

# 基于ROS的全向移动机器人导航系统设计及仿真

田塘, 李涛, 邓春燕, 胡桂川\*, 赵杭, 糜斐

重庆科技学院机械与动力工程学院, 重庆

收稿日期: 2022年1月5日; 录用日期: 2022年2月16日; 发布日期: 2022年2月23日

## 摘要

动态环境复杂多变, 机器人如何在动态环境下实现自主移动是一个难题。本文以麦克纳姆轮底盘为对象, 基于机器人操作系统(ROS), 设计并实现了开放性好和代码复用率高的全向移动自主导航系统。首先, 先对麦轮底盘进行URDF建模, 运动学分析, 根据得到速度关系重写新的底盘ROS节点; 其次结合MOVE\_BASE框架, 搭建自主导航系统, 利用SLAM技术构建二维栅格地图, 结合AMCL和路径规划算法的融合导航算法开展自主导航测试。实验结果表明, 该方法可以实现机器人的自主移动和避障要求, 提升路径规划效果, 验证了机器人自主导航的可行性。

## 关键词

ROS, 自主导航, 全向移动, 自主避障

# Design and Simulation of Omnidirectional Mobile Robot Navigation System Based on ROS

Tang Tian, Tao Li, Chunyan Deng, Guichuan Hu\*, Hang Zhao, Fei Mi

School of Mechanical and Power Engineering, Chongqing University of Science and Technology, Chongqing

Received: Jan. 5<sup>th</sup>, 2022; accepted: Feb. 16<sup>th</sup>, 2022; published: Feb. 23<sup>rd</sup>, 2022

## Abstract

Dynamic environment is complex and changeable, so it is a difficult problem for robot to realize

\*通讯作者。

文章引用: 田塘, 李涛, 邓春燕, 胡桂川, 赵杭, 糜斐. 基于 ROS 的全向移动机器人导航系统设计及仿真[J]. 人工智能与机器人研究, 2022, 11(1): 46-55. DOI: 10.12677/airr.2022.111006

**autonomous movement in dynamic environment. Based on the Robot Operating System (ROS), this paper designs and implements an omnidirectional mobile autonomous navigation system with good openness and high code reuse rate based on the Mecanum wheel chassis. Firstly, URDF modeling and kinematics analysis were carried out for wheat wheel chassis, and new chassis ROS nodes were rewritten. Secondly, the autonomous navigation system was built by combining MOVE\_BASE framework, 2D raster map was constructed by SLAM technology, and autonomous navigation test was carried out by combining AMCL and path planning algorithm. Experimental results show that this method can achieve the requirements of autonomous robot movement and obstacle avoidance, improve the effect of path planning, and verify the feasibility of autonomous robot navigation.**

## Keywords

ROS, Autonomous Navigation, Omnidirectional Movement, Autonomous Obstacle Avoidance

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

近年来,随着智能制造、网络信息和高集成度芯片等新兴技术的飞速发展,移动机器人在智能仓储、智能工厂和物流运输等行业内得到了广泛的应用[1]。但在非线性结构环境下,机器人容易受到环境、随机障碍物等外界因素等影响,从而导致自主导航失效。而机器人在进入没有先验信息的未知环境下,需要解决三个核心问题才能实现自主导航,即地图建模、自我定位和路径规划[2]。针对前两个问题,同步定位与建图(SLAM)技术应运而生,通过机器人自身携带的传感器,感知周围环境、获取深度信息以实现位置环境地图的重建。目前根据后端优化主要分为基于滤波的激光 SLAM 和基于图优化的视觉 SLAM。视觉 SLAM 通常采用单目相机、双目相机和深度相机传感器感知环境,代表算法有 VINS MONO、ORB\_SLAM 系列;激光 SLAM 则采用毫米级激光或激光雷达;代表算法有 EKF-SLAM、Gmapping。

路径规划作为自主导航的核心,分为全局路径规划和局部路径规划。常用的全局路径规划算法有 A\* 算法、蚁群算法以及 RRT 算法,针对原始算法存在的不足,国内外学者也做了大量的研究。Koenig S [3] 等人提出了一种即实时自适应 A\* 算法,可以加快重复 A\* 搜索速度;李静[4]等人提出了一种改进蚁群算法,通过概率选择函数,栅格法地图建模和改进信息素等方式提高了路径的全局规划能力和搜索能力;Gammell J D [5]等人提出了 Informed RRT\* 算法,在保证搜索路径质量的前提下缩短了 RRT 原始算法的路径搜索时间。

基于此,本文基于机器人操作系统(Robot Operating System, ROS)设计了一个由麦克纳姆轮驱动的全向移动导航系统。首先在 GAZEBO 中建立 URDF 机器人模型和地图环境,针对传统 IMU 存在的严重误差累积问题,融合 IMU 和 AMCL 信息增强定位;其次结合 MOVE\_BASE 导航框架,在 GAZEBO 以及 RVIZ 中进行实验测试,使用基于 ROS 的 A\* 导航算法和 TEB 矩优化算法实现自主移动及动态避障。

## 2. 机器人建模与底盘节点

URDF 全称(United Robotics Description Format)统一机器人描述格式,是一个 XML 语法框架下用来

描述机器人的语言格式[6]。为了更为直观地进行仿真和观察机器人的运行状态,使用 URDF 文件对机器人进行建模。通过定义机器人的关节、自由度,世界坐标、惯性矩阵、碰撞属性、质量等属性来尽可能地模拟真实机器人。

### 1) 麦轮底盘设计

编写底盘的 URDF 文件,使其能够接收由导航节点发布的 `cmd_vel` 指令、发布里程计数据。首先在 SOLIDWORKS 中创建机器人模型,并计算惯性参数和碰撞属性;其次在 URDF 文件中为车体框架和麦轮添加惯性参数和碰撞属性,对麦轮在额外添加传动装置和减速比;最后通过控制器插件(ROS\_Control)配置机器人的移动模式,完成底盘设计。URDF 结构图如图 1 所示,其中 `base_footprint` 为车体中心投影点,起定位作用,`base_link` 表示车体部件,`link1~link4` 表示麦轮部件,`laser` 表示激光雷达,蓝色字体部分表示各关节状态,`joint1~joint4` 为转动关节,其余为固定关节。

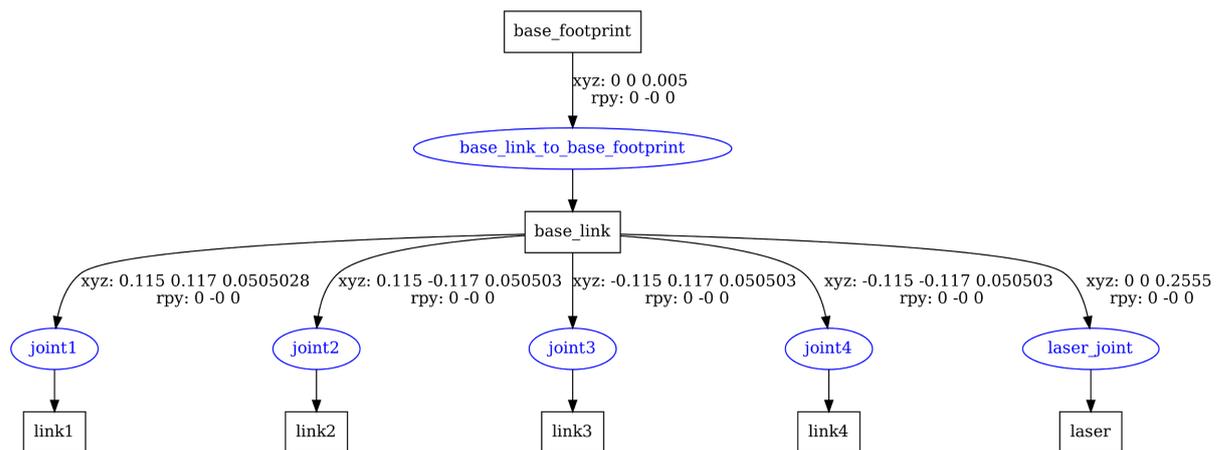


Figure 1. URDF structure diagram

图 1. URDF 结构图

### 2) 底盘节点设计

当前普通差速轮式机器人在狭窄空间环境或者复杂结构环境中表现不佳,往往出现移动困难、转向失败等问题。其原因为常见的差速轮子只有 X 方向的线速度和 Z 轴的角速度,无法进行全向移动,导致在狭窄空间中卡死。而麦克纳姆轮由于包含三个自由度(X、Y 方向上的线速度以及 Z 轴的角速度),转向时可以实现零转弯半径的旋转运动,适用性更广,可在多场景领域内应用。为此需要重新编写新的 ROS 麦轮底盘节点,更新 Y 方向上的线速度,底盘与麦轮的速度关系可由运动学模型分析得到。

ROS 节点可以简单的理解为一个可执行文件,所有节点受 RosMaster 管理,节点之间通过话题通信、节点内部信息通过服务通信传输信息数据。对于麦轮底盘,其节点主要包含以下功能:① 在回调函数中添加麦轮底盘与轮子之间的速度关系公式,等待速度指令订阅消息、接收 `cmd_vel` 话题数据;② 发布里程计信息,将小车的位姿信息传递给 ROS 管理器。ROS 节点可以通过订阅由导航节点发送的 `cmd_vel` 话题数据,控制小车行驶速度,然后根据底盘轮子直径和转速预估底盘的位置。

图 2 为底盘节点设计流程图,首先初始化底盘节点,创建节点句柄,根据句柄添加实例化对象,根据麦轮运动学模型在回调函数内添加麦轮与底盘之间的速度关系;其次底盘节点发布速度消息、订阅速度话题驱使小车运动;最后初始化里程计信息,根据行走的姿态数据计算里程计信息,在 TF 坐标变换后更新信息并发布给 RosMaster。

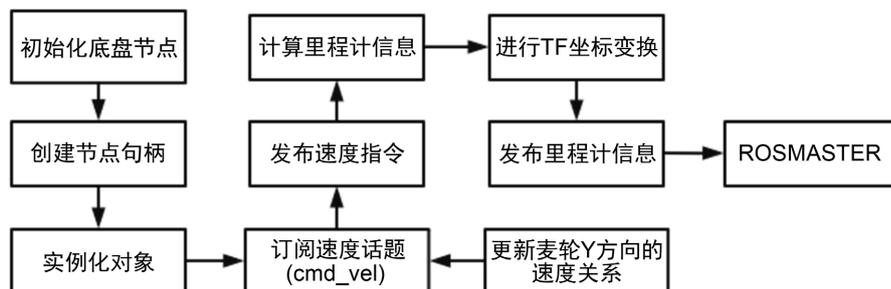


Figure 2. Chassis node design process

图 2. 底盘节点设计流程

### 3. 麦轮底盘运动学模型

和普通四轮小车不同的是，麦克纳姆轮在接收导航节点发送的 `cmd_vel` 速度控制指令时，需要额外增加一个 Y 方向上的速度分量。因此有必要先对麦轮进行运动学分析，以获取各个坐标方向上的分量速度，为机器人的全向移动提供理论依据。图 3 为麦轮安装结构，轮子布局方式采用“O”型安装方式[7]，其中，辊子的角度为  $45^\circ$ ， $\omega$  表示底盘的旋转速度， $v_x$ ， $v_y$  分别为底盘在 X、Y 轴上的线速度； $l_1$ ， $l_2$  分别为轮子中心到底盘重点之间的距离； $\omega_1$ ， $\omega_2$ ， $\omega_3$ ， $\omega_4$  分别表示麦轮的旋转速度。

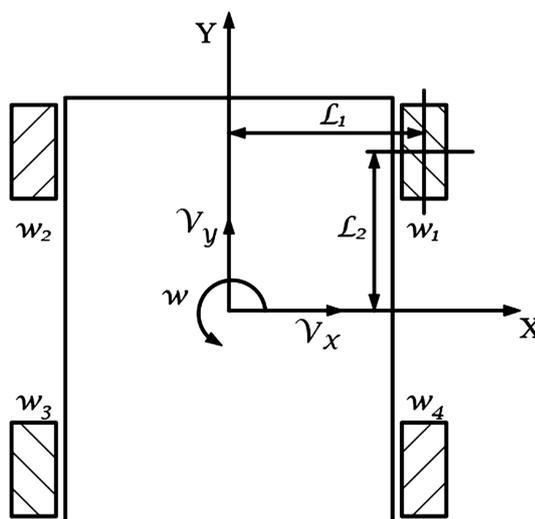


Figure 3. Installation position of Mecanum Wheel

图 3. 麦轮安装示意图

以 1 号轮为例，1 号轮相对于地面的速度为：

$$v_1 = \begin{bmatrix} 0 & \sin \alpha \\ r & \cos \alpha \end{bmatrix} \begin{bmatrix} \omega_1 \\ v_{1r} \end{bmatrix} \quad (\text{式 3-1})$$

1 号轮相对于车体中心的速度为：

$$v_1 = \begin{bmatrix} 1 & 0 & -l_2 \\ 0 & 1 & l_1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (\text{式 3-2})$$

联立(式 3-1)和(式 3-2)，

$$\begin{bmatrix} 0 & \sin \alpha \\ r & \cos \alpha \end{bmatrix} \begin{bmatrix} \omega_1 \\ v_{1r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -l_2 \\ 0 & 1 & l_1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (\text{式 3-3})$$

可得:

$$\omega_1 = \frac{1}{r} \begin{bmatrix} -1 & 1 \\ \tan \alpha & 1 \end{bmatrix} \begin{bmatrix} l_1 + \frac{l_2}{\tan \alpha} \\ v_x \\ v_y \\ \omega \end{bmatrix} \quad (\text{式 3-4})$$

同理, 对其他轮有:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\frac{1}{\tan \alpha} & 1 & l_1 + \frac{l_2}{\tan \alpha} \\ \frac{1}{\tan \alpha} & 1 & -l_1 - \frac{l_2}{\tan \alpha} \\ -\frac{1}{\tan \alpha} & 1 & -l_1 - \frac{l_2}{\tan \alpha} \\ \frac{1}{\tan \alpha} & 1 & l_1 + \frac{l_2}{\tan \alpha} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (\text{式 3-5})$$

由于辊子的偏置角为  $45^\circ$ , 所以化简整理得麦轮的逆运动学模型:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -1 & 1 & l_1 + l_2 \\ 1 & 1 & -l_1 - l_2 \\ -1 & 1 & -l_1 - l_2 \\ 1 & 1 & l_1 + l_2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (\text{式 3-6})$$

通过反解逆运动模型(式 3-6)得到底盘的正运动学模型:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{4}{r} \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ \frac{1}{l_2 + l_1} & -\frac{1}{l_2 + l_1} & \frac{1}{l_2 + l_1} & \frac{1}{l_2 + l_1} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (\text{式 3-7})$$

## 4. 导航系统设计

机器人的导航设计可分为 3 个部分, 地图构建、自身定位和导航规划, 三者之间彼此联系。在开始导航之前, 首先需要通过 SLAM 技术得到全局地图信息; 其次, 基于 IMU 的轮式定位在长时间的运行下会不可避免地产生严重的累计误差, 因此为了提高定位效果, 在导航中融合 AMCL 算法增强定位的鲁棒性; 最后则是利用导航算法实现全局地图的自主行走和避障。对于全局地图规划, 采用 A\*算法寻找最优路径; 利用基于 TEB 算法的局部路径规划器进行实时调整和避障。

### 4.1. 路径规划算法

#### 1) A\*算法

A\*算法综合了 Dijkstra 算法和贪心算法均倾向于取最近节点信息的优点, 因此即可以在两点之间找到最短路径, 同时也能像贪心优先搜索算法一样, 搜索最贴近目标点附近的节点信息, 进而逐渐逼近目

标点。基本原理为：A\*算法最重要的是代价函数  $F(n)$  的设置，通过定义代价函数得到最小的值，进而确定最短的路径。代价函数由两部分组成( $G(n)$ 、 $H(n)$ )，分别表示状态点距离初始状态的实际代价以及距离目标状态的估计代价，通常以  $F(n)=G(n)+H(n)$  表示。

## 2) TEB 算法

TEB 算法(Timed Elastic Band)针对 A\*算法生成的初始轨迹进行后续修正,从而得到平滑优化后路径。具体原理为：在相同的指定时间区间内以固定频率插入多个状态点，将连续时间内规划出的路径近似的认为是一条可以形变的时间弹性带，通过对弹性带实施约束，受力发生形变，这种形变就是它内部的优化算法。通过对其优化，便能找到满足各种约束的最优可行路径。

## 4.2. 地图建模与定位

### 1) Gmapping 地图建模

Gmapping 是基于滤波 SLAM 框架的开源 SLAM 算法，可以实时构建环境地图且构建后的地图精度较高，在 ROS 中通过输入里程计数据和激光雷达数据输出二维栅格地图。图 4 为实现的原理框架：

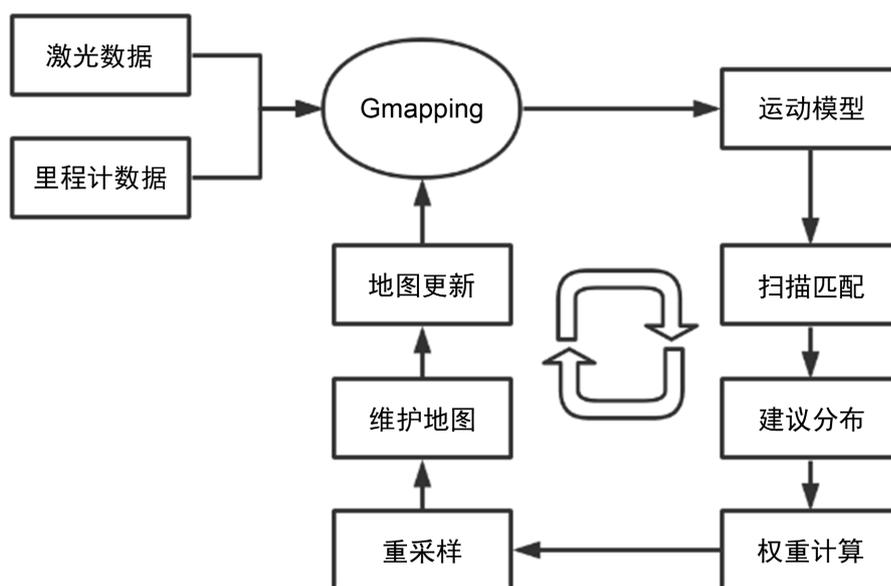


Figure 4. Gmapping diagram building process

图 4. Gmapping 建图流程

Gmapping 通过运动模型给出预测后，以该预测为初值进行一次扫描匹配。找到一个使当前观测最贴合地图的位姿(在高斯分布内表现一个尖峰区域)，之后根据该区域所代表的高斯分布的均值和方差进行计算，得到真实分布的最佳近似。同时，新的粒子继续在高斯分布中重采样得到。

### 2) AMCL 融合定位

由于 IMU 在长时间运行下会发生严重的累计误差，此外，为了减少机器人在运动过程中发生的打滑现象，采用 AMCL (自适应蒙特卡洛定位)融合 IMU 信息对机器人进行增强定位。通过对传感器数据和从地图反馈回来的预估传感器数据相比较，判断出机器人的真实位姿。动态环境下，AMCL 的每个粒子都代表机器人位姿的估计，通过粒子滤波器跟踪机器人位姿，对实际感知状态和粒子群进行价值评估，使其不断趋近于机器人在地图中的真实位置。基于 ROS 实现的 TF 坐标变换对比如图 5 所示：

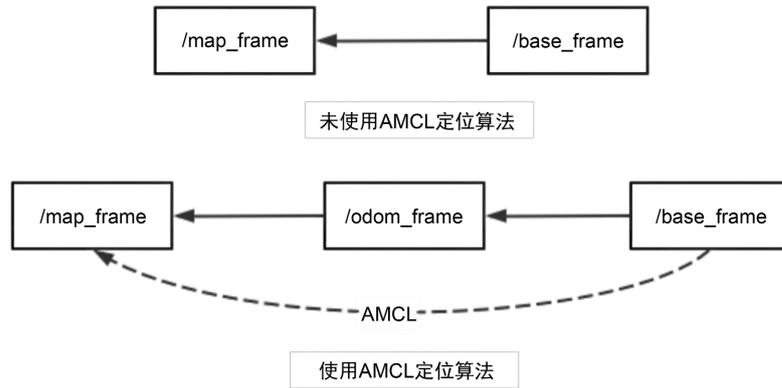


Figure 5. Comparison of TF coordinate trees  
图 5. TF 坐标树对比图

### 4.3. 导航规划

MOVE\_BASE 是一个开源的导航框架，通过编写文件后缀名.yaml 将文件内部信息导入给路径规划层，路径规划层又分为全局路径规划和局部路径规划。在已知环境地图下，全局路径规划为移动机器人计算出一条静态路径，可以快速找到移动机器人位置与目的地之间路径最优解，但是，当已构建环境地图中未知障碍物时，全局路径规划算法比较容易发生导航错误。因此需要结合局部路径规划实现避障。局部路径规划算法在沿着静态路径运动过程中，动态地对机器人进行运动控制，实现在导航过程中机器人的最优运动。MOVE\_BASE 工作流程如图 6 所示：

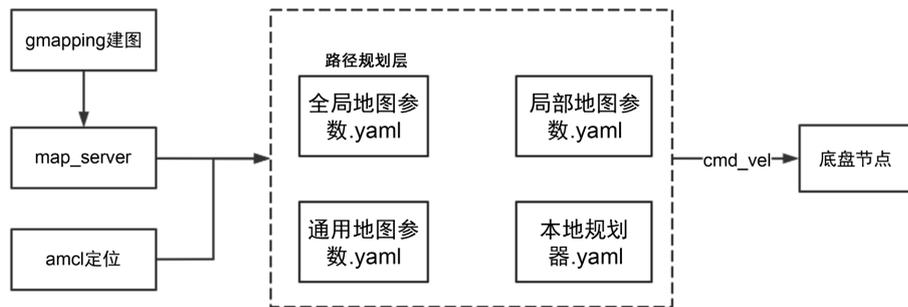


Figure 6. MOVE\_BASE working flow chart  
图 6. MOVE\_BASE 工作流程图

首先通过开源 SLAM 算法 Gmapping 进行地图构建，将所建立好的地图上传到 ROS 中的参数服务器以供 map\_server 节点调用。MOVE\_BASE 的路径规划层根据地图服务信息进行路径规划，结合定位算法向底盘节点发送速度指令(cmd\_vel)驱使电机运动，实现机器人的行走，部分关键参数配置表如表 1 所示：说明如下：

- 1) 全局地图参数：设定地图坐标系/map 和基坐标系/base\_footprint，以供接收地图信息，为了更好地接收地图信息，将发布频率和更新频率设置为 1。
- 2) 局部地图参数：设置了地图坐标系、基坐标系以及局部地图尺寸大小。
- 3) 通用地图参数：设置机器人的外形尺寸，模拟机器人的障碍边界，源传感器的坐标系；为了保护机器人安全，设置冗余尺寸及膨胀半径。
- 4) 本地规划器：设置了机器人的速度和加速度，当机器人卡出时的退出速度，麦轮在 y 方向上的速度等。

**Table 1.** Key parameter configuration table**表 1.** 部分关键参数配置表

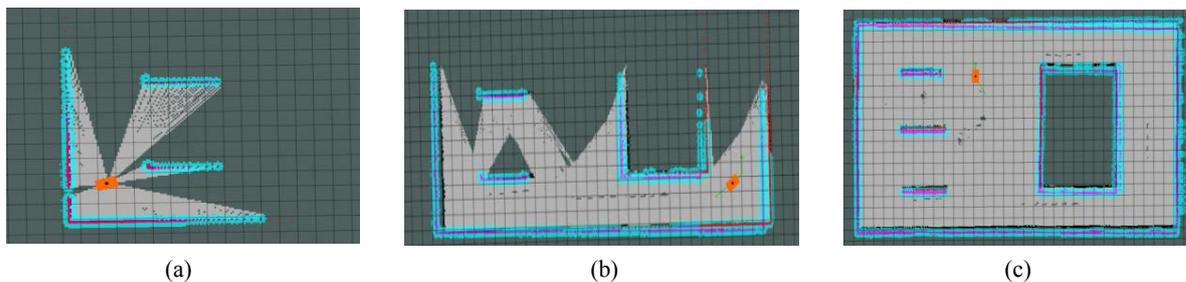
<pre> global_costmap:   global_frame: map   robot_base_frame: base_footprint   update_frequency: 1.0   publish_frequency: 1.0   map_type: costmap </pre>	<pre> local_costmap:   global_frame: odom   robot_base_frame: base_link   width: 5.0   height: 5.0   resolution: 0.01 </pre>
<pre> map_type: costmap obstacle_range: 2.5 raytrace_range: 3.0 footprint: [[0.416,0.318], [0.416, -0.318], [-0.416, -0.318], [-0.416, 0.318]] footprint_inflation: 0.05 observation_sources: scan scan: {data_type: LaserScan, topic: /scan, marking:true,clearing:true, expected_update_rate: 0} </pre>	<pre> TrajectoryPlannerROS:   max_vel_x: 0.45   min_vel_x: 0.1   max_vel_theta: 1.0   min_in_place_vel_theta: 0.4   acc_lim_theta: 3.2   acc_lim_x: 2.5   acc_lim_y: 2.5   holonomic_robot: true   escape_vel: -0.1   y_vels: [-0.3, -0.1, 0.1, -0.3] </pre>

## 5. 实验结果分析

仿真环境为 ROS-Noetic 版本，操作系统为 Ubuntu 20.04，仿真器选用 Gazebo 物理仿真平台。在 GAZEBO 中建立模拟仿真环境，地图环境尺寸为：(长 10 m，宽 8 m)，如图 8(a)。实验目标为地图建模过程中的性能分析，定位效果分析以及自主避障测试；观察所建地图是否有缺失、定位效果是否失效以及是否能够完成自主避障；实验开始前需要配置好机器人的具体属性参数。

### 5.1. 地图建模与定位结果分析

地图建模过程如下：

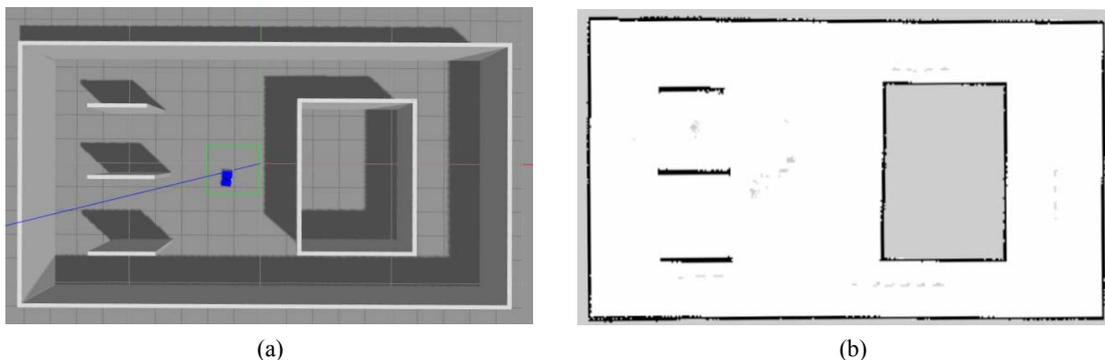


**Figure 7.** Gmapping diagram building process. (a) Start of map construction; (b) Mapping process; (c) End of map establishment

**图 7.** Gmapping 建图过程。(a) 开始建图；(b) 建图中；(c) 建图结束

首先打开 Gazebo 仿真环境，并导入机器人模型，小车初始出现的位置在地图的左下角；其次启动 Gmapping 节点，在 RVIZ 中点击 2D\_GOAL 按钮指定目标点，小车向目标点前进并感知到更多的地图信息，直至完成地图的构建；最后使用 map\_server 地图服务器工具保存创建好的地图。图 7 为地图建模的过程，可以发现，在指定目标点后，小车会自动前往所选目的地，感知周围环境信息及地图建模。

Gazebo 物理仿真环境及建立完成后的地图如下，地图的分辨率大小为 0.02：



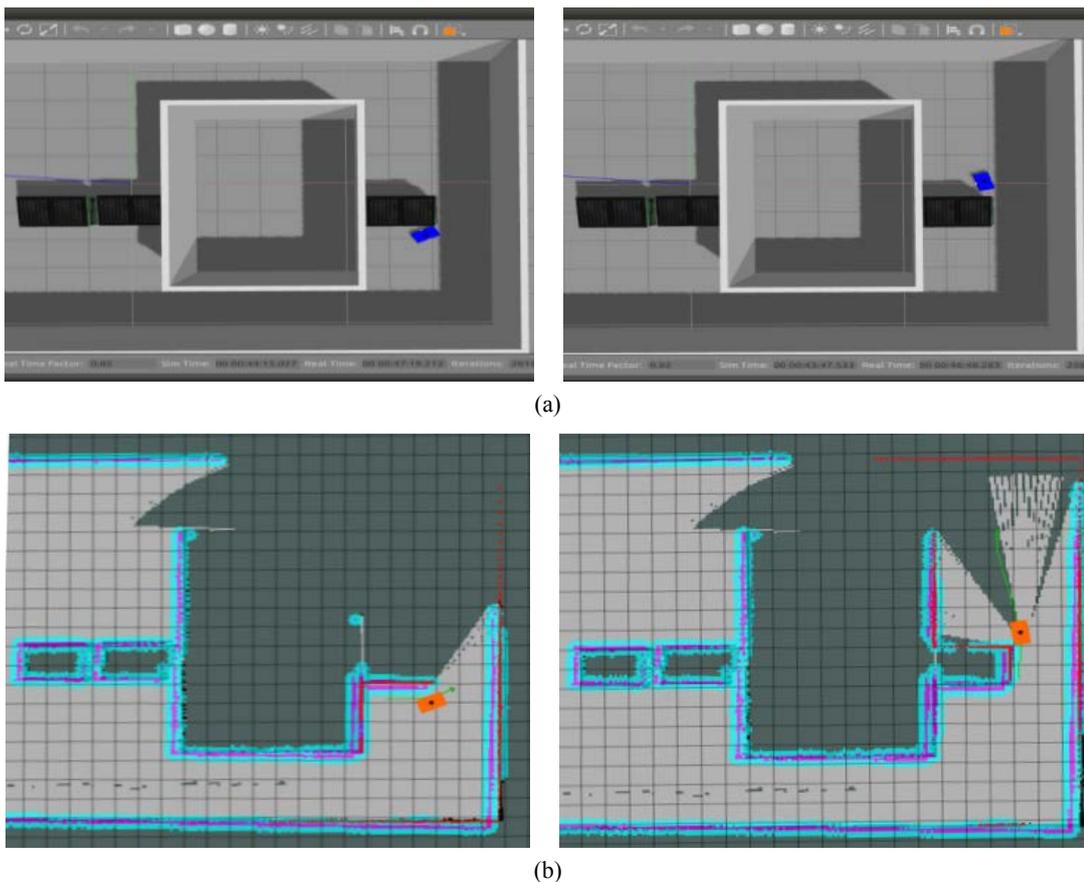
**Figure 8.** Comparison diagram after drawing construction. (a) Physical simulation environment; (b) The completed two-dimensional map

**图 8.** 建图后的对比图。(a) 物理仿真环境；(b) 建立完成后的二维地图

图 8 建图后的对比图，图 8(a)为 gazebo 物理仿真环境，图 8(b)为经 Gmapping 节点启动后建立的地图。可以发现，所得的地图与物理仿真下的环境相差无几。

### 5.2. 导航结果分析

为了测试在动态环境或未知环境下的导航效果，开展机器人的自主移动和避障测试。



**Figure 9.** Experimental comparison of obstacle avoidance. (a) Obstacle avoidance simulation in Gazebo physical simulation environment; (b) Obstacle avoidance simulation of raster map under Rivz

**图 9.** 避障实验对比。(a) Gazebo 物理仿真环境下的避障模拟；(b) Rivz 下的栅格地图避障模拟

图 9 为避障实验对比, 图 9(a)为 Gazebo 物理仿真环境下的避障模拟, 图 9(b)为 Rivz 下的栅格地图避障模拟。首先, 在地图中随机选择目标点, 可以发现, 全局路径规划算法在机器人的基点到目标点之间规划出了一条绿色的路径, 并且机器人逐渐向目标点前进; 当向 GAZEBO 中添加障碍物后, DWA 局部路径规划算法驱使小车移动绕开了障碍物, 并且在 RVIZ 测试平台中逐渐补全地图信息, 所建地图和 GAZEBO 中模拟环境中的地图差别不大, 实现了良好的导航和避障效果。

## 6. 结语及展望

本文首先对麦轮进行了运动学分析, 重新设计了 ROS 底盘节点。结合 ROS 开源平台, 编写了机器人模型, 通过传感器感知里程计信息和深度信息建立了栅格地图, 结合 MOVE\_BASE 导航框架规划路径和动态避障, 设计并实现了移动机器人自主导航功能。试验结果表明, 该机器人可以有效地满足自主导航需求, 能够完成实时避障, 在未知环境下实现自主移动有积极意义, 为之后程序移植到本体样机上提供了技术支持。

## 基金项目

重庆科技学院研究生科技创新计划项目, “基于视觉引导的搬运机器人的路径规划研究”(YKJCX2020313)。

## 参考文献

- [1] 张明岳. 基于 ROS 的室内自主移动与导航机器人研究[J]. 微处理机, 2021, 42(5): 45-48.
- [2] 胡春旭. ROS 机器人开发实践[M]. 北京: 机械工业出版社, 2018: 1-4.
- [3] Koenig, S. and Likhachev, M. (2006) Real-Time Adaptive A\*. *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, Hakodate, 8-12 May, 2006, 281-288. <https://doi.org/10.1145/1160633.1160682>
- [4] 李静, 高俊钗. 基于改进蚁群算法的机器人路径规划[J]. 自动化与仪表, 2020, 35(11): 39-43.
- [5] Gammell, J.D., Srinivasa, S.S. and Barfoot, T.D. (2014) Informed RRT\*: Optimal Sampling-Based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 14-18 September 2014, 2997-3004. <https://doi.org/10.1109/IROS.2014.6942976>
- [6] 仇新, 张禹, 苏晓明. 移动机器人自主定位和导航系统设计与实现[J]. 机床与液压, 2020, 48(10): 88-91.
- [7] 陈博翁, 范传康, 贺骥. 基于麦克纳姆轮的全方位移动平台关键技术研究[J]. 东方电气评论, 2013, 27(4): 7-11.