

控制与决策

Control and Decision



凸优化与A*算法结合的路径避障算法

陈光荣, 郭盛, 王军政, 曲海波, 陈亚琼, 侯博文

引用本文:

陈光荣, 郭盛, 王军政, 等. 凸优化与A*算法结合的路径避障算法[J]. 控制与决策, 2020, 35(12): 2907–2914.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0351>

您可能感兴趣的其他文章

Articles you may be interested in

一种高匹配性的多层代价地图生成算法

A high matching layered costmap generation algorithm

控制与决策. 2020, 35(12): 2883–2888 <https://doi.org/10.13195/j.kzyjc.2018.0721>

无人飞行器航迹方案的VIKOR择优评价

Unmanned aerial vehicle path scheme optimal evaluation based–VIKOR

控制与决策. 2020, 35(12): 2950–2958 <https://doi.org/10.13195/j.kzyjc.2019.0415>

一种基于双编码遗传算法的机动微波接力网组网方法

Mobile microwave relay network construction method based on double coding genetic algorithm

控制与决策. 2020, 35(12): 2915–2922 <https://doi.org/10.13195/j.kzyjc.2019.0347>

融合长短时记忆机制的机械臂多场景快速运动规划

Multi-scene rapid motion planning combining with long and short time memory mechanisms for manipulators

控制与决策. 2020, 35(12): 2968–2976 <https://doi.org/10.13195/j.kzyjc.2018.1387>

基于免疫优化的平面Acrobot线性自抗扰鲁棒镇定

Robust stabilization of planar Acrobot using linear active disturbance rejection control with immune optimization

控制与决策. 2020, 35(12): 3053–3058 <https://doi.org/10.13195/j.kzyjc.2019.0289>

凸优化与 A* 算法结合的路径避障算法

陈光荣^{1†}, 郭盛¹, 王军政², 曲海波¹, 陈亚琼¹, 侯博文³

(1. 北京交通大学机械与电子控制工程学院, 北京 100044; 2. 北京理工大学自动化学院, 北京 100081; 3. 北京市轨道交通线路安全与防灾工程技术研究中心, 北京 100044)

摘要: 为提高足式移动机器人的避障能力和路径规划效率, 提出一种凸优化与 A* 算法结合的路径避障算法. 首先, 基于半定规划的迭代区域膨胀方法 IRI-SDP (iterative regional inflation by semi-definite programming), 通过交替使用两种凸优化算法快速计算出地面环境中无障碍凸多边形及其最大面积内切椭圆, 用于移动机器人的局部避障和任务动作规划; 然后, 结合经典的 A* 算法, 建立机器人局部和世界坐标系、机器人质心轨迹转换模型、碰撞模型和启发式代价函数, 在全局环境中寻找最优成本最小的路径; 最后, 通过仿真实验验证该算法的有效性.

关键词: 凸优化; A* 算法; 路径规划; 避障; 足式移动机器人; 无障碍空间

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0351

开放科学(资源服务)标识码(OSID):

引用格式: 陈光荣, 郭盛, 王军政, 等. 凸优化与 A* 算法结合的路径避障算法[J]. 控制与决策, 2020, 35(12): 2907-2914.



Convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm

CHEN Guang-rong^{1†}, GUO Sheng¹, WANG Jun-zheng², QU Hai-bo¹, CHEN Ya-qiong¹, HOU Bo-wen³

(1. School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China; 2. School of Automation, Beijing Institute of Technology, Beijing 100081, China; 3. Beijing Engineering and Technology Research Center of Rail Transit Line Safety and Disaster Prevention, Beijing 100044, China)

Abstract: To improve the obstacle avoidance ability and path planning efficiency of mobile legged robots, a convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm is proposed. Firstly, a method of iterative regional inflation by semi-definite programming (IRI-SDP) is presented to quickly compute out a large convex polygon of obstacle-free and its largest inscribed ellipse in the given ground environment through alternating two convex optimizations. The obstacle-free region is utilized for obstacle avoidance and task motion planning locally. Then, combining with the classical A-star algorithm via establishing the local and world coordinate system of mobile robots, the transfer model of the mass center of mobile robots, the impact model and the heuristics cost function, the optimal minimum-cost path in the global environment can be found. Finally, simulation results validate the effectiveness of proposed method.

Keywords: convex optimization; A-star algorithm; path planning; obstacle avoidance; mobile legged robot; obstacle-free space

0 引言

目前各式各样的机器人^[1-2], 如机械臂^[3]、双足机器人^[4-5]、四足机器人^[6]、无人车^[7]、无人机^[8]、水下自治机器人^[9]和智能移动体^[10]等, 已经成为工业界和学术界的研究热点. 如何在移动机器人所在的位置找到一块无障碍空间进行路径规划和任务动作规划是控制机器人不与障碍物发生碰撞的关键.

机器人的运动规划包含了步态规划、任务规划和

路径规划. 步态规划是基于关节空间的, 任务规划是基于操作空间的, 而路径规划是基于全局环境空间的. 常用的路径规划算法有 A* 算法^[11], RRT 算法^[12]和势场法^[13]等. A* 算法是一种静态路网中求解最短路径最有效的直接搜索方法, 也是许多其他问题的常用启发式算法. A* 算法是一种完备的算法, 它的优势在于它对解的捕获能力是完全的, 但是由此产生的缺点就是算法复杂度较大, 而且算法也难以运用

收稿日期: 2019-03-24; 修回日期: 2019-09-30.

基金项目: 中央高校基本科研业务费专项资金项目(2019JBM051); 北京市轨道交通线路安全与防灾工程技术研究中心开放研究基金项目(RRC201701); 国家自然科学基金项目(51875033).

责任编辑: 巩敦卫.

[†]通讯作者. E-mail: grchen@bjtu.edu.cn.

在动态移动障碍物环境下. 因此,改进的A*算法得到了广泛研究,如辛煜等在文献[14]提出的改进A*算法将可搜索邻域个数从离散的8个调整为无限个,因此可移动方向也从 $\pi/4$ 的整数倍变为连续任意方向,能够有效解决传统A*算法在栅格地图上进行路径规划时,求解得到的路径长度不是最短以及转折点较多的问题. 该算法虽然增加了计算量,但是能够得到更优更平滑的规划路径. 虽然缩小栅格也能得到更优更平滑的规划路径,但相比于缩小栅格,该方法有其优势和独到之处,故能应用到无人车系统中. 此外,A*算法与优化算法相结合的路径规划算法,如与蚁群^[15-16]、粒子群^[17]和萤火虫^[18]等算法相结合,也得到了深入研究以进一步提高实用性能.

通常,足式机器人从当前位置移动到目的地的同时还需要执行相应的移动动作和任务动作,例如探测、搜索、搬运、协作和人-机-环境交互等操作,在进行此类操作动作时需考虑地形和障碍物等因素影响,因此需要机器人路径规划与其本体全身运动规划相结合,路径规划需要为机器人本体全身运动规划留有足够空间. 针对以上问题,本文提出一种基于凸优化^[19]与A*算法相结合的路径规划方法. 该方法允许A*算法用大尺度网格规划出一个最优路径的大致趋势,提高效率,再利用IRI-SDP方法^[20-21]对移动机器人所在的位置求解出一块最大的凸多边形无障碍区域,以便机器人在该区域进行运动规划和避障处理. 如此一来,基于传统A*算法实现了机器人从初始位置到目的地的路径规划;凸优化寻找机器人当前位置周边的最大无障碍凸多边形,机器人在该最大无障碍凸多边形内便可进行相应的动作规划和落脚点选择,实现避障的同时完成移动和任务动作. 本文的主要创新点如下:

1) 找到了凸优化与A*算法相结合的思路 and 实现方法;

2) 在代价函数中引入无障碍凸多边形边缘的距离裕度 D ,惩罚过度靠近障碍物,避免发生碰撞.

1 凸优化与A*算法相结合的路径避障算法

本节首先介绍一种基于半定规划的迭代区域膨胀方法IRI-SDP^[20-21],是一种通过一系列凸优化来快速计算出地面环境中无障碍凸多边形和椭圆的新方法. 该算法可以搜寻一个最大的内切椭圆和把它从地面障碍物集合中分隔出来的线集合. 计算所得到的无障碍区域可用于移动机器人的路径规划和任务执行等. 该方法交替使用两种凸优化: 1) 利用二次规

划生成一个线集合在给定的障碍物集合中分割出一个无障碍凸多边形; 2) 利用半定规划在线集合分割的无障碍凸多边形中找到一个面积最大的椭圆. IRI-SDP的每次迭代都将重新定义凸多边形和椭圆来增加内切椭圆的面积,即移动机器人的可通行无障碍区域. 该算法可用于平面2维,空间3维和 N 维,算法的计算时间正比于障碍物数量. 计算速度足够快时,算法可用于具有移动障碍物环境,即引入障碍物的时间维度. 本文以四足机器人在地面环境的路径规划作为研究背景,讨论算法在平面2维上的应用. 注意该算法适用于机器人移动速度大于障碍物移动速度且算法计算周期大于机器人每个动作执行时间.

1.1 凸优化: IRI-SDP算法

将线集合围成的凸多边形用线性约束表示为

$$L = \{x | Ax \leq b\}. \quad (1)$$

其中: $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, m 为线集合元素个数, n 为维数,平面 $n = 2$.

参考单位圆的形式,将椭圆表示为

$$E(C, d) = \{x = C\tilde{x} + d | \|\tilde{x}\| \leq 1\}. \quad (2)$$

其中: $C \in \mathbf{R}^{n \times n}$; $x, \tilde{x}, d \in \mathbf{R}^n$.

由于椭圆的面积大小正比于 C 行列式的对数^[19],即 $\ln(\det C)$ 是凹函数,具有最大值,则在地面环境中寻找一个最大的内切椭圆和把它从障碍物集合中分隔出来的线集合的问题就是解决如下这样非凸优化的问题:

$$\max \ln(\det C).$$

$$\text{s.t. } a_j^T v_k \geq b_j, v_k \in O_j, j = 1, 2, \dots, N;$$

$$\sup_{\|\tilde{x}\| \leq 1} a_i^T (C\tilde{x} + d) \leq b_j, \forall i = [1, 2, \dots, N]. \quad (3)$$

其中: a_j^T 是 A 的第 j 行, b_j 是 b 的第 j 个元素, O_j 是第 j 个凸多边形障碍物的顶点集合(任何障碍物都可以用凸多边形框住), v_k 是障碍物的第 k 个顶点, N 是有效障碍物个数. 第1个约束使障碍物的所有顶点都位于线集合的一侧; 第2个约束使所求的椭圆上的所有的点都位于该线集合的另一侧. 满足以上两个约束即可保证椭圆与每一个障碍物都完全分隔. 为了避免直接求解非优化问题带来的困难,本文利用IRI-SDP算法先求出满足第1个约束的线集合,再求出满足第2个约束的最大椭圆,然后根据障碍物数量的递增进行逐步交替重新求解. 算法的目的是求解最大无障碍凸多边形 L 和内切椭圆 E ,满足 $E \in L$ 且 L 与障碍物集合 O 只可能在边界上有交集存在,基本流程如下.

```

% 初始化椭圆, 机器人初始位置和障碍物集合
 $C_0 = \xi I_{n \times n}, d_0 = q_0, O_j \in O, v_{jk} \in O_j, i = 0;$  %  $\xi$  任意小
% 进入迭代求解最优分隔线集合和最大内切椭圆
while  $\det(C_i - C_{i-1}) / \det C_{i-1} > e;$  % 误差限  $e$ 
    % 求解给定椭圆的最优分隔线集合
     $[A_{i+1}, b_{i+1}] = \text{separated\_lines}(C_i, d_i, O);$ 
    % 求解给定线集合的最大内切圆
     $[C_{i+1}, d_{i+1}] = \text{inscribed\_ellipse}(A_{i+1}, b_{i+1});$ 
     $i = i + 1;$  % 准备下次迭代
end
 $A_i, b_i, C_i, d_i;$  % 输出最大无障碍凸多边形和内切椭圆

```

1.1.1 求解给定椭圆的最优分隔线集合

求解优化的线集合就是在更进一步扩大内切椭圆面积的同时确保内切椭圆不与障碍物相交. 从几何角度上看, 就是找到一些与障碍物边界相切、与椭圆的扩张方向正交的分隔线.

给定椭圆式(2), 定义椭圆 E 的标准扩张形式为

$$E_\tau(C, d) \equiv \{x = C\tilde{x} + d \mid \|\tilde{x}\| \leq \tau\}, \tau \geq 1. \quad (4)$$

为了寻找障碍物 O_j 上离椭圆最近的点, 可将问题转化为找到一个合适的 τ 满足

$$\begin{aligned} \tau^* &= \arg \min_{\tau} \tau; \\ \text{s.t. } E_\tau \cap O_j &\neq \emptyset. \end{aligned} \quad (5)$$

定义 x^* 为 $E_\tau \cap O_j$ 的交点, 即可计算线集合 $a_j^T x^* = b_j$ 是正交于 E_τ 且穿过 x^* , 因此该超平面分隔 E_τ 和 O_j . 由于 $E \in E_\tau, \tau \geq 1$, 该超平面也分隔 E 和 O_j . 该方法每次迭代都要为一个障碍物求解一条分隔线, 在障碍物数目很多的情况下需要大量的计算, 实际应用中没有必要为每个障碍物都计算其分隔线, 这就需要去掉冗余的障碍物.

1.1.2 求解离椭圆最近的点

代替直接求解式(5)的 τ 值, 可以将问题简化成单个更容易解决的最小距离规划问题. 定义椭圆 E 和凸形障碍物 O_j 的顶点集合 $v_{j1}, v_{j2}, \dots, v_{jp}$, 根据椭圆定义式(2)将该问题映射到单位圆上, 即

$$\begin{aligned} &\text{椭圆空间} \\ E(C, d) &\equiv \{x = C\tilde{x} + d \mid \|\tilde{x}\| \leq 1\} \\ O_j &= \text{凸多边形}(v_{j1}, v_{j2}, \dots, v_{jp}) \\ v_{jk} &= C\tilde{v}_{jk} + d \end{aligned}$$

圆空间

$$\begin{aligned} \tilde{E}(C, d) &\equiv \{\tilde{x} \mid \|\tilde{x}\| \leq 1\} \\ \tilde{O}_j &= \text{凸多边形}(\tilde{v}_{j1}, \tilde{v}_{j2}, \dots, \tilde{v}_{jp}) \\ \tilde{v}_{jk} &= C^{-1}v_{jk} - d \end{aligned}$$

只要找到变换后的障碍物 $\tilde{O}_j$ 上离原点最近的点, 然后利用 $C\tilde{v}_{jk} + d$ 变换即可求得原问题离障碍物 O_j 最近的点. 该问题可表述成

$$\begin{aligned} &\arg \min_{\tilde{x}, \alpha \in \mathbf{R}^p} \|\tilde{x}\|^2; \\ &\text{s.t. } [\tilde{v}_{j1} \ \tilde{v}_{j2} \ \dots \ \tilde{v}_{jp}] \alpha = \tilde{x}, \\ &\sum_{i=1}^p \alpha_i = 1, \alpha_i \geq 0. \end{aligned} \quad (6)$$

其中 $\tilde{x}$ 是 $\tilde{v}_{jk}$ 的凸组合, 该问题的解就是离原点最近的点. 虽然这是一个二次规划问题, 但可以转成高效求解的非负约束最小距离问题^[22]. 可以极大地减小计算时间. 该优化得到的点 $\tilde{x}^*$, 经过原映射 $x^* = C\tilde{x}^* + d$, 即为障碍物 O_j 离椭圆 E 最近的点.

1.1.3 求解垂直正交线

求解垂直正交线最简单的方式就是考虑椭圆式(2)的反表达式

$$E(C, d) = \{x \mid (x - d)^T C^{-1} C^T (x - d) \leq 1\}. \quad (7)$$

通过计算椭圆在 $x^* = C\tilde{x}^* + d$ 点处函数的梯度即可求得一个垂直于椭圆的向量.

$$\begin{aligned} a_j &= \frac{d}{dx} [(x - d)^T C^{-1} C^T (x - d)]|_{x^*} = \\ &2C^{-1} C^T (x^* - d). \end{aligned} \quad (8)$$

一旦求得 a_j , 由垂直正交线经过点 x^* , 可得

$$b_j = a_j^T x^*. \quad (9)$$

1.1.4 去掉冗余的障碍物

在有很多障碍物的地面环境中, 根据 1.1.2 节和 1.1.3 节得到的大部分分隔线对于寻找一个无障碍椭圆凸多边形区域是没有用的. 如图1和图2所示, 在每次迭代中, 只有几条分隔线是需要被用来完全分隔障碍物和椭圆的. 通过消除冗余的分隔线, 可以极大地提高寻找最大内切椭圆的效率.

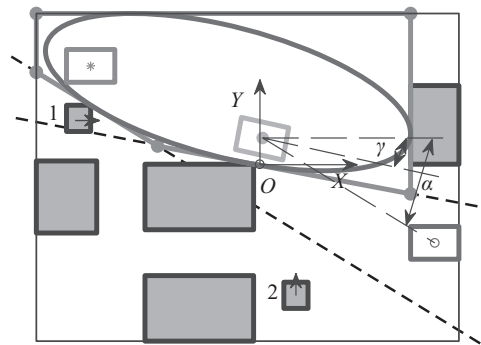


图1 地面仿真环境

对于一个给定的障碍物 O_j , 计算得到一组 a_j, b_j 对所有的 $x \in O_j$ 有 $a_j^T x \geq b_j$, 即得到一条分隔线 $b_j = a_j^T x^*$ 分离障碍物 O_j 和椭圆 E , 然后可以遍历其他的障碍物 $O_k (k \neq j)$, 判定其所有的顶点是否满足 $a_j^T v_k \geq b_j$. 如果满足, 则障碍物 $O_k (k \neq j)$ 和椭圆 E 同样被上面计算的分隔线分隔; 如果不满足, 则计算出另一条分隔线来分隔障碍物 $O_k (k \neq j)$ 和椭圆 E .

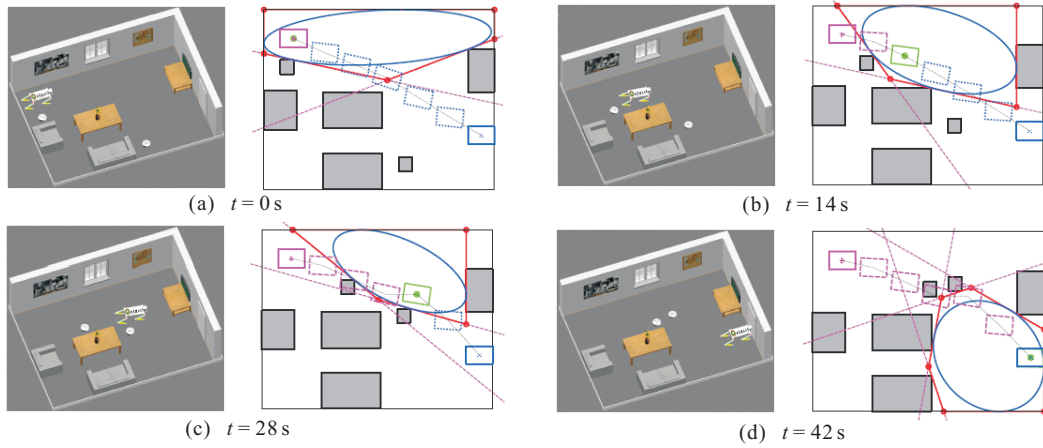


图2 不同时刻仿真结果

为了进一步提高效率,在计算开始选取一个具有离椭圆最近的顶点的障碍物,这样计算得到的分隔线容易将椭圆和远处的障碍物分隔,然后由近及远地计算直到所有的障碍物集合和椭圆都被求得的分隔线集合分隔开来。

结合 1.1.1~1.1.4 小节,给定椭圆 E (式 (2)) 和凸多边形障碍物集合 O ,求解 A 和 b 定义的分隔线集合能正切于椭圆 E 的扩张集合式 (4),且满足 $\{x|Ax \leq b\} \cap O = \emptyset$. 该程序的主要步骤如下。

```
function separated_lines(C, d, O)
% 初始化障碍物处理集合
O_useless =  $\emptyset$ ; O_remaining = O; i = 1;
while O_remaining  $\neq \emptyset$ 
% 求解离椭圆最近的点
x* = closest_ponit_in_obstacles(C, d, O);
% 求解垂直正交线
[a_i, b_i] = tangent_line(C, d, x*);
% 去掉冗余的障碍物
for all O_j  $\in$  O_remaining
if a_j^T v_jk  $\geq$  b_j,  $\forall v_jk \in O_j$ 
O_remaining = O_remaining \ O_j;
O_useless = O_useless  $\cup$  O_j;
end
end
i = i + 1; % 准备下次迭代
end
% 输出给定椭圆的最优分隔线集合
A =  $\begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \end{bmatrix}$ , b =  $\begin{bmatrix} b_1 \\ b_2 \\ \vdots \end{bmatrix}$ .
```

需注意的是,假设障碍物个数为 N ,当 $N = 0$ 时, separated_lines 不进入 while 循环;当 $N = 1$ 时, separated_lines 进入 while 循环 1 次;当 $N \geq 2$ 时, separated_lines 进入 while 循环次数由障碍物分布决定,最小循环次数为 1,最大循环次数为 N . 如图 1 和图 2 所示,仿真环境总共有 6 个障碍物(阴影区域),第

1 排只用到 2 个障碍物(进入循环 2 次);第 2 排和第 3 排都只用到 3 个障碍物(进入循环 3 次);第 4 排只用到 4 个障碍物(进入循环 4 次). 因此去掉冗余的障碍物算法的时间复杂度 $\leq N$.

1.1.5 求解给定凸多边形的最大面积内切圆

计算多边形的最大面积内切圆已有很好的研究,高效的实际应用算法也很容易找到. 以一个单位圆的形式来定义椭圆 E (式 (2)),椭圆的面积正比于 C 的行列式大小. 求解多边形 $L = \{x|Ax \leq b\}$ 内的最大面积内切圆问题可以表述为

$$\begin{aligned} \max_{C,d} \quad & \log(\det C); \\ \text{s.t.} \quad & \sup_{\|\tilde{x}\| \leq 1} a_i^T (C\tilde{x} + d) \leq b_i, \forall i = [1, 2, \dots, N], \\ & C \geq 1. \end{aligned} \quad (10)$$

参考文献[6],约束可写成不包含 $\tilde{x}$ 的形式,即

$$\begin{aligned} \max_{C,d} \quad & \log(\det C); \\ \text{s.t.} \quad & \sup_{\|\tilde{x}\| \leq 1} \|a_i^T C\| + a_i^T d \leq b_i, \forall i = [1, 2, \dots, N], \\ & C \geq 1. \end{aligned} \quad (11)$$

该问题即为凸优化问题,可以利用 Mosek 中包含的 Ben-Tal and Nemirovski 提出的半定优化和圆锥二次约束优化来计算该椭圆^[23].

1.2 A*算法与避障

A*算法是一类适用于全局环境信息已知的路径规划方法,A*算法引入了启发式函数,因此A*算法亦称启发式算法. 本文应用A*算法的核心思想为:在局部坐标质心轨迹规划的基础上进行以初始支撑足为起始点的启发式搜索,再结合求得的最大无障碍凸多边形,通过机器人质心轨迹转换模型和环境模型在局部坐标系中进行碰撞检测,最后根据不同的质心轨迹转换损耗构造启发式函数,通过A*算法在世界坐

标系中规划出机器人质心轨迹^[14].

1.2.1 机器人模型

本文中的机器人模型为四足机器人,如图2所示,每条腿3个自由度,简化表示为 $q \in \mathbf{R}^{12}$.

1.2.2 地面环境模型

在世界坐标系中,机器人第 $n-1$ 步后的质心轨迹可表示为

$$X_{n-1} = (x_{n-1}, y_{n-1}, h_{n-1})^T, \quad (12)$$

则机器人第 n 步的质心轨迹可表示为

$$\begin{aligned} X_n &= X_{n-1} + \Delta X_{n-1}, \\ \Delta X_{n-1} &= (\Delta x_{n-1}, \Delta y_{n-1}, \Delta h_{n-1})^T. \end{aligned} \quad (13)$$

其中 $x_{n-1}, y_{n-1}, h_{n-1}$ 和 $\Delta x_{n-1}, \Delta y_{n-1}, \Delta h_{n-1}$ 分别表示第 $n-1$ 步后机器人在世界坐标系中的位置(前侧向坐标和质心高度)和到第 n 步要产生的位置变化量, ΔX_{n-1} 表示基于第 $n-1$ 步后机体坐标系下机器人质心下一步可以到达的位置,显然 $\|\Delta X_{n-1}\| \leq S_{\max}$, $S_{\max}$ 为机器人最长步长,由其运动空间决定.

1.2.3 机器人质心轨迹转换模型

如果机器人第 n 步后的机体坐标系下支撑点为 $S_n = (0, 0, 0, 0, 0)$,则落足点为

$$\begin{aligned} T_{ni} &= (\Delta x_{ni}, \Delta y_{ni}, \Delta h_{ni}, \|\Delta X_{ni}\|, \Delta \gamma_{ni}, d_{ni})^T, \\ i &= 1, 2, \dots, M. \end{aligned} \quad (14)$$

其中: $\Delta \gamma_{ni}$ 为机器人第 n 步后需要产生的偏转角, d_{ni} 为机器人离无障碍物凸多边形边缘的最小距离, M 为可行落足点的数量.

1.2.4 碰撞模型

根据求解得到的线集合 $Ax = b$,四足机器人可以简化成如图2所示的长方形.在机体坐标系下可以计算得到机器人各个顶点的坐标为

$$\begin{aligned} X_B = \left\{ x \mid x \in \left\{ \left(\frac{L}{2}, \frac{W}{2} \right), \left(\frac{L}{2}, -\frac{W}{2} \right), \right. \right. \\ \left. \left. \left(-\frac{L}{2}, \frac{W}{2} \right), \left(-\frac{L}{2}, -\frac{W}{2} \right) \right\} \right\}. \end{aligned} \quad (15)$$

经旋转变换平移可得机器人在世界坐标系中的坐标

$$X_W = X + T(\gamma)X_B = X + \begin{bmatrix} \cos \gamma & \sin \gamma \\ -\sin \gamma & \cos \gamma \end{bmatrix} X_B. \quad (16)$$

其中: L 和 W 分别为机器人的长和宽, X 和 γ 分别为机器人质心位置和方位角.如果有 $AX_W \leq b$,则机器人不会与障碍物发生碰撞.

1.2.5 构建代价函数

根据A*算法,选择代价函数

$$f(n) = g(n) + h(n). \quad (17)$$

其中: $f(n)$ 是从初始节点 X_s 经由节点 X_n 移动到目标节点 X_t 的移动损耗, $g(n)$ 是从初始节点 X_s 移动到节点 X_n 的实际代价, $h(n)$ 是从节点 X_n 到目标节点 X_t 的估计移动代价.具体可写为

$$\begin{aligned} g(n) &= w_{g1}\|X_{n-1} - X_s\| + w_{g2}\|\Delta x_{n-1}\| + w_{g3}\|\Delta y_{n-1}\| + \\ &w_{g4}\|\Delta h_{n-1}\| + w_{g5}\Delta \gamma_{n-1} + w_{g6}\delta_{n-1}; \\ h(n) &= w_{f1}\|X_t - X_n\| + w_{f2}\alpha; \\ \delta_{n-1} &= \begin{cases} D + \frac{1}{d_{n-1}}, & d_{n-1} \leq D; \\ D, & d_{n-1} > D. \end{cases} \end{aligned} \quad (18)$$

其中: X_s 和 X_t 分别为起始点和目标点的世界坐标位置; D 为无障碍凸多边形边缘的距离裕度,惩罚过度靠近障碍物,避免发生碰撞; α 为机器人朝向与目标所在方位的夹角,如图1所示.

整个算法流程如下.注意机器人寻找的可达区域只在无障碍凸多边形中进行,可以一定程度减少A*算法的计算时间.

```

world_init( $X_s, X_t$ ); % 初始化起始节点和目标节点
open_table=[]; close_table=[]; % 定义空 open 和 close 表
while  $X_s \sim X_t \mid X_{min} \sim X_t$ ; % 是否为目标节点
     $X_s = X_{min}$ ; % 将此节点定为新的起始节点
    % 把初始节点加入 open 表
    open_table=[open_table,  $X_s$ ];
    grid:  $X_s + \Delta X \rightarrow X$ ; % 网格法寻找可达区域
     $AX_W \leq b \rightarrow T_i = X - X_s$ ; % 碰撞检测
    % 将可行的落足点加入 open 表
    open_table=[open_table,  $X_s + T_i$ ];
    % 将起始节点作为他们的父节点
    father_node: open_table \  $X_s \rightarrow X_s$ ;
    % 在 open 表删除起始节点
    open_table=open_table \  $X_s$ ;
    % 再将其加入 close 表
    close_table=[close_table,  $X_s$ ];
    % 搜寻 open 表中  $f(n)$  最小的节点  $X_{min}$ 
     $f(n) : \text{open\_table} \rightarrow X_{min}$ ;
end
% 从目标节点沿其父节点移动到起始节点
father_node:  $X_t \rightarrow X_s$ ; % 即得到规划轨迹

```

1.3 算法收敛性

当该方法仍保证优化的收敛性时,IRI-SDP 不保证在地面环境中找到一个无障碍最大椭圆,仅保证在给定凸多边形内找到一个最大面积内切圆.由于分隔线集合 L 与扩张的椭圆 E_r 正切,原椭圆 E 总是被包含在分隔线集合 L 组成的多边形内.此外,求解椭圆最大化利用的SDP算法求解的是全局最优,即得到的椭圆面积不会小于 E .如若不然,将会有更大面积的可行解存在,这与SDP的全局最优相悖.只要给

定的地面环境有界,椭圆的面积相对于整个地面面积就有上界,即最大值. 由于椭圆的面积有上界且每次迭代都会单调递增,其最终会收敛到一个最终值,虽然算法不保证迭代次数为多少才能保证收敛,但可人为设定 IRI-SDP 迭代次数上限以满足动态优化的时间窗.

A* 算法是基于动态规划的典型图搜算法,主要依据式(17)的代价函数对遍历节点排序,引导搜索到达目标,即保证收敛. 当式(17)的 $g(n)$ 相比 $h(n)$ 无穷大时,该算法类似广度优先算法(BFS);而无穷小时,该算法类似深度优先算法(DFS). A* 算法是一种最快的最有效的直接搜索方法,算法中的距离估算值与实际值越接近,最终搜索速度越快.

本文中,IRI-SDP 算法保证了局部地面环境无障碍区域搜索的收敛性,A* 算法保证了全局路径规划的最优化.

2 仿真验证

本文利用 Webots 搭建地面环境,联合 Matlab 做控制器进行仿真验证. 仿真主要集中于路径规划算法,具体的四足机器人控制(如高精度位置,节能,柔顺等)参见文献[24-30]. 如图1所示,四足机器人起始位置为 $(-2.8, 1.5)$ m,目标位置为 $(2.9, -1.2)$ m,机器人平均速度大于 0.2 m/s,单个动作/单步执行时间为 0.3 s,远大于算法计算周期,算法单次迭代时间小于 1.2 ms,设定最大迭代次数 $N = 100$. 阴影部分为障碍物,其中移动障碍物1和移动障碍物2分别沿 x 轴和 y 轴方向以 0.1 m/s 速度运动. 仿真参数如表1所示,仿真结果简化如图2所示,图中星号为起始位置,叉标记为目标位置,实心圆为机器人质心位置,虚线框为机器人走过路径,点线框为规划路径,虚线为简化质心轨迹,可以看出:

1) 凸优化并不是对环境中的所有障碍物都进行最优分隔线的求取,而是对距离移动机器人较近一些的障碍物,包括移动障碍物求解有效的分隔线集合,从而组成无障碍的凸多边形区域;

2) 机器人的运动轨迹规划都在该无障碍凸多边形区域内进行,故机器人整体都落在该区域内;

3) 过度近距离惩罚裕度 D 的存在,使得机器人与障碍物会保持一定的距离,而不是紧挨着擦肩而过;

4) 如四足动物一样,由于侧向感知能力远弱于前向,导致了机器人前向加转向的总运动代价与侧向运动代价在一个数量级上,故机器人不会只依靠前向和侧向的运动来达到目标位置,还伴有转向运动;

5) 机器人的局部凸优化算法和全局路径规划算法都是实时计算的,由于移动障碍物的存在,会干扰到原有规划的路径,机器人需要实时规划出新的全局最优路径,以达到目标位置. 所以,机器人最开始的规划轨迹与其走过的轨迹存在较大差异.

表1 仿真参数

参数	值	参数	值
w_{g1}	1 m^{-1}	w_{f1}	1 m^{-1}
w_{g2}	100 m^{-1}	w_{f2}	1 rad^{-1}
w_{g3}	200 m^{-1}	L	1.1 m
w_{g4}	1 m^{-1}	W	0.6 m
w_{g5}	$1\,000\text{ rad}^{-1}$	D	0.2 m
w_{g6}	10 m^{-1}	N	100

为了量化凸优化和距离裕度 D 的优势,本文比较了有无结合凸优化时和结合凸优化下不同距离裕度 D 时四足机器人离障碍物的最小距离,如图3所示,其中 0 ~ 17 s 主要为与移动障碍物1的距离,17 ~ 30 s 主要为与移动障碍物2和中间桌子的最小距离,30 ~ 38 s 主要为与移动障碍物1的距离,38 ~ 42 s 主要为与门的距离. 同时还比较了不同距离裕度 D 对实际移动损耗 $f(n)$ 的影响,如图4所示. 由拟合曲线趋势可知,在图1给定的仿真环境下,距离裕度在 $D = 0.4$ m 附近,移动损耗最小,主因为 $D < 0.4$ m 时,容易碰上障碍物; $D > 0.4$ m 时,移动距离变大. 注意四足机器人移动速度具有波动性且还存在上下左右的晃动,故 D 值相对较大. 总的来说,结合凸优化之后机器人可以在不提高移动损耗的基础上更高效地避障,并且移动机器人与障碍物的距离裕度相对特定的环境都有着特定的最优值.

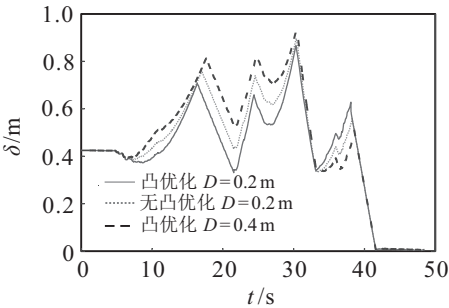


图3 对比仿真实验

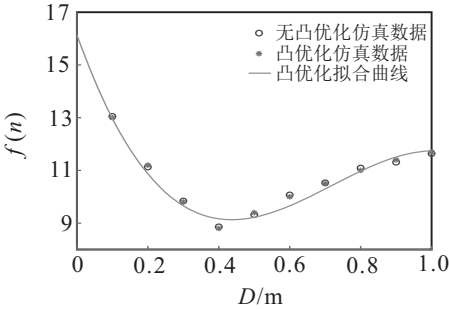


图4 距离裕度与实际移动损耗的关系

此外,本文通过与文献[14]中所提出的改进 A* 算法进行对比分析,突出本文算法的优势和适用场景. 图 5 为本文算法和文献[14]算法在起始点上的规划结果图,其规划结果数据对比如表 2 所示. 文献[14]算法规划的轨迹更加平滑,曲率半径更小,转折点更少,路径更短,但与障碍物最小距离较小,易发生碰撞,且算法计算时间相对较长. 本文算法加入了距离裕度 D 惩罚过度靠近障碍物,因而具有较大的与障碍物最小距离,且算法计算时间相对较短,对控制器的计算性能要求较低,更适于对负载要求较高的足式机器人. 况且足式机器人本身行走较慢,落脚点离散不连续,不必要求连续平滑曲率较小的路径规划,为了避免与障碍物碰撞也不严格要求规划路径最短,故本文算法在足式机器人领域具有一定优势,而文献[14]算法在无人车领域用途更大.

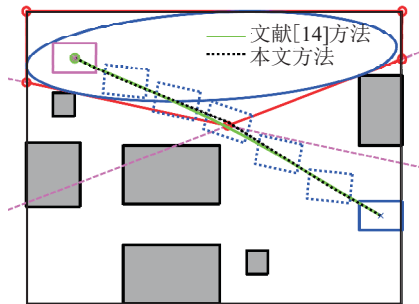


图 5 本文算法和文献[14]算法在起始点上的规划结果

表 2 本文算法和文献[14]算法规划结果数据对比

指标	本文算法	文献[14]算法
转折点个数	3	0
路径长度	6.327 5 m	6.314 6 m
平均曲率	$0.040\ 3\ \text{m}^{-1}$	$0.009\ 4\ \text{m}^{-1}$
与障碍物最小距离	0.336 7 m	0.254 1 m
算法计算时间比	1	2.137

3 结 论

本文针对足式移动机器人在复杂环境中的高效避障和路径规划问题,提出了一种凸优化与 A* 算法相结合的路径避障算法,并仿真验证了该算法的有效性. 本文的主要贡献如下: 1) 交替使用两种凸优化算法来快速计算出地面环境中无障碍凸多边形及其最大面积内切椭圆,用于移动机器人的局部避障动作规划. 2) 结合经典的 A* 算法,建立机器人局部和世界坐标系,机器人质心轨迹转换模型,碰撞模型和启发式代价函数在全局环境中寻找最优成本最小的路径. 3) 联合 Webots 和 Matlab 对本文提出的算法进行仿真,仿真结果验证了所提出算法的有效性.

本文的研究可拓展到更多维空间,可运用到其他类足式移动机器人,如双足机器人可落脚区域等,未来的研究还将集中在机器人的优化学习和自主决策

等.

参考文献(References)

[1] 杨健健,唐至威,王子瑞,等. 基于 VSPSO 和 A-G 网络的掘进机动态路径规划[J]. 控制与决策, 2019, 34(3): 642-648.
(Yang J J, Tang Z W, Wang Z R, et al. Dynamic path planning of roadheader based on VSPSO and A-G net[J]. Control and Decision, 2019, 34(3): 642-648.)

[2] 张晓平,阮晓钢,肖尧,等. 基于内发动机机制的移动机器人自主路径规划方法[J]. 控制与决策, 2018, 33(9): 1605-1611.
(Zhang X P, Ruan X G, Xiao Y, et al. Mobile robot autonomous path planning method based on intrinsic motivation mechanism[J]. Control and Decision, 2018, 33(9): 1605-1611.)

[3] 陈钢,郭雯,贾庆轩,等. 基于运动学模型重构的单关节故障机械臂容错路径规划[J]. 控制与决策, 2018, 33(8): 1436-1442.
(Chen G, Guo W, Jia Q X, et al. Fault tolerant path planning method for manipulator with single joint failure based on kinematics model reconstruction[J]. Control and Decision, 2018, 33(8): 1436-1442.)

[4] Chen G R, Wang J Z, Wang L P. Gait planning and compliance control of a biped robot on stairs with desired ZMP[J]. IFAC Proceedings Volumes, 2014, 47(3): 2165-2170.

[5] Chen G R, Wang J Z, Wang L P, et al. Design and simulation of a hydraulic biped robot[C]. The 32nd Chinese Control Conference. Xi'an, 2013: 4244-4249.

[6] Winkler A, Havoutis I, Bazeille S, et al. Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots[C]. 2014 IEEE International Conference on Robotics and Automation (ICRA). Hong Kong: IEEE, 2014: 6476-6482.

[7] Li J Q, Deng G Q, Luo C W, et al. A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems[J]. IEEE Transactions on Vehicular Technology, 2016, 65(12): 9585-9596.

[8] 张毅,孟启源,杨秀霞. 基于双旋 Lyapunov 矢量场的无人机避障算法[J]. 控制与决策, 2018, 33(8): 1514-1522.
(Zhang Y, Meng Q Y, Yang X X. Obstacle avoidance algorithm for UAV based on bi-rotation Lyapunov vector fields[J]. Control and Decision, 2018, 33(8): 1514-1522.)

[9] Cui R X, Li Y, Yan W S. Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 46(7): 993-1004.

[10] 张玮,马焱,赵捍东,等. 基于改进烟花-蚁群混合算法的智能移动体避障路径规划[J]. 控制与决策, 2019, 34(2): 335-343.
(Zhang W, Ma Y, Zhao H D, et al. Obstacle avoidance path planning of intelligent mobile based on improved fireworks-ant colony hybrid algorithm[J]. Control and Decision, 2019, 34(2): 335-343.)

- [11] Duchoň F, Babinec A, Kajan M, et al. Path planning with modified a star algorithm for a mobile robot[J]. *Procedia Engineering*, 2014, 96: 59-69.
- [12] 刘佳顺, 刘检华, 张之敬, 等. 基于任意时间RRT算法的三维自动布线技术[J]. *机械工程学报*, 2016, 52(13): 156-165.
(Liu J S, Liu J H, Zhang Z J, et al. Anytime RRT based cable automatic routing under three-dimensional environment[J]. *Journal of Mechanical Engineering*, 2016, 52(13): 156-165.)
- [13] 朱毅, 张涛, 宋靖雁. 未知环境下势场法路径规划的局部极小问题研究[J]. *自动化学报*, 2010, 36(8): 1122-1130.
(Zhu Y, Zhang T, Song J Y. Study on the local minima problem of path planning using potential field method in unknown environments[J]. *Acta Automatica Sinica*, 2010, 36(8): 1122-1130.)
- [14] 辛煜, 梁华为, 杜明博, 等. 一种可搜索无限个邻域的改进A*算法[J]. *机器人*, 2014, 36(5): 627-633.
(Xin Y, Liang H W, Du M B, et al. An improved A* algorithm for searching infinite neighbourhoods[J]. *Robot*, 2014, 36(5): 627-633.)
- [15] 王晓燕, 杨乐, 张宇, 等. 基于改进势场蚁群算法的机器人路径规划[J]. *控制与决策*, 2018, 33(10): 1775-1781.
(Wang X Y, Yang L, Zhang Y, et al. Robot path planning based on improved ant colony algorithm with potential field heuristic[J]. *Control and Decision*, 2018, 33(10): 1775-1781.)
- [16] 游晓明, 刘升, 吕金秋. 一种动态搜索策略的蚁群算法及其在机器人路径规划中的应用[J]. *控制与决策*, 2017, 32(3): 552-556.
(You X M, Liu S, Lv J Q. Ant colony algorithm based on dynamic search strategy and its application on path planning of robot[J]. *Control and Decision*, 2017, 32(3): 552-556.)
- [17] 王学武, 严益鑫, 顾幸生. 基于莱维飞行粒子群算法的焊接机器人路径规划[J]. *控制与决策*, 2017, 32(2): 373-377.
(Wang X W, Yan Y X, Gu X S. Welding robot path planning based on Levy-PSO[J]. *Control and Decision*, 2017, 32(2): 373-377.)
- [18] 薛晗, 邵哲平, 潘家财, 等. 基于文化萤火虫算法的足球机器人动态路径规划[J]. *控制与决策*, 2018, 33(11): 2015-2020.
(Xue H, Shao Z P, Pan J C, et al. Cultural firefly algorithm for dynamic path planning of soccer robot[J]. *Control and Decision*, 2018, 33(11): 2015-2020.)
- [19] Boyd S, Vandenberghe L. *Convex optimization*[M]. Cambridge: Cambridge University Press, 2004.
- [20] Deits R L H. Convex segmentation and mixed-integer footstep planning for a walking robot[D]. Boston: Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2014.
- [21] Tedrake R. Drake: Model-based design and verification for robotics[EB/OL]. (2019-09-25)[2019-09-29]. <https://drake.mit.edu/>.
- [22] Lawson C L, Hanson R J. *Solving least squares problems*[M]. Philadelphia: Siam, 1995.
- [23] Mosek A. MOSEK optimization toolbox for Matlab release 9.0.105[EB/OL]. (2019-09-25)[2019-09-29]. <https://docs.mosek.com/9.0/toolbox.pdf>.
- [24] Chen G R, Wang J Z, Wang S K, et al. Compliance control for a hydraulic bouncing system[J]. *ISA Transactions*, 2018, 79: 232-238.
- [25] Chen G R, Wang J Z, Wang S K, et al. Indirect adaptive robust dynamic surface control in separate meter-in and separate meter-out control system[J]. *Nonlinear Dynamics*, 2017, 90(2): 951-970.
- [26] Chen G R, Wang J Z, Wang S K, et al. Energy saving control in separate meter in and separate meter out control system[J]. *Control Engineering Practice*, 2018, 72: 138-150.
- [27] 陈光荣, 王军政, 汪首坤, 等. 基于主被动负载的负载独立口双阀节能控制系统研究[J]. *北京理工大学学报*, 2016, 36(10): 1053-1058.
(Chen G R, Wang J Z, Wang S K, et al. Separate meter in and separate meter out energy saving control system using dual servo valves under complex load conditions[J]. *Transactions of Beijing Institute of Technology*, 2016, 36(10): 1053-1058.)
- [28] Chen G R, Wang J Z, Ma L L, et al. Observer-based and energy saving control of single-rod electro-hydraulic servo system driven by servo motor[C]. *American Control Conference*. Piscataway: IEEE, 2015: 2224-2229.
- [29] 陈光荣, 王军政, 赵江波, 等. 基于虚拟分解控制的液压足式机器人单腿稳定阻抗控制[J]. *机器人*, 2017, 39(5): 704-714.
(Chen G R, Wang J Z, Zhao J B, et al. Stable impedance control of a single leg of hydraulic legged robot based on virtual decomposition control[J]. *Robot*, 2017, 39(5): 704-714.)
- [30] 陈光荣, 王军政, 汪首坤, 等. 自适应鲁棒控制器设计新方法在电液伺服系统中的应用[J]. *自动化学报*, 2016, 42(3): 375-384.
(Chen G R, Wang J Z, Wang S K, et al. Application of a new adaptive robust controller design method to electro-hydraulic servo system[J]. *Acta Automatica Sinica*, 2016, 42(3): 375-384.)

作者简介

陈光荣(1989—), 男, 讲师, 博士, 从事机器人控制与智能化技术的研究, E-mail: grchen@bjtu.edu.cn;

郭盛(1972—), 男, 教授, 博士生导师, 从事机器人学和机械设计及理论等研究, E-mail: shguo@bjtu.edu.cn;

王军政(1964—), 男, 教授, 博士生导师, 从事机器人控制、伺服控制技术等研究, E-mail: wangjz@bit.edu.cn;

曲海波(1983—), 男, 副教授, 博士生导师, 从事机器人学、机械设计及理论等研究, E-mail: hbqu@bjtu.edu.cn;

陈亚琼(1983—), 男, 博士后, 从事机器人学、机械设计及理论等研究, E-mail: chenyaqiong@bjtu.edu.cn;

侯博文(1985—), 男, 讲师, 博士, 从事轨道交通减噪与防灾技术的研究, E-mail: bwhou@bjtu.edu.cn.