

基于ROS的自主导航算法优化

柳晓峰, 邝坚

(北京邮电大学软件学院, 北京 100876)

摘要: 本文的目的是基于ROS当前的导航算法进行优化, 实现适用于不同结构机器人的导航算法, 以解决部分结构下机器人导航效果不佳的问题, 从而摆脱软件层面对机器人机械结构的限制。本文主要针对自主导航模块中的代价地图进行改进, 对各个层级进行拆解分析, 并对其中膨胀层进行分层处理, 使其满足不同机械结构的需求。通过这种方式改进后的代价地图可以根据机器人实际的结构、尺寸等参数进行调整, 自主导航过程更加流畅, 提高了稳定性。最后文中给出了实验结果, 验证了本方案的可行性。

关键词: 机器人操作系统; 自主导航; 机械结构

中图分类号: TP242

ROS-based autonomous navigation algorithm optimization

LIU Xiaofeng, KUANG Jian

(Beijing University of Posts and Telecommunications Beijing 100876)

Abstract: The purpose of this paper is to optimize the current navigation algorithms based on ROS, to implement navigation algorithms suitable for robots with different structures, to solve the problem of poor robot navigation under some structures, and to get rid of the restrictions on the mechanical structure of the robot at the software level. This article focuses on improving the cost map in the autonomous navigation module, disassembling and analyzing each level, and layering the expansion layer to meet the needs of different mechanical structures. The cost map improved in this way can be adjusted according to the actual structure and size of the robot, and the autonomous navigation process is smoother and the stability is improved. Finally, the experimental results are given in the article, which verifies the feasibility of this scheme.

Key words: Robot operating system; Autonomous navigation; Mechanical structure

0 引言

近几年人工智能技术的飞速发展, 为自主导航领域提供了有力的技术支撑, 使得该领域, 尤其是轮式自主导航机器人^[1]领域逐步走入大众视野, 京东、美团等公司也纷纷在无人配送机器人领域投入了大量的研发精力。作为机器人领域应用最为广泛的基础平台, Robot Operating System (机器人操作系统^[2], 简称 ROS) 为用户提供了大量辅助功能, 使得开发者可以统一在该平台下进行研发, 从而达到标准化的目的。

在自主导航方面, ROS 提供了自主导航所需要的底层模型, 例如 costmap_2d^[3]等, 满足了基本的导航需求。开发者在 ROS 上做自主导航相关的研究时不需要关心底层模型, 只需要在其基础上对导航算法进行实现即可。这对大部分开发者, 尤其是早期的一些开发者来说提供了极大的便利。这些底层模型在应用于圆形结构的机器人 (例如扫地机器人), 或是其他近似圆形结构的机器人时相对稳定, 效果较好。但是在应用于类似于前轮驱动的四轮方形结构机器人时效果不佳, 这是由于目前的模型本身不完善造成的。

作者简介: 柳晓峰 (1994-), 男, 物联网与智能硬件

通信联系人: 邝坚 (1966-), 男, 教授、硕导, 物联网与智能硬件. E-mail: jkuang@bupt.edu.cn

因此, 本文将针对上述问题对 ROS 中的相关底层模型进行优化, 使其在不同机械结构的自主导航机器人上都有较好的表现, 从而使开发者不必受限于机械结构的限制, 可以根据实际场景选择最符合需求的结构。

1 自主导航流程简述

这一节中将对一个完整的自主导航流程中的核心模块逐一进行简述, 以定位问题的相关部分从而对其进行改进。

1.1 地图构建

自主导航系统往往依托于一张提前构建好的地图, 仅有小部分导航系统采用不提前构建地图的探索式方案, 其导航表现远不如前者, 因此地图构建通常是自主导航系统的核心模块。最常用的建图方案为 Simultaneous Localization And Mapping (即时定位与地图构建^[4], 建成 SLAM)。SLAM 算法众多, 根据传感器的不同可分为激光 SLAM^[5]和视觉 SLAM^[6]等。图 1 为室内环境下通过 Rtabmap^[7] (一种视觉 SLAM 方法) 构建的地图。



图 1 SLAM 地图

Fig. 1 SLAM map

1.2 全局路径规划

在自主导航中, 路径规划^[8]分为全局路径规划和局部路径规划两个部分, 其中全局路径规划负责在地图中寻找从机器人当前位置到目标位置的最优路径。常见的全局路径规划算法有 Dijkstra 算法^[9]、A*算法等, 不同算法各有利弊, 需根据实际需求选取合适的算法, 本小节以 Dijkstra 算法为例做简要介绍。

Dijkstra 算法也即: 单源最短路径算法。算法的基本思想是: 每次找到离源点最近的一个顶点, 然后以该顶点为中心进行扩展, 最终得到源点到其余所有点的最短路径。算法具体流程如下:

- (1) 通过 Dijkstra 计算图 G 中的最短路径时, 需要指定起点 s (即从顶点 s 开始计算);
- (2) 此外, 引进两个集合 S 和 U 。 S 的作用是记录已求出最短路径的顶点 (以及相应的最短路径长度), 而 U 则是记录还未求出最短路径的顶点 (以及该顶点到起点 s 的距离);

(3) 初始时, S 中只有起点 s ; U 中是除 s 之外的顶点, 并且 U 中顶点的路径是“ s 到该顶点的路径”。然后, 从 U 中找出路径最短的顶点, 并将其加入到 S 中; 接着, 更新 U 中的顶点和顶点对应的路径。然后, 再从 U 中找出路径最短的顶点, 并将其加入到 S 中; 接着, 更新 U 中的顶点和顶点对应的路径;

(4) 重复该操作, 直到遍历完所有顶点。

1.3 局部路径规划

局部路径规划, 又称本地实时规划, 主要工作是根据全局路径结果以及周围环境信息生成实时速度指令, 下发给控制模块控制机器人移动^[10]。在机器人自主导航的过程中, 除了在 SLAM 方法实时构建地图的过程中采集到的静态障碍物(如围墙、树木等)外, 还需要应对在行驶过程才会出现的动态障碍物(行人、车辆等), 此类障碍物由于建图时并不在环境中, 且往往是移动状态, 所以需要通过激光雷达等传感器对周围动态障碍物信息进行实时采集, 并将障碍物映射到地图中相对应的位置。局部路径规划便需要在已有的静态地图和实时的动态障碍物中寻求实时的局部最优轨迹, 控制机器人安全稳定的行驶到目的地。

局部路径规划是利用 ROS 中 `base_local_planner` 包实现的, 该包使用 Trajectory Rollout 和 Dynamic Window approaches 算法计算机器人每个周期内应该行驶的速度和角度(dx , dy , $d\theta$ velocities)。通过地图数据, 通过算法搜索到达目标的多条路径, 利用一些评价标准(是否会撞击障碍物, 所需要的时间等等)选取最优的路径, 并且计算所需要的实时速度和角度。

2 代价地图优化

根据上一节内容可以发现, 影响机器人导航性能的主要是局部路径规划部分, 又因为本文目的是使同样的导航算法可以在不同结构的机器人上都有较好的表现, 所以本文研究内容集中在与局部路径规划算法紧密相关的代价地图模块——`costmap_2d`。本节将对代价地图进行简要概述, 并结合不同架构机器人的行驶特点对其进行改进。

2.1 代价地图概述

无论是激光雷达还是如 kinect 或 xtion pro 深度相机作为传感器跑出的 2D 或 3D SLAM 地图, 都不能直接用于实际的导航, 必须将地图转化为代价地图。ROS 的代价地图通常采用网格(grid)的形式, 每个网格可以存储 0~255 个不同的数值, 不同的数值代表该网格离障碍物的远近程度。此外, 可以根据不同的数值将网格分为三种情况: 占用区域(Occupied), 即该网格有障碍物; 自由区域(Free), 即该网格无障碍物; 未知区域(Unknown Space)。

代价地图是在 SLAM 地图的基础上进行一系列处理得到的, 其中膨胀操作是这个转化过程的核心。膨胀就是从占用 cell 向外传播代价值的过程, 随着传播距离增加代价值会依据膨胀半径递减。此外, 代价地图由多层组成, 每种功能放置一层中。代价地图通常静态地图是一层, 障碍物是另一层。膨胀层是最后一层。这三层组合成了 master map(最终的代价地图), 供给路线规划模块使用。

2.2 机械结构

由于运动方式简单、安全稳定等原因,采用圆形结构的机器人早已走进千家万户。最常见的扫地机器人、在物流系统中发挥重要作用的分拣机器人等都是采用的圆形结构,其中分拣机器人虽然整体外观为正方体结构,但本质上是圆形运动方式。

如图 2 所示,扫地机器人的底盘设计基本都保持一致,采用两个驱动轮、一个或两个万向轮,驱动方式都为双轮差速驱动。采用这种结构的好处在于机器人在任何时候都可以通过原地旋转的方式调整姿态,转向灵活,很少出现无法移动的情况,稳定性高。

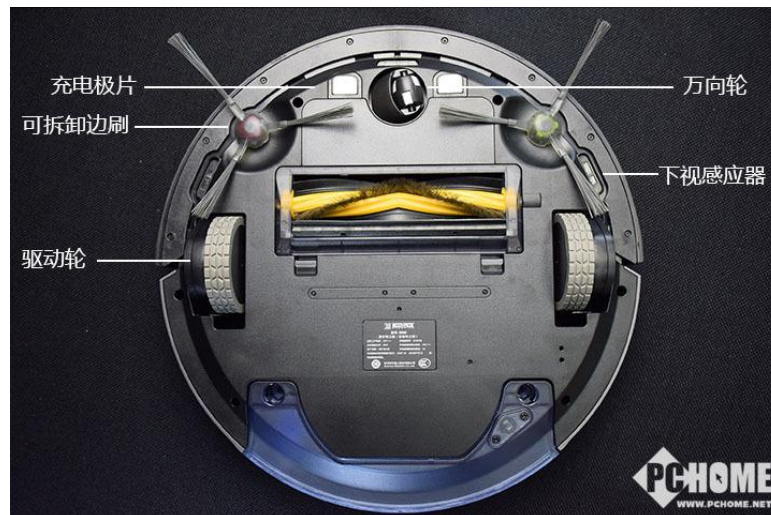


图 2 扫地机器人底盘

Fig. 2 Chassis of sweeping robot

虽然将机器人设计成圆形或正方形更方便,效果也更好,但不是所有的应用场景都适合使用圆形结构。例如,在大型的仓库,由于货物不像分拣机器人那样多为小件货物,所以往往需要设计成长方形结构。

不同于汽车驱动,一个移动机器人系统的驱动方式以双轮差速驱动为主。采用双轮差速驱动可以很方便地进行姿态调整,甚至可以原地旋转,这使得机器人的移动十分灵活,便于机器人进行避障,在移动机器人领域有着重要的地位。然而当车体为长方形时,四轮结构的机器人只能采用前轮为驱动轮、后轮为万向轮或者后轮为驱动轮、前轮为万向轮的方式。

由于双轮差速驱动在进行原地旋转时需要左右两驱动轮以相同的速度进行反向转动,所以原地旋转的旋转中心为两驱动轮的中心,导致原地旋转时从动轮的旋转半径较大。也正是由于这个原因,此种结构的机器人在自主导航时的表现相较于圆形机器人较差。

2.3 代价地图改进

根据前面两个小节的分析可知,当前 ROS 中的代价地图中没有考虑机器人的运动学中心,在圆形机器人上面表现较好,但不适应四轮的前驱或后驱机器人,需要对代价地图进行优化。另外,由于代价地图采用分层结构,主要包含三层,其中 StaticLayer 层和 ObstacleLayer

层的数据直接来自传感器或地图，不能也不应该对这两层的数据进行更改。因此，要对代价地图进行优化，便需要对与路径规划直接关联的 InflationLayer 层进行改进。

机器人在代价地图中的模型为两个同心圆，一个机器人的轮廓外切圆和一个机器人内切圆，对于圆形或方形机器人，这两个同心圆半径相当、相差不大，但对于长方形机器人这两个同心圆便会有较大的差距，尤其是进行原地旋转时，这个差距由于旋转中心转移到驱动轮轴的中心而扩大到原来的两倍。因此需要将圆形机器人与长方形机器人进行区分，使得代价地图在长方形机器人上面也有较好的表现。具体改进方法如下：

(1) 设定膨胀层时以机器人的原地旋转中心为准

采用这种方式进行改进的优点在于：对圆形机器人来讲，由于机器人的几何中心与旋转中心一致，因此不会对圆形机器人产生新的影响，不必对圆形机器人进行多余的改进或优化。对于长方形机器人，首先以驱动轮的轴中心为圆心，以圆心到驱动轮的距离为半径得到新的圆；再驱动轮的轴中心为圆心，以圆心到机器人轮廓的最远距离为半径得到新的外接圆。这两个新的同心圆作为机器人在代价地图中的新模型。

这样在机器人新模型的内切圆中，机器人原地旋转时内切圆部分不会发生平移，仅在地图中相同位置进行旋转。而内切圆以外、外接圆以内的部分会因为旋转中心偏移，导致原地旋转时这一部分车体在地图中的位置发生变更，原地旋转也不再是真正意义上的原地，使机器人失去了原地旋转带来的灵活性。因此需要使机器人尽量避免行驶到会限制机器人旋转的位置。

(2) 膨胀区域划分为行驶自由区和行驶受限区

根据新的机器人模型，原本单一的 InflationLayer 层，不再只是用于表示距离障碍物的远近，可以根据该距离对 InflationLayer 层进行划分，将车体内切圆以外、外接圆以内的距离设定为行驶受限区域，将外接圆以外的距离设定为行驶自由区域。在行驶自由区域中，车体的原地旋转不会受到任何障碍物的干扰，因此尽管原地旋转时部分车体会发生位置变更，但这个变更并不会导致车体移动受到阻碍，原地旋转的灵活性仍然存在。而对于行驶受限区域，车体原地旋转有可能受到障碍物的阻碍，此时原地旋转的角度受到限制，灵活性被破坏。

为了让机器人尽量不要走到行驶受限区域，采取增加代价的方式，使机器人移动到该区域的代价显著提高。改变原先膨胀区域代价值的计算方式，当进入受限区域时，不再与自由区域的代价值保持连续，而是变为断崖式的差值，这样在机器人的路径规划模块选取路径时便不会轻易选取行驶受限的路线，从而达到减少机器人进入受限区域的效果。

同时为了使路径规划算法兼容改进后的代价地图，根据新的膨胀区域，需要在路径规划中添加对行驶自由区域和受限区域的处理，基本思想如图 3 所示。受限获取膨胀区域的数据，根据数值所在区间判断当前属于行驶自由区域还是受限区域。若为行驶自由区域，则不需要进行其他操作，与之前版本保持一致即可；若为行驶受限区域，则进一步获取机器人模拟轨迹的角速度，当角速度超过一定阈值（若以该阈值进行移动，将进入受限区域）后，对该轨迹的权重加大，降低轨迹评分。

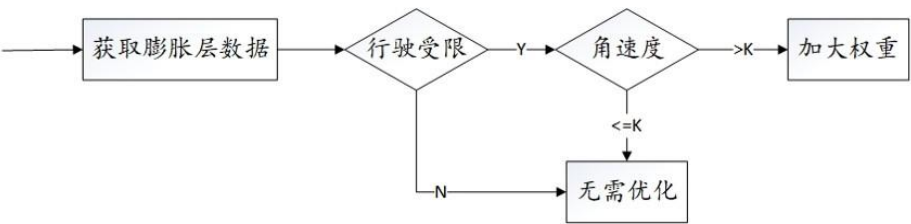


图 3 优化后的路径规划流程
Fig. 3 Optimized path planning process

165 **3 实验结果**

本节针对文中提出的自主导航算法优化方案进行了实验，重点测试了常规导航算法在方形结构机器人上的表现，验证了本方案的可行性。

170 **3.1 实验平台**

为验证不同机械结构机器人在改进后的表现，选取实际应用中最常见的两种机械结构：四轮前驱结构作为实验对象。如图 4 所示，本实验平台采用四轮独立悬挂底盘，缓冲由不平路面传给底盘或机器人的冲击力，并衰减由此引起的震动，避免轮子悬空，保证码盘对轮子转动角度记录的有效性。驱动方式为双轮差速驱动，前两轮为驱动轮,后轮使用万向轮。底盘四轮的独立悬挂，应用特质铰链加上减震器构成，前后轮均为橡胶充气轮，较刚性轮有较好的柔性，结合悬挂之后可轻松越过街道上常见的 6-8CM 高的减速带。底盘依靠铝制管材切割，焊接，搭建而成，成本较低，而且框架结构也使车体更轻，可随意安置电路元器件，改动更加容易。



图 4 四轮导航机器人
Fig. 4 Four-wheeled navigation robot

180 **3.2 代价地图模块测试**

本次测试针对优化后的代价地图模块进行，验证优化后代价地图的功能是否正常，能否达到预期效果。其具体测试用例如下：

测量四轮前驱机器人的尺寸数据,根据尺寸数据修改代价地图相关参数,主要包括机器人轮廓、机器人内切圆及外接圆半径、膨胀半径等。代价地图模块应在静态地图基础上根据各个参数生成代价地图。

测试改进后代价地图能否正常加载静态地图;

测试改进后代价地图是否存在与参数设置对应的膨胀区域;

测试膨胀区域是否有明显分层;

测试动态障碍物能否生成正常的代价地图;

测试代价地图生成时间是否存在异常。

经过多次测试,上述测试用例通过率全部在 95% 以上,代价地图模块各个功能得到验证,具备可用性与稳定性。

3.3 导航算法测试

本次测试针对优化后的 dwa 局部路径规划算法进行,验证在改进的代价地图基础上 dwa 算法的兼容性,并对 dwa 算法的两种极端情况处理进行验证。为比较四轮前驱结构和六轮结构的效果差异,在测试过程中对两种结构的机器人进行同条件测试,尽可能控制变量唯一。其具体测试用例如下:

室内环境下对两种结构机器人进行直线行驶,测试行驶过程是否稳定高效;

室内环境下对两种结构机器人进行左右两侧直角转弯,测试转弯过程是否顺畅,是否出现撞墙等情况;

室内环境进行较长时间自主导航,测试是否出现异常;

室内环境下进行静态障碍物躲避,事先放置障碍物于机器人行驶路线中,测试两种结构的机器人的避障过程是否稳定顺畅,有无撞障碍物等情况;

室内环境下进行动态障碍物躲避,在机器人行驶过程中添加行人作为动态障碍物,测试避障过程是否稳定顺畅;

测试突然出现在机器人前方较近距离的情况下,机器人能否及时停住,能否按照特殊情况处理方式进行躲避;

测试机器人后退过程中是否检测周围障碍物信息,是否出现撞击障碍物等情况。

图 5 为室内环境下转向测试时的可视化数据。

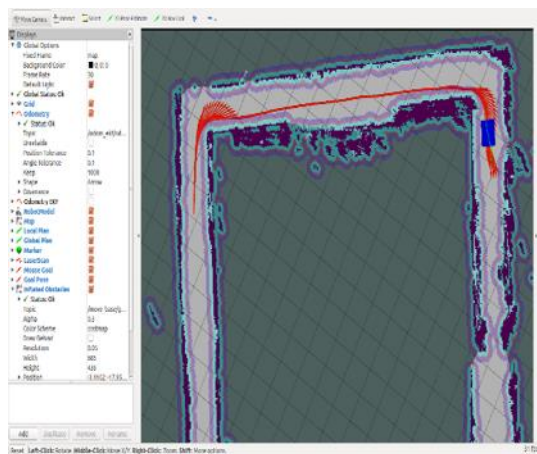


图 5 室内转向测试

Fig. 5 Indoor steering test

经过多次测试上述测试用例通过率全部超过 90%，导航过程的流畅性明显优于未经优化的版本，验证了本方案的先进性，达到了预期效果，满足了同时适应多种机械结构机器人的需求。

4 结论

本自主导航技术仍然存在诸多问题，离广泛落地还有一些距离，本文结合实验室在自主导航机器人领域的研究成果与课题积累，针对机械结构差异导致自主导航效果不佳的问题提出解决方案，使日后的机器人应用开发不必受限于软件缺陷，为此主要进行了以下研究工作：

（1）改进 ROS 平台中局部路径规划的代价地图模型以适应不同结构机器人导航需求，对改模型各个层级进行拆解分析，并结合改进需求对膨胀层进行分区域化改进，分为行驶自由区和行驶受限区，可根据机器人硬件参数对膨胀区域进行调整，以适应各种机械机构。

（2）优化 ROS 中局部路径规划算法，以兼容改进后的代价地图模型，并结合实际应用中的常见问题对算法逻辑进行改进，包括：添加机器人后方的速度采样，对后方障碍物信息进行采集，以免发生意外；对突然出现的近距离障碍物添加避障策略，原地急停，等待一段时间后再进行避障，提高了机器人行驶过程的安全性、高效性。

[参考文献] (References)

[1] 雷碧波. 基于 ROS 平台的室内定位算法设计与实现[D].浙江理工大学,2017.

[2] 黄开宏,杨兴锐,曾志文,卢惠民,郑志强.基于 ROS 户外移动机器人软件系统构建[J].机器人技术与应用,2013(04):37-41+44.

[3] 曹其新,黄先群,蒋宇捷,朱笑笑.动态环境中移动机器人多状态转换自主导航[J].华中科技大学学报(自然科学版),2019,47(10):22-27.

[4] 方正,吴成东.自主导航：赋予移动机器人智能感知与运动的能力[J].自动化博览,2019(08):68-72.

[5] 周兰兰.基于 ROS 的移动机器人多目标路径规划[J].电脑知识与技术,2019,15(21):176-177+187.

[6] 杨记周. 一种室内移动机器人定位和路径规划的算法优化[D].中国科学技术大学,2019.

[7] 宋浩.自主导航车轨迹跟踪控制方法研究[J].技术与市场,2019,26(08):102-103.

[8] 郑建杰,张海涛,陈寅.基于激光 SLAM 的导航车软硬件系统[J].电子技术与软件工程,2019(15):72-74.

[9] 梁爽. 智慧工厂中多移动机器人路径规划研究[D].上海交通大学,2017.

[10] 郑潇峰. 基于 ROS 的移动机器人室内激光导航研究[D].重庆邮电大学,2017