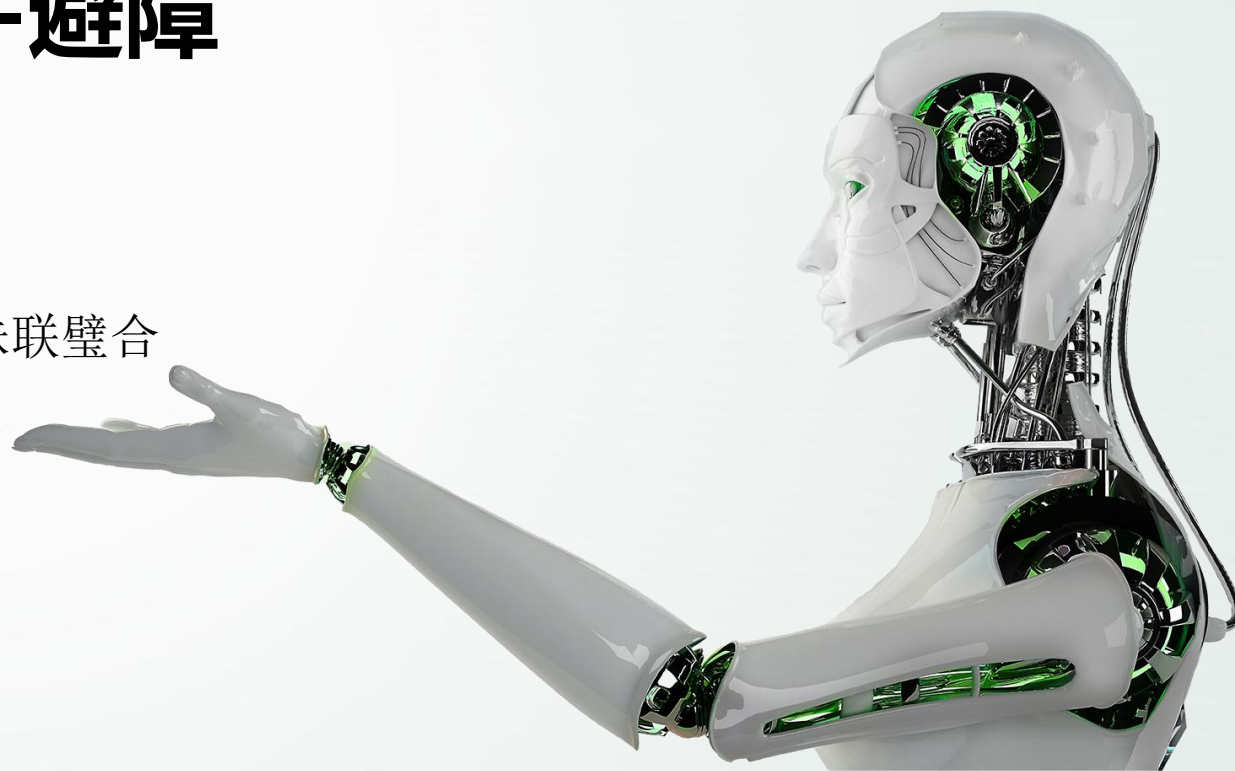


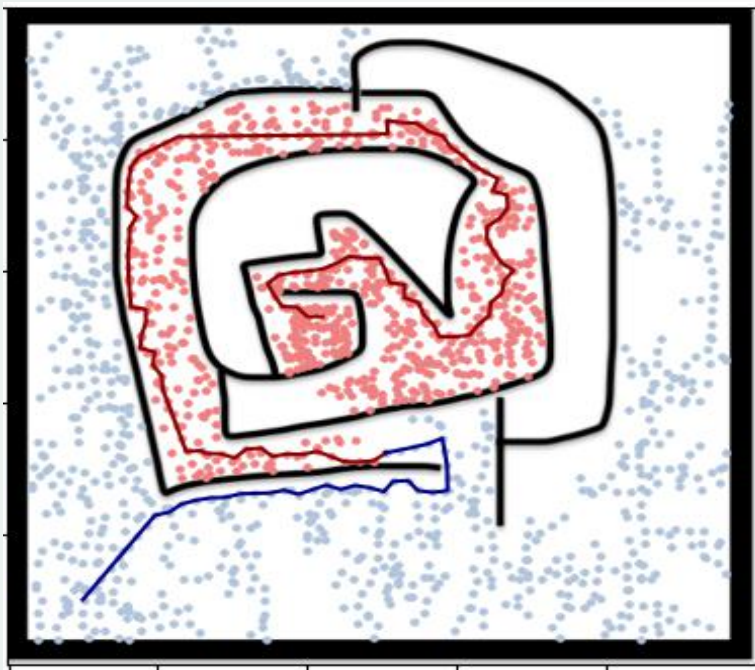
第五章

MAPF局部路径优化-避障

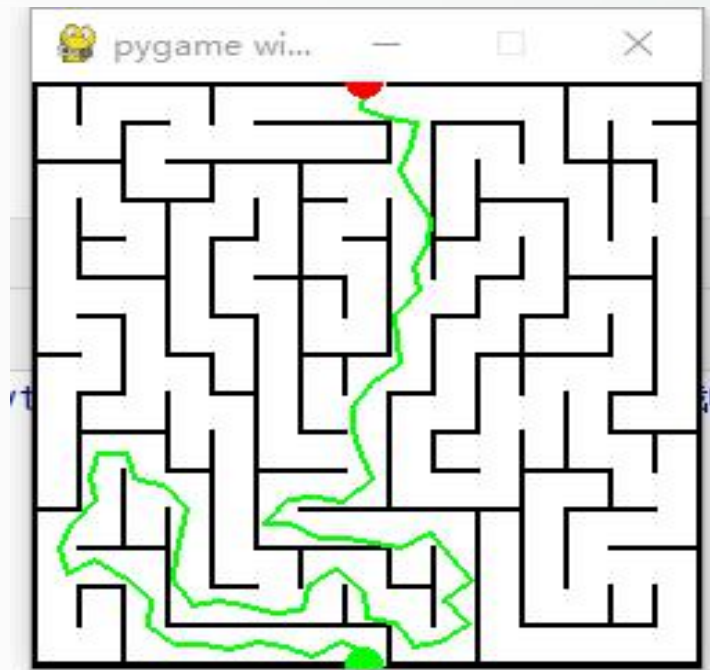
1. 作业点评
2. 机器人避障-传感器与算法的珠联璧合
3. 常见的局部路径优化算法
4. Bug算法及其改进
5. 人工势场法及其改进



1. 作业点评



rrt connection算法进行求解，程序运行基本稳定。



上课展示的代码中的算法进行了改进，使得算法不会出现跨过障碍物的情况。算法可以再极为复杂的情况下寻找路径。

综述小文章的总结

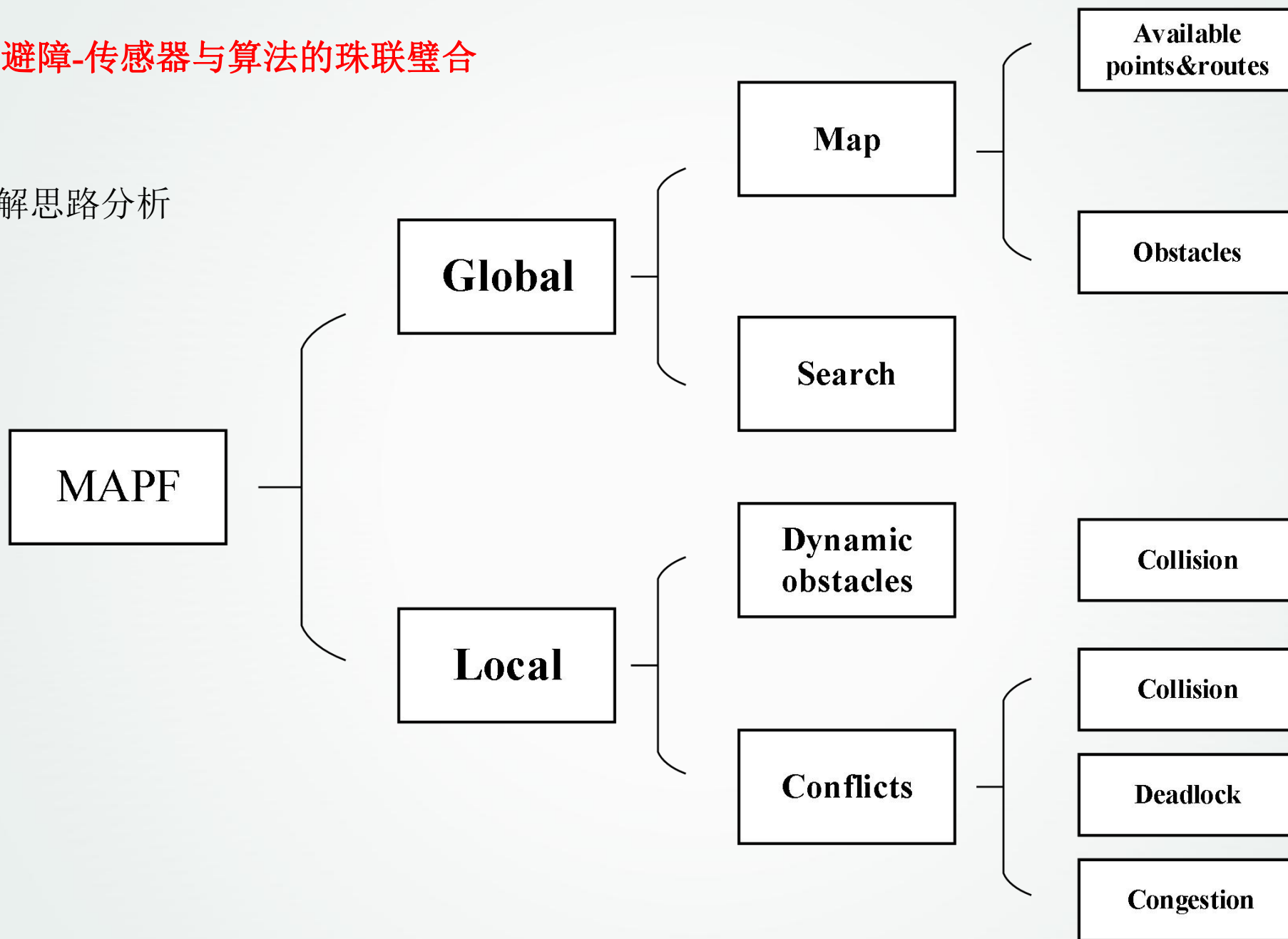
1. MAPF应用部分缺失
2. 算法整理不够系统

关于作业是否提供源代码

1. 小的case一般会提供代码，有些会提供多种语言的代码
2. 课程中讲解的论文涉及的算法整体框架、策略及完整算法不提供，需要自己复现
3. 思路大于编程。

2. 机器人避障-传感器与算法的珠联璧合

MAPF求解思路分析



2. 机器人避障-传感器与算法的珠联璧合

感知周边环境信息是局部避障的第一步

IMU+算法优化

机器人需要通过传感器 实时获取自身周围障碍物信息，
包括尺寸、形状和位置等信息。

超声波传感器

作用距离较短
探测盲区
测量周期较长
传感器干扰

红外传感器

测量距离近
透明/近似黑体无法测距

激光传感器

性能良好
低端产品类似于红外传感器
高端产品价格高

视觉传感器

价格低
精度差
对算力要求高

3. 常见局部路径优化算法

基于局部优化的算法：PFM, Fuzzy, RL, BP, GA等

基于应激反应的算法：Bug算法及其各种变形

全局于局部混合的算法：先全局再局部，基于规则

新型局部优化算法：CBS，改进RL等

4. Bug算法及其改进

Bug算法的基本思想是在发现障碍后，围着检测到的障碍物轮廓行走，从而绕开障碍物。

<https://www.codeproject.com/Articles/646347/Robotics-Motion-Planning-and-Navigation-Bug-Algorithm>

4. Bug算法及其改进

Bug1算法

1. $q_{\text{start}} \rightarrow q_{\text{goal}}$,
2. 遇到障碍物, 到达撞击点 q^H_1 ;
3. 从 q^H_1 开始环绕障碍物, 得到障碍物轮廓并返回;
4. 从起点出发, 运行至距离终点最近的点 q^L_1 ;
5. 再次遇到障碍物, 到达撞击点 q^H_2 , 然后沿着障碍物边界移动,
6. 到达 q^L_2 ; 继续朝目标前进到达目标位置。

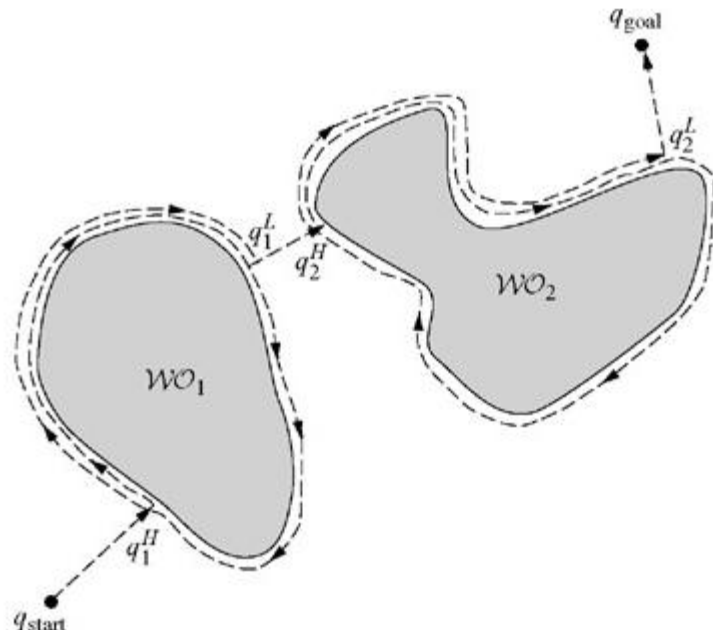


Figure 2.1: The Bug1 algorithm successfully finds the goal.

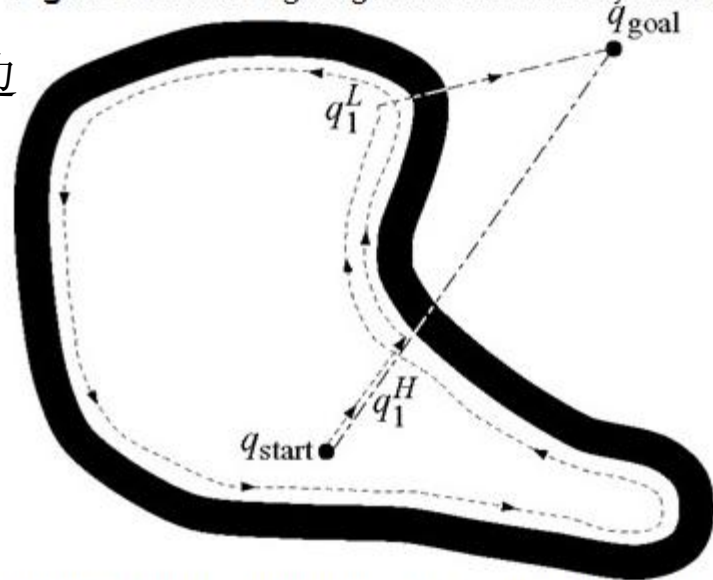


Figure 2.2: The Bug1 algorithm reports the goal is unreachable.

4. Bug算法及其改进

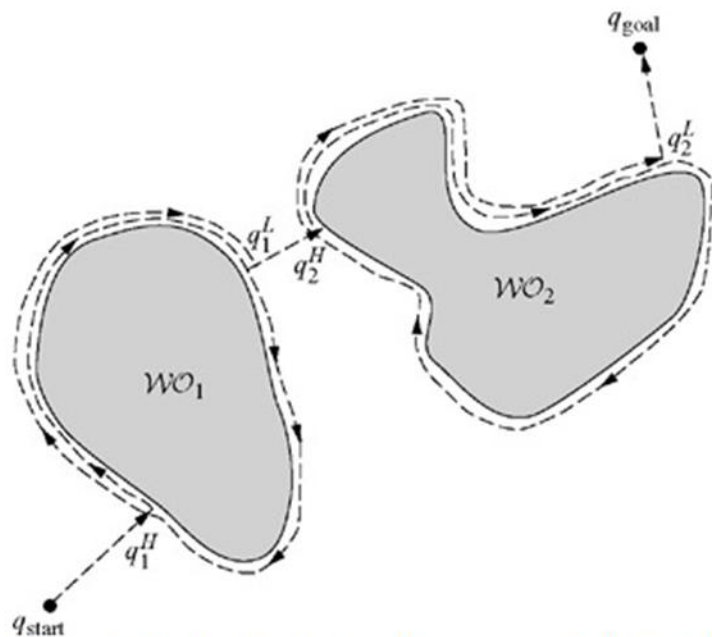


Figure 2.1: The Bug1 algorithm successfully finds the goal.

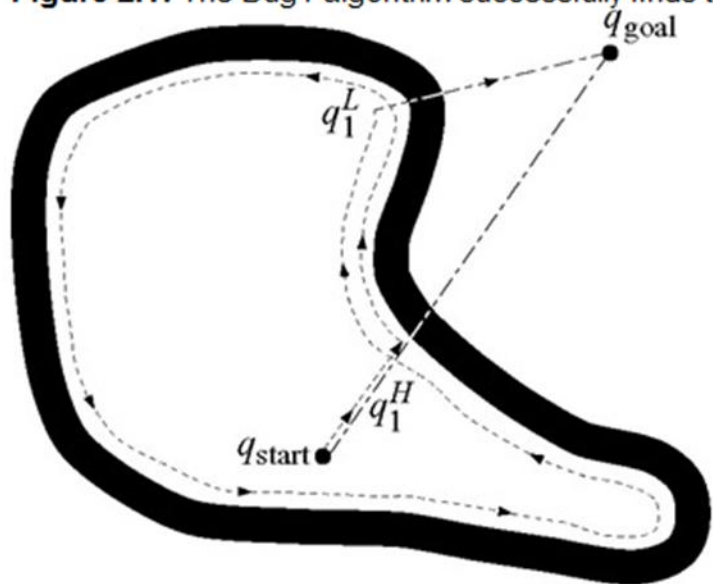
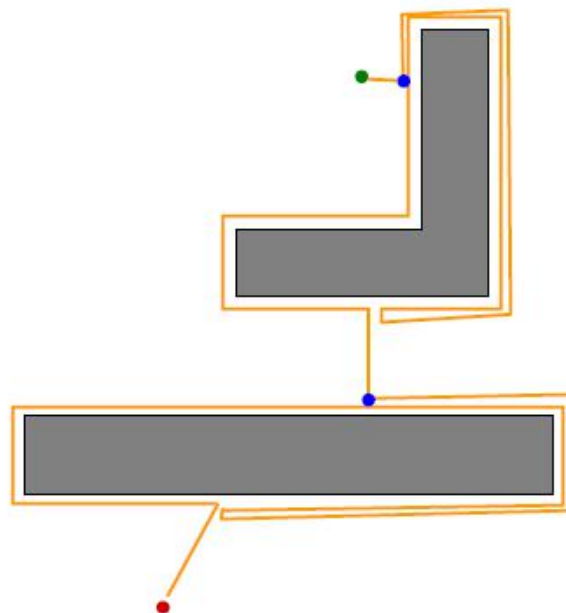


Figure 2.2: The Bug1 algorithm reports the goal is unreachable.

Bug1算法



"Bug 1" algorithm

- 1) head toward goal
- 2) if an obstacle is encountered, circumnavigate it *and* remember how close you get to the goal
- 3) return to that closest point (by wall-following) and continue

Bug1算法的效率很低，但可以保证机器人能到达任何可达的目标

4. Bug算法及其改进

Bug2算法

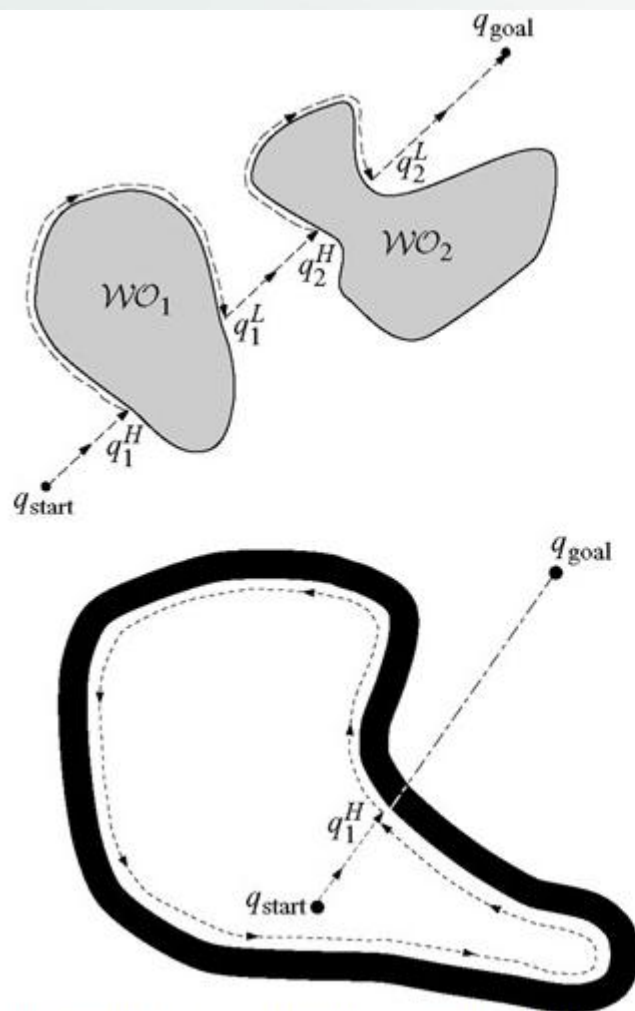
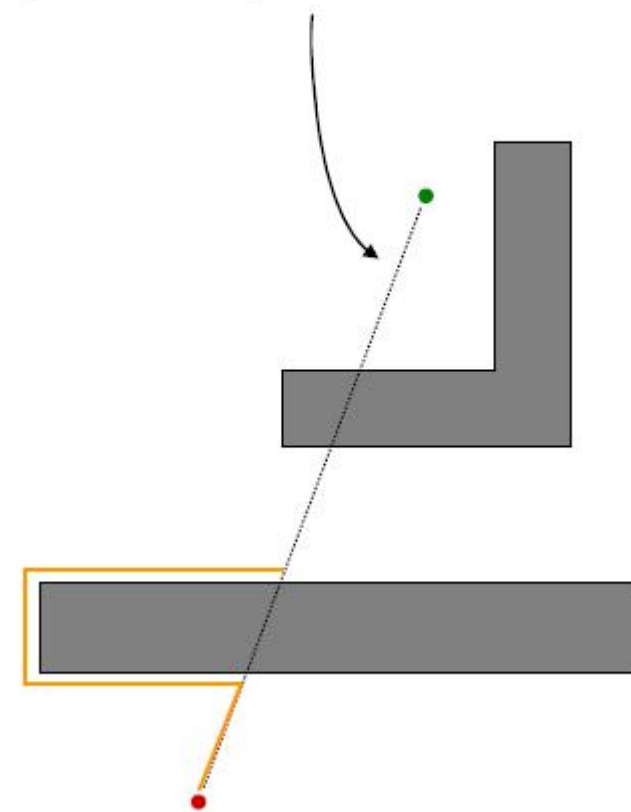


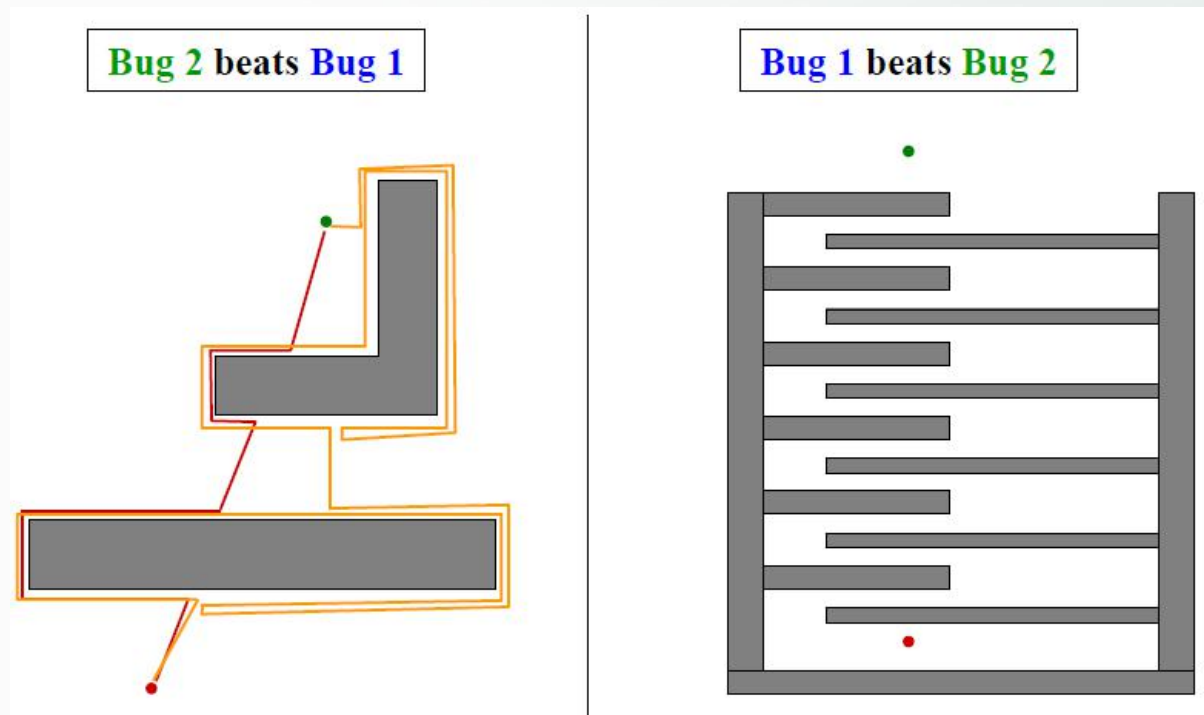
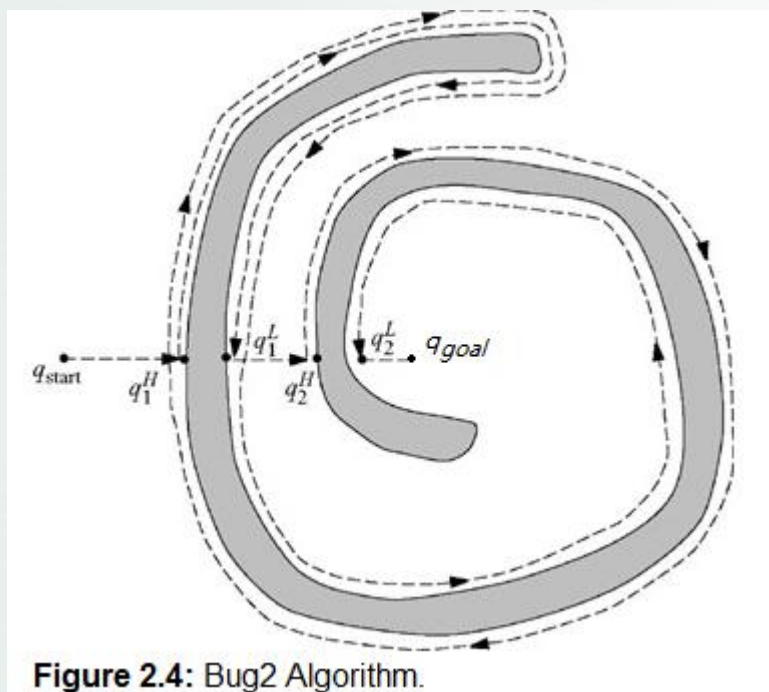
Figure 2.3: (Top) The Bug2 algorithm finds a path to the goal. (Bottom) The Bug2 algorithm reports failure.

Call the line from the starting point to the goal the *m-line*



4. Bug算法及其改进

Bug1 vs Bug2算法



说明： Bug1是一种穷举算法，Bug2是一种贪心算法。在原链接中，作者认为Bug1在处理复杂路况时更有优势，其实不然。穷举算法的最大弊端就是空间爆炸。

Tangent Bug算法

*TangentBUG*算法是对*BUG2*算法的改进。利用机器人上距离传感器的感知对障碍物做出提前规避。

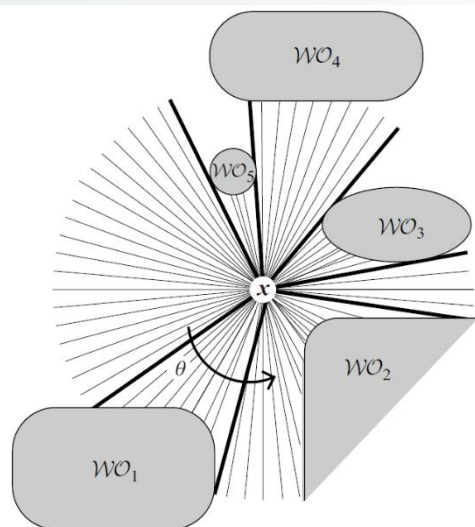


Figure 2.5 The thin lines are values of the raw distance function, $\rho_R(x, \theta)$, for a fixed $x \in \mathbb{R}^2$, and the thick lines indicate discontinuities, which arise either because an obstacle occludes another or the sensing range is reached. Note that the segments terminating in free space represent infinitely long rays.

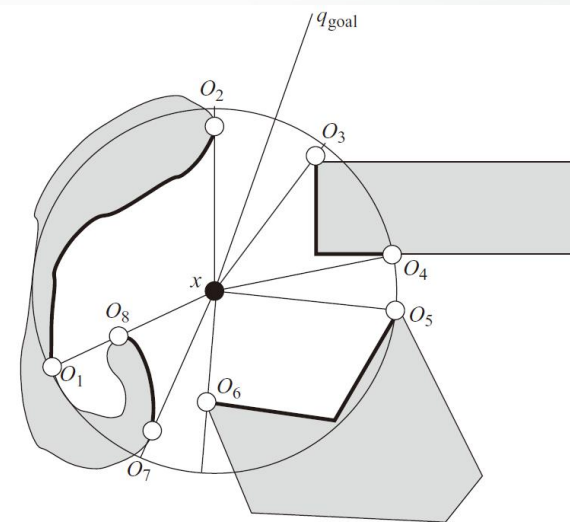


Figure 2.6 The points of discontinuity of $\rho_R(x, \theta)$ correspond to points O_i on the obstacles. The thick solid curves represent connected components of the range of $\rho_R(x, \theta)$, i.e., the intervals of continuity. In this example, the robot, to the best of its sensing range, believes there is a straight-line path to the goal.

Tangent Bug算法

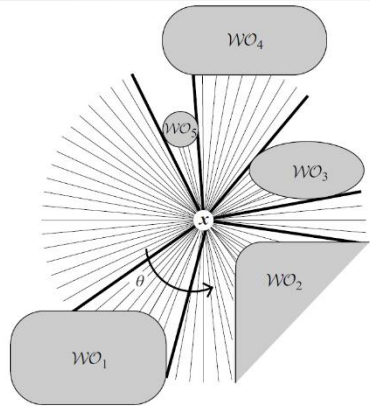


Figure 2.5 The thin lines are values of the raw distance function, $\rho_R(x, \theta)$, for a fixed $x \in \mathbb{R}^2$, and the thick lines indicate discontinuities, which arise either because an obstacle occludes another or the sensing range is reached. Note that the segments terminating in free space represent infinitely long rays.

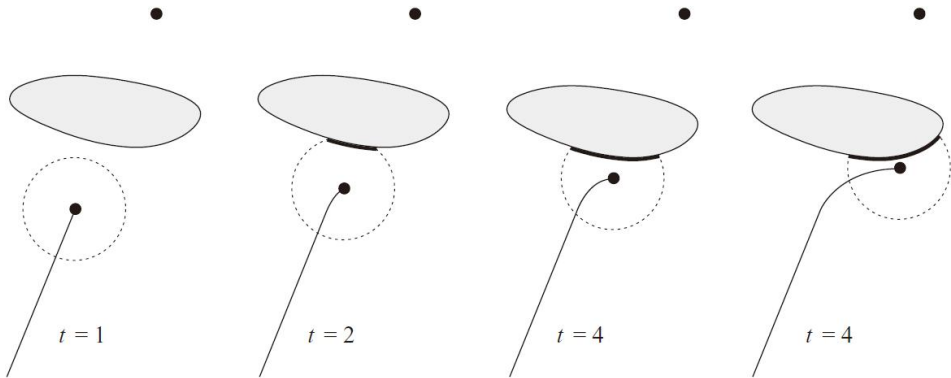


Figure 2.9 Demonstration of motion-to-goal behavior for a robot with a finite sensor range moving toward a goal which is “above” the light gray obstacle.

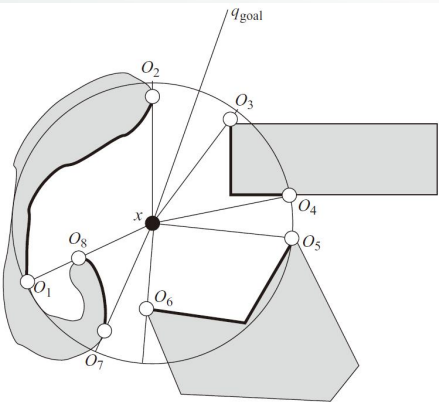


Figure 2.6 The points of discontinuity of $\rho_R(x, \theta)$ correspond to points O_i on the obstacles. The thick solid curves represent connected components of the range of $\rho_R(x, \theta)$, i.e., the intervals of continuity. In this example, the robot, to the best of its sensing range, believes there is a straight-line path to the goal.

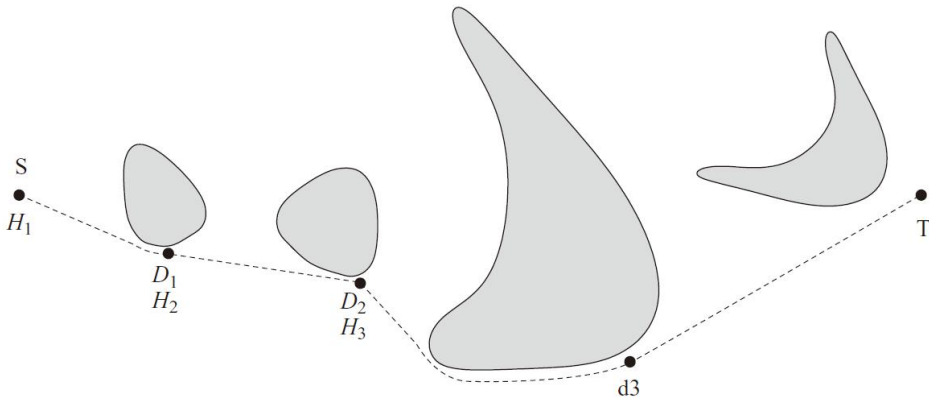


Figure 2.13 Path generated by Tangent Bug with infinite sensor range. The dashed-lines correspond to the motion-to-goal behavior and there is no boundary-following.

5. PFM算法及其改进

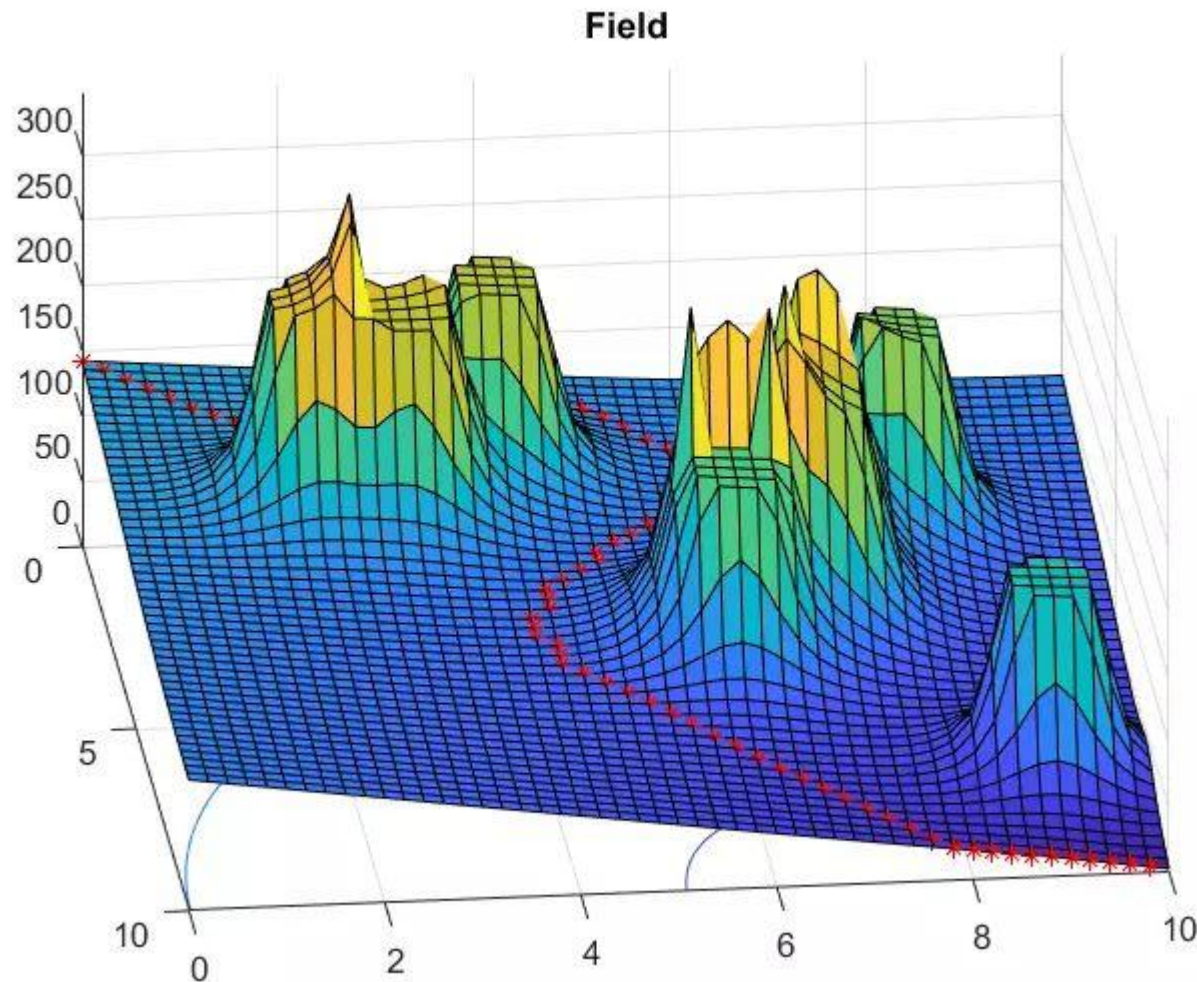
PFM算法

人工势场法路径规划是由Khatib提出的一种虚拟力法。

它的基本思想是将机器人在周围环境中的运动，设计成一种抽象的**人造引力场**中的运动

目标点对移动机器人产生“**引力**”，
障碍物对移动机器人产生“**斥力**”，
最后通过求**合力**来控制移动机器人的运动。

应用势场法规划出来的路径一般是比较平滑并且安全，但是这种方法存在**局部最优**点问题。



PFM算法

第35卷第3期

山东理工大学学报(自然科学版)

Vol.35 No.3

2021年5月

Journal of Shandong University of Technology(Natural Science Edition)

May 2021

文章编号: 1672-6197(2021) 03-0001-06

改进人工势场法的移动机器人局部路径规划

王迪, 李彩虹, 郭娜, 高腾腾, 刘国名

(山东理工大学 计算机科学与技术学院, 山东 淄博 255049)

5. PFM算法及其改进

PFM算法

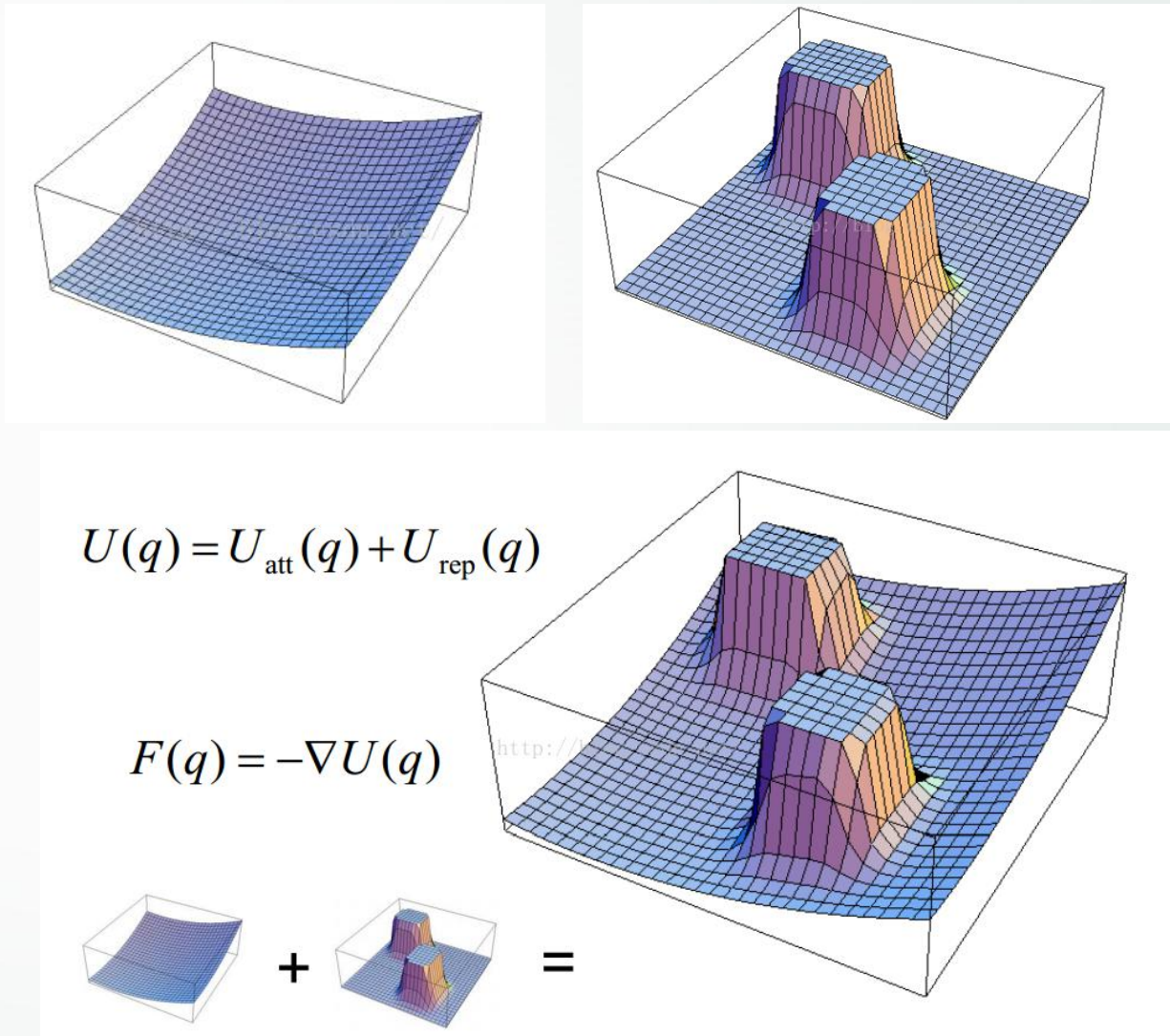
基于人工势场的移动机器人路径规划算法是 Khatib 提出的^[19]。基本思想是在机器人工作的环境中建立一个人工势场,目标点对机器人产生引力,障碍物对机器人产生斥力,由引力和斥力叠加起来的合力对机器人产生引导作用,使机器人到达目标点。为了解决路径规划过程中的目标不可达问题,本文采取添加目标点与机器人距离的改进斥力势场^[20],使目标点处的势场为全局最小点。人工势场定义为

$$U = U_{\text{att}} + U_{\text{rep}}, \quad (1)$$

式中: U 为势场和; U_{att} 为目标点对机器人产生的引力场; U_{rep} 为障碍物对机器人产生的斥力场。定义目标点对机器人的引力和障碍物对机器人的斥力分别对应引力场和斥力场的负梯度,根据偏导得出人工势场对机器人的作用力 F 为

$$F = F_{\text{att}} + F_{\text{rep}}, \quad (2)$$

式中: F 为合力; F_{att} 为目标点对机器人产生的引力; F_{rep} 为障碍物对机器人产生的斥力。



5. PFM算法及其改进

引力势场函数定义为

$$U_{\text{att}} = \frac{1}{2}k \|X - X_g\|^2, \quad (3)$$

式中: $X_g = (x_g, y_g)$ 为目标点的坐标; $X = (x, y)$ 为机器人的坐标; U_{att} 为目标点对机器人产生的引力场; k 为引力增益系数, $\|X - X_g\|$ 为机器人到目标点的欧几里得距离。目标点对机器人的引力为引力势场函数的负梯度:

$$F_{\text{att}} = -k \|X - X_g\|. \quad (4)$$

斥力势场函数定义为

$$U_{\text{rep}} = \begin{cases} \frac{1}{2}m(\frac{1}{d} - \frac{1}{d_0})^2 d_a^2, & d \leq d_0, \\ 0, & d > d_0, \end{cases} \quad (5)$$

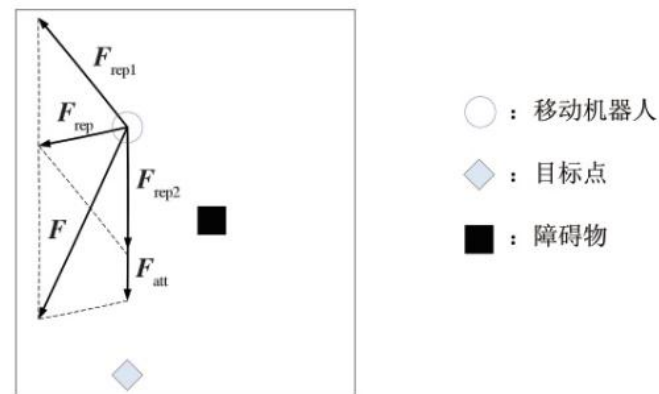
式中: m 为斥力增益系数; d 为机器人与障碍物的距离; d_0 为障碍物影响距离; d_a 为机器人与目标点之间的距离。障碍物对机器人的斥力为

$$F_{\text{rep}} = \begin{cases} F_{\text{rep1}} + F_{\text{rep2}}, & d \leq d_0, \\ 0, & d > d_0, \end{cases} \quad (6)$$

$$F_{\text{rep1}} = m(\frac{1}{d} - \frac{1}{d_0}) \frac{1}{d^2} d_a^2, \quad (7)$$

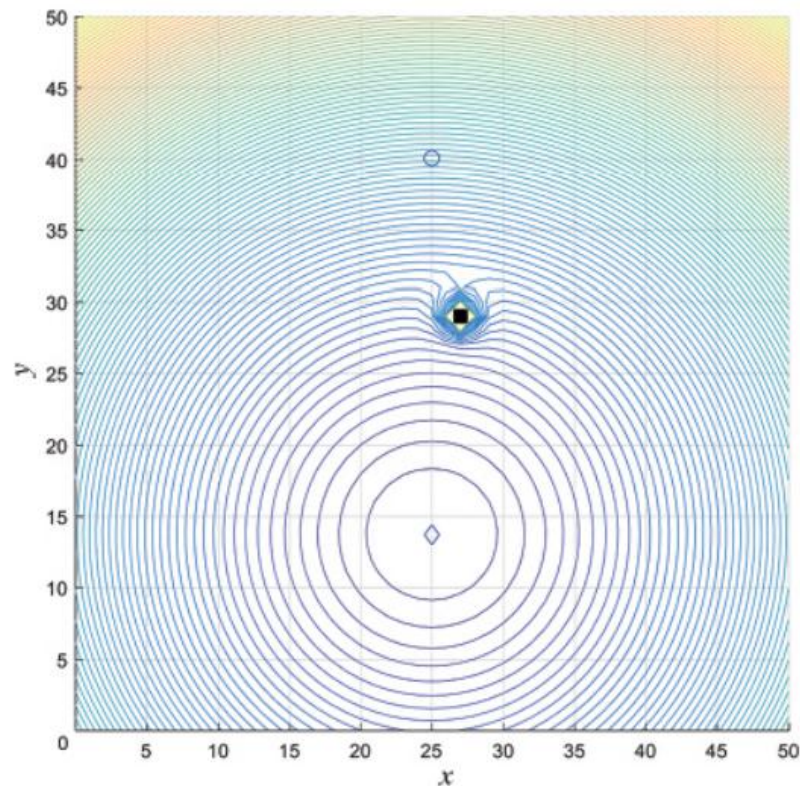
$$F_{\text{rep2}} = -m(\frac{1}{d} - \frac{1}{d_0})^2 d_a^2, \quad (8)$$

式中: 分量 F_{rep1} 由障碍物指向机器人; 分量 F_{rep2} 由机器人指向目标点。添加目标点与机器人距离, 从力的角度上, 增强了引力, 避免目标点附近有障碍物时, 斥力过大, 机器人无法到达目标点的情况。人工势场法受力分析如图 1 所示。



5. PFM算法及其改进

引力势场和斥力势场构成的人工势场等势图如图 2 所示。在势场的作用下,机器人沿着梯度下降的方向移动。目标点处的势场为 0,是全局最小点。



构造人工势场: 起点高, 终点低, 形成高低差; 障碍物设置为山峰;

梯度下降法: 下山的过程。以当前位置为基准, 最陡峭的地方, 朝着下降方向走一步, 继续以当前位置为基准, 再找最陡峭的地方, 直到最后到达最低处。

PFM的改进

引力势函数的改进：如果当前点与目标点的距离太远，会形成很大的引力势，使得机器人的运动速度过快。采用分段的引力势函数解决问题

斥力势函数的改进：如果当前点与两个障碍物等距，可能会造成机器人在障碍物间的中线上来回跳动。重新定义斥力势函数为当前点与最近的障碍物点之间的距离

距离的计算：一般来说，我们用直线距离来度量两点间的距离。可更换或改进距离计算公式

局部极小值问题：局部极小值点，虽然梯度为零，但并不是我们想要的终点，在局部最小值处加入一个扰动，跳出局部极小值。

2 改进策略

2.1 虚拟目标点法

利用人工势场法进行路径规划,当机器人所受的引力和斥力和为0时,存在局部极小点问题,导致机器人在某一区域内震荡或静止不动,陷于陷阱区域,无法到达目标点。本文通过设置虚拟目标点和环境判断参数来解决这个问题。

当机器人在移动过程中进入陷阱区域,采取设置虚拟目标点的策略,即在机器人所受斥力分量 F_{repl} 左转 90° 的方向上的合适位置,设置虚拟目标点。忽略目标点产生的引力和障碍物产生的斥力,在虚拟目标点的引导下,使机器人向前移动一步,移动后撤销虚拟目标点,计算当前步骤下的合力及下一步坐标。重复以上步骤,直到机器人逃出局部极小点区域。

采用虚拟目标点法逃出局部极小点的示意图如

图3所示。图3中箭头表示机器人下一步的移动方向。当前机器人所处位置为点A,因为引力与斥力的合力两者方向相反导致震荡,陷入局部极小点。机器人在所受斥力分量 F_{repl} 左转 90° 方向上的点B处,设置虚拟目标点,经虚拟目标点牵引到达点B。在点B处,机器人在合力的作用下,将往点A方向移动。经判定,机器人陷入了局部极小点陷阱,在点A和点B之间发生震荡。根据改进措施,在点B处机器人所受斥力分量 F_{repl} 左转 90° 方向上的点C处,设置虚拟目标点。经虚拟目标点的引导,机器人到达C处。重复此步骤,直到机器人逃出局部极小点区域,最终到达目标点。

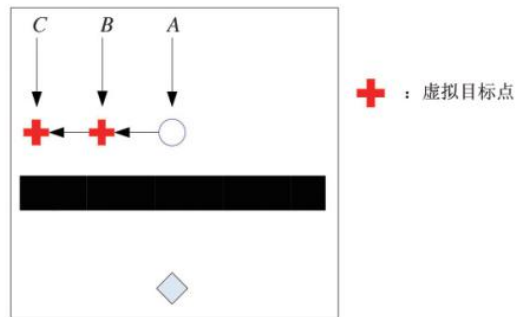


图3 虚拟目标点法示意图

Fig.3 Schematic diagram of the virtual target method

本章作业

1. 阅读本章讲解案例的文章
2. 阅读参考文献压缩包中的其他文献
3. 在bug算法和PFM算法中任选一种求解迷宫问题