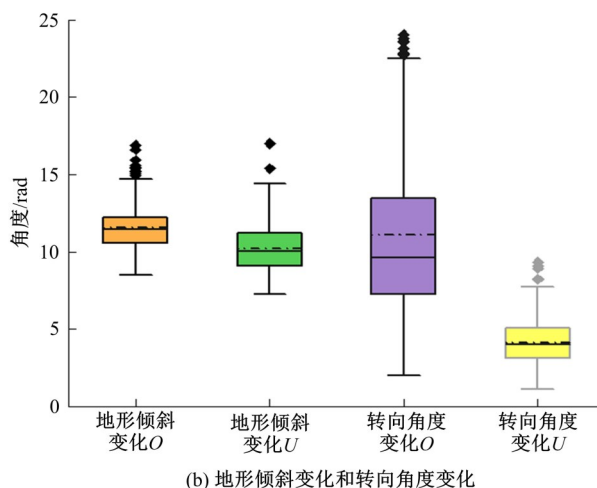


图6 结果分布图

Fig. 6 Distribution of results



(a) 质心高度变化和地形起伏变化



(b) 地形倾斜变化和转向角度变化

图7 结果对比

Fig. 7 Comparison of results

高度的调整,更节省能量。在地形方面,改进算法的地形起伏和地形倾斜相比原始算法分别降低了13.4%和11.4%,能够减少机器人在有起伏的地形上行走,并且地形倾斜变化也有所减缓,提高了

四足机器人行走时的安全性和稳定性。在转向变化上,改进算法相比原始算法降低了62.7%,显著改善了KD-RRT-Connect在规划时出现的路径转向次数较多和部分转向角度大的情况,改进算法的路径更平滑。方差结果显示,改进算法指标的方差除了倾斜程度略大于原始算法外,其余指标的方差均小于原始算法,说明在绝大多数情况下,改进算法的浮动较小,规划结果更稳定,保证了机器人行进时的稳定性。

4 结束语

本文针对目前多模态四足机器人在路径规划中存在不必要的跳跃和行走部分路径质量变化大的问题,提出了一种基于离散采样的解决方案。通过优化原始路径中多模态四足机器人的运动策略,离散采样获取周围地形信息,最终得到平滑优化的新路径。仿真结果表明,相比于原始算法,改进算法能够缩短路径长度,减少地形起伏变化和地形倾斜程度变化,降低机器人自身对质心高度的调节和平滑路径的急转部分,减小了总体转向角度,并且规划的结果浮动更小。本文方法能够优化多模态四足机器人的运动策略从而降低能耗,考虑地形信息并提高行进路线的稳定性,为具有多种运动模式的多模态机器人在实际环境中的全局路径规划提供参考。

参考文献:

- [1] 张秀丽,王琪,黄森威,等.一种多模型融合的仿猎豹四足机器人复杂运动控制方法[J]. 机器人, 2022, 44(6): 682-693.
Zhang Xiu-li, Wang qi, Huang Sen-wei, et al. A multi-model fusion based complex motion control approach for a cheetah-mimicking quadruped robot[J]. Robot, 2022, 44(6): 682-693.
- [2] 张帅帅,荣学文,李贻斌,等.崎岖地形环境下四足机器人的静步态规划方法[J]. 吉林大学学报:工学版, 2016, 46(4): 1287-1296.
Zhang Shuai-shuai, Rong Xue-wen, Li Yi-bin, et al. Static gait planning method for quadruped robots on rough terrains[J]. Journal of Jilin University(Engineering and Technology Edition), 2016, 46(4): 1287-1296.
- [3] 周坤,李川,李超,等.面向未知复杂地形的四足机器人运动规划方法[J]. 机械工程学报, 2020, 56(2): 210-219.

- Zhou Kun, Li Chuan, Li Chao, et al. Motion planning method for quadruped robots walking on unknown rough terrain[J]. *Journal of Mechanical Engineering*, 2020, 56(2): 210-219.
- [4] Nguyen Q, Powell M J, Katz B, et al. Optimized jump on the MIT cheetah 3 robot[C]//International Conference on Robotics and Automation(ICRA), Montreal, Canada, 2019: 7448-7454.
- [5] Jenelten F, He J, Farshidian F, et al. DTC: deep tracking control[J]. *Science Robotics*, 2024, 9(86): No. eadh5401.
- [6] 崔伟, 朱发证. 机器人导航的路径规划算法研究综述[J]. *计算机工程与应用*, 2023, 59(19): 10-20.
Cui Wei, Zhu Fa-zheng. Review of path planning algorithms for robot navigation[J]. *Computer Engineering and Applications*, 2023, 59(19): 10-20.
- [7] LaValle S. Rapidly-exploring random trees: a new tool for path planning[J]. *Research Report*, 1998: No. 9811
- [8] Kavraki L E, Svestka P, Latombe J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[J]. *IEEE Transactions on Robotics and Automation*, 1996, 12(4): 566-580.
- [9] Fankhauser P, Hutter M. A universal grid map library: implementation and use case for rough terrain navigation[J]. *Robot Operating System(ROS)——the Complete Reference*, 2016(1): 99-120.
- [10] Fankhauser P, Bloesch M, Gehring C, et al. Robot-centric elevation map with uncertainty estimates[M]//*Mobile Service Robotics*. London, UK: World Scientific, 2014: 433-440.
- [11] 张慧, 荣学文, 李贻斌, 等. 四足机器人地形识别与路径规划算法[J]. *机器人*, 2015, 37(5): 546-556.
Zhang hui, Rong Xue-wen, Li Yi-bin, et al. Terrain recognition and path planning for quadruped robot[J]. *Robot*, 2015, 37(5): 546-556.
- [12] Wermelinger M, Fankhauser P, Diethelm R, et al. Navigation planning for legged robots in challenging terrain[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), Daejeon, South Korea, 2016: 1184-1189.
- [13] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. *The international journal of robotics research*, 2011, 30(7): 846-894.
- [14] Norby J, Johnson A M. Fast global motion planning for dynamic legged robots[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, 2020: 3829-3836.
- [15] Kuffner J J, LaValle S M. RRT-connect: an efficient approach to single-query path planning[C]//IEEE International Conference on Robotics and Automation. FranciscoSan, USA, 2000: 995-1001.
- [16] Wellhausen L, Hutter M. Rough terrain navigation for legged robots using reachability planning and template learning[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021: 6914-6921.
- [17] Fernbach P, Tonneau S, Del Prete A, et al. A kinodynamic steering-method for legged multi-contact locomotion[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), Vancouver, Canada, 2017: 3701-3707.
- [18] 吴振宇, 刘小飞, 王义普. 基于DKRRT*-APF算法的无人系统轨迹规划[J]. *吉林大学学报: 工学版*, 2023, 53(3): 781-791.
Wu Zhen-yu, Liu Xiao-fei, Wang Yi-pu. Trajectory planning of unmanned system based on DKRRT* -APF algorithm[J]. *Journal of Jilin University(Engineering and Technology Edition)*, 2023, 53(3): 781-791.
- [19] De Boor C. On calculating with B-splines[J]. *Journal of Approximation Theory*, 1972, 6(1): 50-62.
- [20] Steinbeck M, Koschke R. Tinspline: a small, yet powerful library for interpolating, transforming, and querying nurbs, b-splines, and bézier curves[C]//IEEE International Conference on Software Analysis, Evolution and Reengineering(SANER), Honolulu, USA, 2021: 572-576.
- [21] Stellato B, Banjac G, Goulart P, et al. OSQP: an operator splitting solver for quadratic programs[J]. *Mathematical Programming Computation*, 2020, 12(4): 637-672.