

基于激光 SLAM 的展厅服务机器人 自主导航研究



吴志刚*, 蔡浩聪, 朱耀晖, 陈敏

江西理工大学能源与机械工程学院, 江西南昌 330013

摘要: 作为引导和介绍参观者进入展厅的重要工具, 机器人需要面对可能存在的复杂环境和不断移动的参观者。首先, 它需要具备能够实现准确定位、导航以及动态避障的能力。近年来, 同步定位和建图技术(Simultaneous localization and mapping, SLAM)已被广泛用于解决未知室内环境中的定位和导航问题。本研究基于机器人操作系统(Robot operating system, ROS), 通过 2D 激光雷达采集环境信息, 结合服务机器人的惯性测量单元 (Inertial measurement unit, IMU)、里程表和其他传感器, 通过 Cartographer SLAM 算法绘制展厅环境图。实现了路径规划和动态避障功能。选择了全局路径规划算法, 以及局部路径规划的动态窗口法 (Dynamic Window Approach, DWA) 算法进行路径规划。环境地图的构建、路径规划和动态避障功能通过模拟和真实环境得到了验证。在定位测试中, 当机器人以 0.5 米/秒的速度移动时, 机器人与目标点的平均位置偏差和标准偏差 (Standard deviation, SD) 分别小于 7cm 和 3cm。平均航向偏差小于 9°, SD 小于 3°。该机器人的定位和导航能够很好地满足展馆服务机器人的工作要求。

关键词: 展厅; 服务机器人; 导航; 雷达; 定位与建图

DOI: [10.57237/j.mse.2022.01.004](https://doi.org/10.57237/j.mse.2022.01.004)

Research on Autonomous Navigation of Showroom Service Robot Based on ROS and Laser SLAM

Wu Zhigang*, Cai Haocong, Zhu Yaohui, Chen Min

School of Energy and Mechanical Engineering, Jiangxi University of Science and Technology, Nanchang 330013, China

Abstract: As an important tool for guiding and introducing visitors to the exhibition hall, the robot needs to confront the complex environment that may exist and the constantly moving visitors. First of all, it is needed to have the ability to enable accurate positioning and navigation as well as dynamic obstacle avoidance. In recent years, simultaneous localization and mapping (SLAM) technology has been widely used to address localization and navigation issues in unknown indoor environments. Based on the robot operating system (ROS), this study collects environmental information by 2D LiDAR, and then combines the service robot's Inertial measurement unit (IMU), odometer, and other sensors to map the exhibition hall environment by the Cartographer SLAM algorithm. Path planning and dynamic obstacle avoidance functions are also implemented, in path planning, the algorithm is selected for global path planning and the Dynamic Window Approach (DWA) algorithm for local path planning. The environment map construction, path planning and dynamic obstacle avoidance functions are verified by simulation and in real environment. The average position deviation and standard

基金项目: 江西省教育厅科技一般项目 (GJJ200833).

*通信作者: 吴志刚, wzgang2017@163.com

收稿日期: 2022-12-04; 接受日期: 2022-12-29; 在线出版日期: 2023-01-05

<http://www.mechscieng.com>

deviation (SD) of the robot from the target point were less than 7 cm and less than 3 cm, respectively, when the robot was set to move at 0.5 m/s for the positioning test. The average heading deviation was less than 9° and SD was less than 3°. The robot's positioning and navigation can well meet the working requirements of the pavilion service robot.

Keywords: Exhibition Hall; Service Robot; Navigation; LIDAR; SLAM

1 引言

服务机器人一直是机器人研究领域的热点, 机器人通过各种复杂的传感器系统获取外部压力、声音、图像等信息, 感知、识别和适应外部环境。Hu 等[1]针对医疗服务机器人在不规则场景下的精准服务和应急救援需求, 实现了机器人在服务场景下的导航和路径规划。大型商场中的 OSHbot 导购机器人[2], 它使用激光传感器进行自主定位和导航, 引导顾客。以及餐厅、家庭和展览馆, 实现物品运送、娱乐、清洁和顾客引导等服务任务。显然, 服务机器人需要具备多功能性, 如除了定位和导航之外, 还需要具备语音交互等功能, 以及在不确定环境中自主完成任务的能力。系统集成也是关键因素之一[3]。一项关键技术是机器人的 SLAM 技术[4], Ekvall 等[5]解决了一个与场景相关的问题, 自助服务机器人应该检测和识别物体, 并使用 SLAM 来构建环境地

图。导航也是服务机器人的关键能力之一, Althaus 等[6]所开发的机器人被加入到人群中作为观众, 实验表明, 当人群形成变化时, 机器人能够重新定位并实现自主导航。

在现有的 SLAM 算法中, 通常依赖于 LiDAR、RGB-D 传感器或摄像头[7], 本文主要研究 LiDAR 在服务机器人中的使用。展厅服务机器人通过计算其与障碍物的距离, 并利用从传感器扫描得到的环境数据估计机器人在未知空间的位置姿态, 构建环境地图。同时, 在满足距离、计算时间、通信延迟和能源消耗等优化条件的前提下。在环境复杂、随时可能出现的动态障碍物的展厅中, 利用路径规划算法, 规划出服务机器人从当前位置到目标位置的安全路径, 以满足展厅服务中的引导引入任务。

2 系统硬件搭建



图 1 服务机器人系统架构

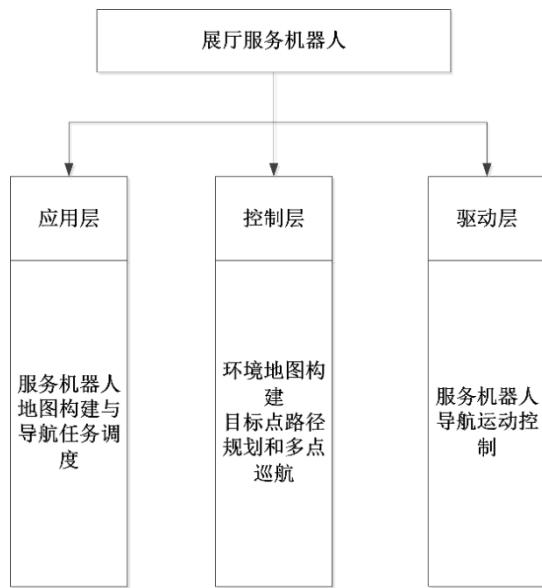


图 2 展厅服务机器人软件架构

2.1 系统硬件架构

本文所研究的机器人可分为传感器系统、控制系统、驱动系统和执行机构四个部分。传感器系统主要由编码器、单线 LIDAR、IMU 组成，控制系统采用 Jetson Nano 和 NVIDIA 的 PC，驱动系统由电池、Arduino、电机和电机驱动模块组成，主体采用亚克力板和铝材组装，机器人的

行走由两个直流电机驱动主动轮和两个保持平衡的万向轮作为机器人的执行器。系统结构如图 1 所示。

2.2 系统软件架构

算法在一个自行设计的展厅服务机器人上进行了测试。采用 ROS 分布式设计框架，允许在运行时进行单独设计，编译框架中的各个功能模块。机器人本体控制系统采用 Jetson Nano，整个软件系统基于 Ubuntu 18.04 和 Melodic 版本的 ROS 系统，如图 2 所示。PC 端用于处理复杂算法和图形显示和功能操作的远程实现[8-9]。

3 环境地图构建

考虑到本研究使用的是具有 2G 显存的 Jetson 纳米作为主控制器，其算法功率水平有限，并且通过比较模拟环境、真实环境和中央处理单元（CPU）利用率[10-12]，本文选择 Cartographer SLAM 算法，其能够生成分辨率 $r = 5 \text{ cm}$ 的光栅图。通过完成对附近子图的扫描，在前端匹配最新的 LiDAR 扫描，并将其插入到短时间内获得最佳插入位置的子图中[13]。位置估计误差是在整幅图的扫描匹配过程中随着时间的推移而逐渐积累的。为了消除这个错误，通过有约束的环回检测在后台线程中进行优化[12]。算法流程图如图 3 所示。

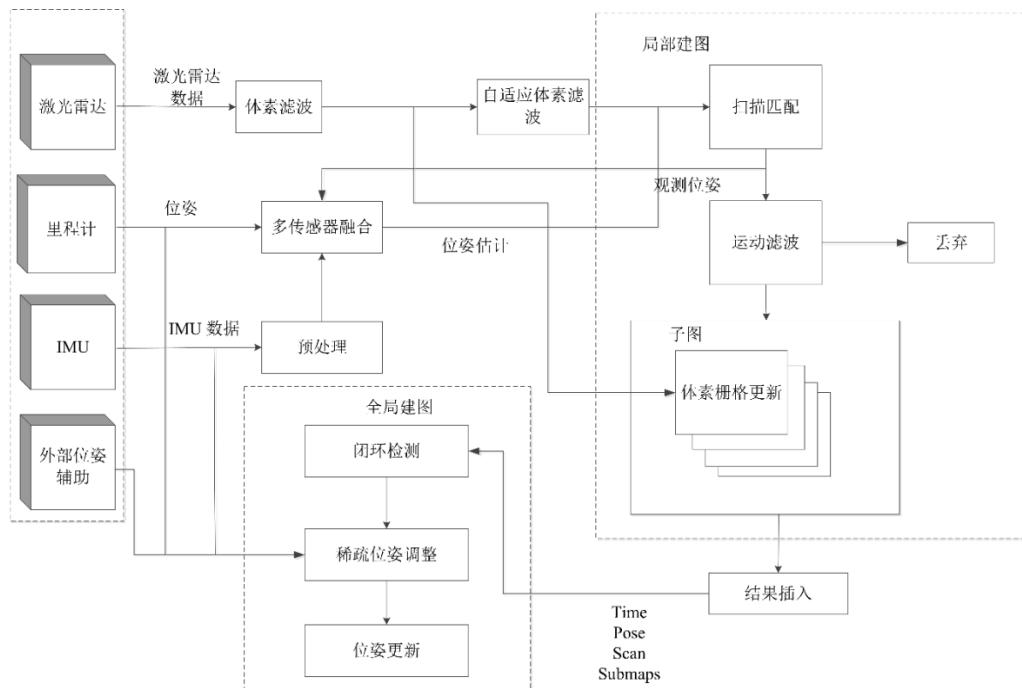


图 3 Cartographer 算法流程图

局部 SLAM 和全局 SLAM 的综合使用是 Cartographer 实现二维定位和地图构建的方法。它们都强调位置向量 $\xi = [\xi_x, \xi_y, \xi_\theta]^T$ 的优化, 并通过雷达观测或扫描 (x, y) 的平移和 δ_θ 旋转来实现。每一个连续扫描都匹配到世界坐标系上的一个子映射。在本地 SLAM 中使用, 扫描匹配过程会产生误差的积累, 这些误差可以通过全局优化过程来消除。在局部建图中, 首先, k 以雷达旋转中心为坐标系, 取一次雷达扫描转弯获得的距离点 $\{h_k\}, k=1, 2, \dots, L, h_k \in \mathbb{R}^2$; 在局部子图中, 使用第一帧的雷达位姿作为参考, 后面添加的雷达位姿用相对位移矩阵 $T_\xi = (R_\xi, t_\xi)$ 表示。因此, 将雷达帧内的数据点 h_k 转化为局部子图坐标帧, 可表示为式(1)。

$$T_\xi \cdot h_k = \begin{bmatrix} \cos \xi_\theta & -\sin \xi_\theta \\ \sin \xi_\theta & \cos \xi_\theta \end{bmatrix} h_k + \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} \quad (1)$$

通过式(1), 新的雷达数据将转换到子地图坐标系中, 此时新的雷达数据点将覆盖子地图的一些栅格 $\{M_{old}\}$, 每个栅格以三种状态存在: 未知、未命中和命中。雷达扫描点覆盖的网格应被占用, 而雷达扫描束开始和结束的区域内不存在障碍物, 该区域覆盖的网格应不被占用。由于雷达扫描分辨率和范围的限制, 未被雷达扫描点覆盖的网格应为未知网格。由于子图中的网格可能被多个雷达扫描帧覆盖, 因此需要迭代更新网格的状态。

在全局建图中, 采用的是稀疏位姿图来做全局优化。设所有雷达扫描帧对应的机器人全局位姿 $\Xi^s = \{\xi_j^s\}, j=1, 2, \dots, L, n$ 和所有局部子图对应的全局位姿 $\Xi^m = \{\xi_i^m\}, i=1, 2, \dots, m$ 通过 Scan-to-map Matching 产生的局部位姿 ξ_{ij} 进行关联, 数学表达式如下所示:

$$\arg \min_{\Xi^s, \Xi^m} \frac{1}{2} \sum_{ij} \rho(E^2(\xi_i^m, \xi_j^s, \Sigma_{ij}, \xi_{ij})) \quad (2)$$

其中,

$$E^2(\xi_i^m, \xi_j^s, \Sigma_{ij}, \xi_{ij}) = e(\xi_i^m, \xi_j^s, \xi_{ij})^T \sum_{ij}^{-1} e(\xi_i^m, \xi_j^s, \xi_{ij}) ;$$

$$e(\xi_i^m, \xi_j^s, \xi_{ij}) = \xi_{ij} - \begin{bmatrix} R_{\xi_i^m}^{-1} \cdot [t_{\xi_i^m} - t_{\xi_j^s}] \\ \xi_{i:\theta}^m - \xi_{j:\theta}^s \end{bmatrix} \text{。式中, } j \text{ 是雷达扫描帧的序号, } i \text{ 是局部子图的序号。}$$

描帧得序号, i 是子图的序号。而雷达扫描数据在局部子图中还具有局部位姿, 比如 ξ_{ij} 表示序号为 j 的雷达扫描帧在序号为 i 的局部子图中的局部位姿。而损失函数 ρ 用于惩罚那些过大的误差项。

本文考虑到应用场景为展厅, 没有楼梯, 地面也平滑。加上使用里程计和 IMU, 相比 Gmapping 算法, 基于图优化 Cartographer 算法的建图效果要好, 因此选择 Cartographer SLAM。

4 路径规划

路径规划是移动机器人自主导航的重要问题之一, 通常在给定的光栅地图上进行。通过遍历整个栅格图, 找到从起始点到目标点的路径, 考虑路径的性能, 保证最优路径的距离、平滑度、能量消耗和各种约束条件, 使机器人无碰撞运动。然而, 在真实场景中运行时, 由于机器人传感器的准确性和环境的可变性, 真实的环境和位置信息会有一些误差[14]。

4.1 全局路径规划

基于构建的地图完成机器人在展厅的路径规划, 利用基于 A^* 算法[15]的图搜索结构进行全局路径规划。它将经典的 Dijkstra 算法与启发式算法中的最佳优先搜索 (BFS) 相结合。根据代价函数进行搜索, 将满足代价函数的目标点作为下一个被搜索点, 继续上述过程, 搜索最优路径[16]。算法如图 4 所示。首先设定地图和机器人的当前点和目标点, 然后对地图进行初始化。取第一个位置, 判断是否成功找到解并输出路径。将下一个可达位置添加到开放列表中, 并判断记录的最短路径是否需要更新。记录最短路径经过的最后一点, 并遍历或更新当前已知的最短路径。

成本评价函数表示为:

$$f(n) = h(n) + g(n) \quad (3)$$

式中, $f(n)$ 为当前位置 n 的成本评价函数, $g(n)$ 为机器人从初始位置到当前位置 n 的实际成本, $h(n)$ 为机器人从当前位置 n 到目标位置的估计成本。 $h(n)$ 的选择直接影响到 A^* 算法的成功率和准确率, 搜索的效率与实际值的接近程度成正比。本文讨论的是栅格地图模型, 且为两轮驱动, 只能在四个方向移动时,

对于 $h(n)$ 距离模型采用曼哈顿距离。曼哈顿距离由目标节点与当前节点在 x 坐标差值与 y 坐标差值绝对值的和表示：

$$h(n) = |n.x - g.x| + |n.y - g.y| \quad (4)$$

式中 n 为机器人当前节点， g 为机器人目标节点。

4.2 局部路径规划

在展厅服务机器人的实际应用场景中，经常会有机器人四处走动，这可能会阻碍机器人预先规划的路线。为了确保机器人的安全和成功完成任务，需要局部实时避障，因此选择 DWA 算法来实现展厅机器人的局部实时避障功能。算法如图 5 所示。首先，通过类加载模块加载 BaseLocalPlanner 的子类，并调用初始化函数以获得机器人的当前位置等。然后，通过收集传感器每个维度的线速度和角速度，并将其放入 sample_params 容器，对速度样本进行采样。基于采样的速度获得相应的运动轨迹，并通过轨迹评估函数评估每个轨迹，最后发布局部规划。

Algorithm 1: A^* algorithm

```

1: Input: map, current_position, target_position
2: Output: the shortest path
3: Init openlist[], closedlist[]
4: board = Board(map,target_position)
5: board.GetMsg(current_position).IsUnk = 0
6: Openlist.append(current_position)
7: While (openlist! = []):
8:   current_position = openlist[0]
9:   openlist.remove(current_position)
10:  If (current_position == target_position) Then
11:    print("Successfully found the solution ")
12:    return
13:  For i in [current_position.x-1,current_position.x+1]: Do
14:    For j in [current_position.y-1,current_position.y+1]: Do
15:      If (IsInBoard(i,j)): Then
16:        new_G=board.GetMsg(current_position).G+14
17:        If (board.mapx[i][j].IsUnk): Then
18:          board.mapx[i][j].IsUnk = 0
19:          openlist.append(Position(i,j))
20:          board.mapx[i][j].parent = current_position
21:          board.mapx[i][j].G = new_G
22:
23:        If (board.mapx[i][j].G > new_G): Then
24:          board.mapx[i][j].parent = current_position
25:          board.mapx[i][j].G = new_G
26:        End If
27:      End If
28:    End If
29:  End For
30: End For
31: openlist.sort(key = lambda elem : board.GetMsg(elem).GetF())
32: End While

```

图 4 A^* 算法伪代码

Algorithm 2: DWA algorithm

```

1: Input: robotPose, robotGoal, robotModel
2: Output: the best  $v$  and  $\omega$ 
3: desired  $v$  = calculate  $v$  (robotPose, robotGoal)
4: laserscan = readScanner()
5: allowable_v = generateWindow(robot  $v$ , robotModel)
6: allowable_ω = generateWindow(robot  $\omega$ , robotModel)
7: While (not arrived the goal )
8:   For  $v$  in allowable_v Do
9:     For  $\omega$  in allowable_ω Do
10:      dist = find_dis( $v$ ,  $\omega$ , laserscan, robotModel)
11:      breakDist = calculateBreakingDistance( $v$ )
12:      If (dist > breakDist) Then
13:        heading = hDiff(robotPose, goalPose,  $v$ ,  $\omega$ )
14:        clearance = (dist - breakDist)/(dmax - breakDist)
15:        cost = costFunction(heading, clearance, abs(desired_v -  $v$ ))
16:        If (cost > optimal) Then
17:          best_v =  $v$ 
18:          best_ω =  $\omega$ 
19:          optimal = cost
20:        End If
21:      End If
22:    End For
23:  End For
24:  set robot trajectory to best_v, best_ω
25: End While

```

图 5 DWA 算法的伪代码

通过 DWA 算法，窗口内的速度空间(v, ω)可以通过对速度空间的离散化将其分解为有限的组合。在获得机器人速度样本后，根据机器人采样速度模拟推导出相应的运动轨迹，并通过轨迹评估函数评估每个轨迹，如公式(5)所示：

$$G(v, \omega) = \alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega) \quad (5)$$

式中， α 、 β 、 γ 是每一项的权重因子， $heading(v, \omega)$ 表示评价机器人采取 (v, ω) 速度空间时，所在路径移动方向与目标的角度值 θ 的函数。 $vel(v, \omega)$ 表示运动轨迹的速度大小，运动轨迹越快到达目标，评分越高。可表示为：

$$heading(v, \omega) = 1 - \frac{\theta}{\pi} \quad (6)$$

$$vel(v, \omega) = \frac{(v, \omega)}{(v_{\max}, \omega_{\max})} \quad (7)$$

5 实验

5.1 展厅场景实测

实验场景选择校史馆进行，其面积 $15 \times 8m^2$ 。用

上述 SLAM 算法构建环境地图, 如图 6 所示。当移动机器人穿过校史馆并返回起点时, 地图从灰色变为白色, 并构建了完整的地图。如图 7 所示, 当测试路径规划和动态避障时, 当机器人靠近行人时, 地图上会出现黑色障碍物。机器人重新规划路径, 最终到达目标点。全局路径规划算法使用基于图搜索结构的 A^* 算

法, 而局部路径规划法则使用 DWA 算法。在服务机器人的路径规划中, 局部路径基本上符合全局路径, 当行人突然出现在展厅服务机器人路径前方时, 可以很好地完成自主避障, 并改变全局路径规划路线以到达目标点。



图 6 服务机器人对展厅进行环境建图

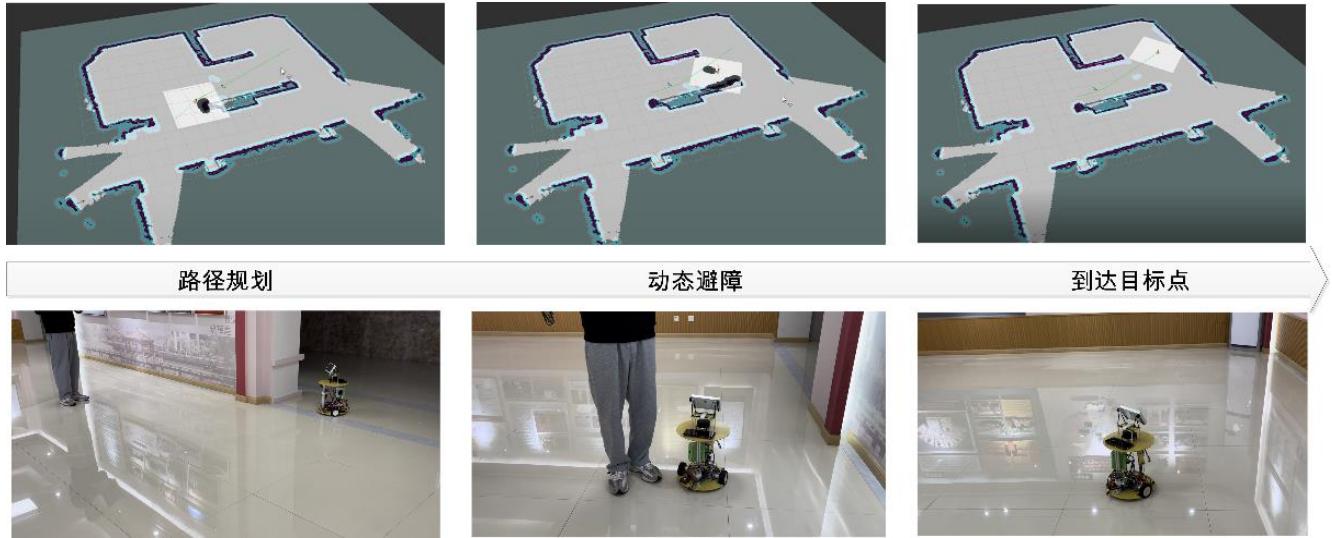


图 7 展厅环境下服务机器人路径规划和动态避障

5.2 展厅场景实测

通过获取机器人的位置和姿态来进行定位精度测试是证明机器人导航系统准确性和可靠性的有效手段。通过 `move_base_simple/goal` 获取导航路径中随机选择的四个目标点的位置和姿态信息, 然后通过订阅用 `python` 编写的 `move_base` 中的反馈和结果主题来获取

机器人的位置和姿势信息, 以到达目标点。通过将机器人的速度设置为 0.5m/s , 实验测试了机器人的定位精度。此外, 为了避免实验期间的累积误差, 当执行 10 次完整导航和定位时, 必须返回初始位置并在每次结束时重新开始测试。如图 8 和图 9 所示, 对展厅服务机器人进行了 10 次定位测试, 记录了每个点位置的信息以及四元数信息。将记录的数据与设定的目标点数据进行比较, 得出位置偏差和航向偏差分别不超过

7cm 和 9°。标准偏差分别不超过 3cm 和不超过 3°。从图 8 可以看出, 当目标点 2 出现异常时, 服务机器人进行了测试。分析目标位置后, 目标点 1 和 4 位于展厅入口, 左右两侧都有墙壁, 目标点 3 靠近展厅墙壁, 周围较窄。而目标点 2 位于展厅中心, 因此目标点 1、3 和 4 周围的环境结构特征信息比目标点 2 更多。虽然

存在异常, 但该服务机器人的导航和定位精度可以满足其在室内环境中的任务要求。对于使用粒子滤波器跟踪机器人姿态的定位系统, 考虑了机器人现有 Jetson nano CPU 的消耗以及粒子数量对定位精度的影响。将定位算法中 `max_particles` 和 `min_particle` 的值分别设置为 5000 和 500, 机器人在连续测试下具有良好的性能。

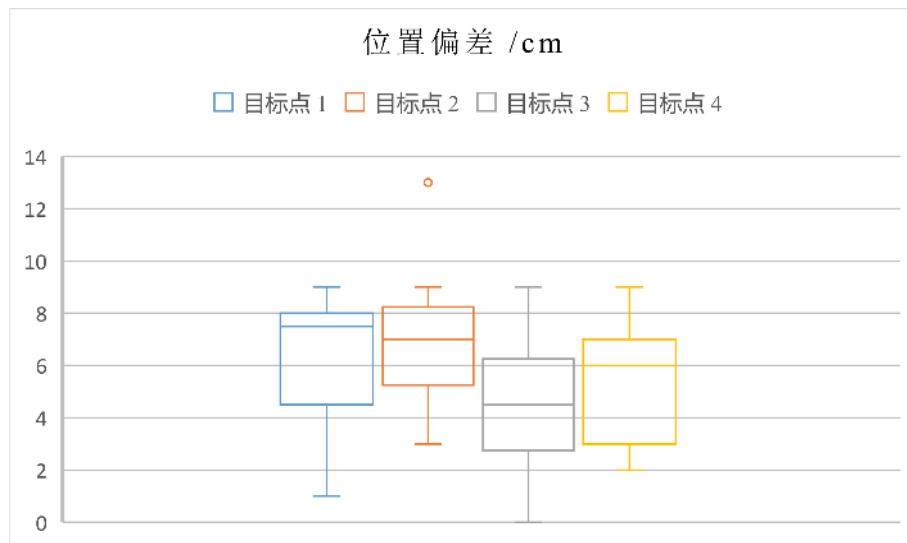


图 8 位置偏差

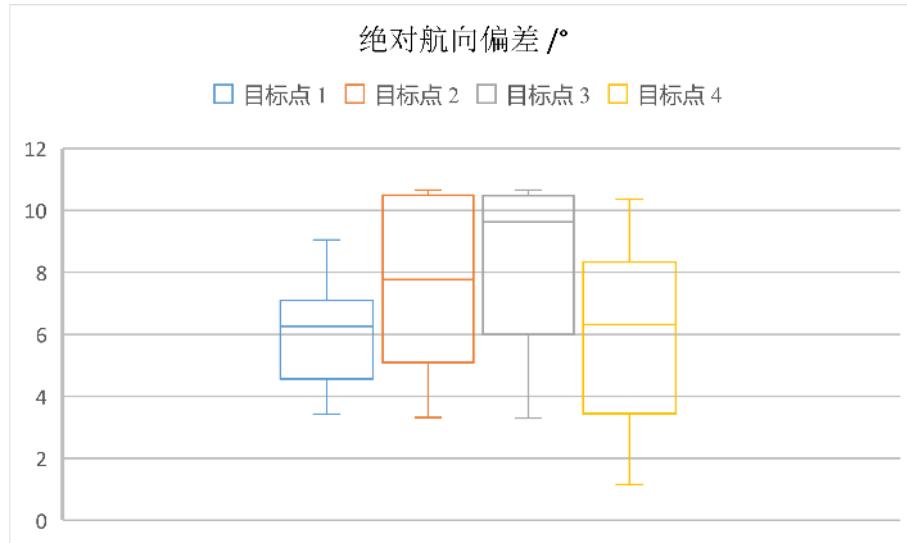


图 9 绝对航向偏差

6 结论

本文研究了基于 ROS 和 SLAM 的移动机器人在展厅服务中的应用。环境信息通过 2D LiDAR 获取, 而数据通过融合编码器获取的里程表信息与 IMU 融合。主控制器为 Jetson nano, 软件核心控制系统通过 ROS 基于 Ubuntu

开发, 通过分布式节点通信实现信息交互。在实际实验中使用 Cartographer SLAM 算法进行地图构建。路径规划是通过结合全局路径规划和局部路径规划算法来执行的, 其中全局路径规划算法使用基于图搜索结构的 A^* 算法, 而局部路径规划算法则使用 DWA 算法。在展厅测试后发现, 平均位置偏差和平均航向偏差分别不超过 7 厘米和 9°,

满足室内展厅服务机器人的运动要求。解决室内展厅服务机器人的定位和导航问题。将来的工作中,将现有的地图构建和导航算法应用于不同类型的移动机器人,并根据不同的机器人在不同的工作条件下制定相应的策略。此外,后续还与人工智能相结合,实现机器人的语音交互和行人识别。

参考文献

- [1] HU Y, WANG Y, WANG S, et al. Fusion Key Frame Image Confidence Assessment of the Medical Service Robot Whole Scene Reconstruction [J]. *Journal of Imaging Science and Technology*, 2021, 65 (3): 1-9.
- [2] 王玉山, 王伟, 李仲阳, 等. 浅谈服务机器人的应用现状和发展前景 [J]. *机电工程技术*, 2017, 46 (2): 94-96.
- [3] Lee I. Service Robots: A Systematic Literature Review [J]. *Electronics*. 2021, 10 (21): 2658.
- [4] Luo, R. C., Lai, C. C. Enriched Indoor Map Construction Based on Multisensor Fusion Approach for Intelligent Service Robot. *IEEE Trans. Ind. Electron*, 2012, 59 (8): 3135-3145.
- [5] Ekwall, S.; Kragic, D.; Jensfelt, P. Object detection and mapping for service robot tasks [J]. *Robotica* 2007, 25 (2): 175-187.
- [6] Althaus P, Ishiguro H, Kanda T, et al. Navigation for human-robot interaction tasks [C]. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. IEEE, 2004: 1894-1900.
- [7] Li. B, Wang. Y, Zhang. Y, et al. Gp-slam: laser-based slam approach based on regionalized gaussian process map reconstruction [J]. *Autonomous Robots*, 2020, 44 (6): 947-967.
- [8] 孙博雅. 移动机器人 SLAM 技术 [J]. *电子技术与软件工程*, 2018 (02): 95.
- [9] LIN ZC, ZHENG S. Research on image matching and camera pose resolution in mobile robot vision SLAM process [J]. *Machine Design and Manufacturing Engineering*, 2017, 46 (11): 13-18.
- [10] Santos J M, Portugal D, Rui P R. An Evaluation of 2D SLAM Techniques Available in Robot Operating System [C]// 11th IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR 2013). IEEE, 2013: 1-6.
- [11] Jiang S, Wang S, Yi Z, et al. Autonomous Navigation System of Greenhouse Mobile Robot Based on 3D Lidar and 2D Lidar SLAM [J]. *Frontiers in Plant Science*, 2022, 13: 815218.
- [12] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM [C]. 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 1271-1278.
- [13] Li Z, Xu B, Wu D, et al. A Mobile Robotic Arm Grasping System with Autonomous Navigation and Object Detection [C]. 2021 International Conference on Control, Automation and Information Sciences (ICCAIS) 2016: 543-548.
- [14] 王消为, 贺利乐, 赵涛. 基于激光雷达与双目视觉的移动机器人 SLAM 研究 [J]. *传感技术学报*, 2018, 31 (3): 394-399.
- [15] CAO ZG, JIANG SW, ZHANG J, et al. A Unified Framework for Vehicle Rerouting and Traffic Light Control to Reduce Traffic Congestion [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2017, 18 (7): 1958-1973.
- [16] 宋晓茹, 任怡悦, 高嵩, 等. 移动机器人路径规划综述 [J]. *计算机测量与控制*, 2019, 27 (4): 1-5, 17.