

# Design of Autonomous Navigation System for Mobile Robot

Jishu Gao Xinmiao Huang Peiru Chen

China University of Mining and Technology, Xuzhou, Jiangsu, 221116, China

## Abstract

In order to realize the environment map construction, location, path planning and navigation of mobile robot in unknown environment, an autonomous navigation system of mobile robot based on robot operating system (ROS) is proposed in this paper. In this design, adaptive Monte Carlo algorithm is used for localization, and map construction based on Cartographer algorithm of 2D Lidar SLAM algorithm, global path planning based on Dijkstra algorithm and local path planning based on TEB algorithm are realized in unknown environment. The experimental platform was built to verify that the robot could complete the tasks of positioning, navigation and obstacle avoidance well, which verified the feasibility of the design of mobile robot autonomous navigation system.

## Keywords

mobile robot; positioning; path planning; map building

# 移动机器人自主导航系统设计

高集殊 黄鑫淼 陈沛儒

中国矿业大学机电工程学院, 中国·江苏 徐州 221116

## 摘要

为实现移动机器人在未知环境下的环境地图构建、定位、路径规划和导航, 论文提出了一种基于机器人操作系统 (ROS) 的移动机器人自主导航系统。该设计采用自适应蒙特卡洛算法进行定位, 实现了在未知环境下基于2D激光雷达SLAM算法中Cartographer算法的地图构建、基于Dijkstra算法的全局路径规划以及基于TEB轨迹规划算法的局部路径规划。搭建实验平台进行验证, 机器人可以很好地完成定位, 导航和避障任务, 验证了移动机器人自主导航系统设计的可行性。

## 关键词

移动机器人; 定位; 路径规划; 地图构建

## 1 引言

伴随着现代高新科技的快速发展, 对移动机器人自主导航技术的研究也不断深入, 其在服务、军事、探测、救援等各个领域的应用也越来越多, 其中室内服务型机器人因其应用广泛而成为当下研究的热点。但室内空间环境往往复杂多变, 对机器人的工作产生了极大挑战<sup>[1]</sup>。为提高移动机器人的操作效率和自主能力, 论文提出一种基于机器人操作系统 (ROS) 的移动机器人自主导航系统, 通过SLAM技术即时构建环境地图, 基于自适应蒙特卡洛算法与多传感信息融合确定机器人位姿, 最终通过Dijkstra和TEB算法实现机器人从起始点到目标点的轨迹规划和实时避障。

**【基金项目】**论文是在国家级大学生创新训练项目的支持下开展。项目为野外移动机器人自主导航系统设计(项目编号: 202010290021Z)。

**【作者简介】**高集殊(1999-), 男, 中国内蒙古赤峰人, 在读本科生, 从事机器人工程研究。

## 2 移动机器人硬件系统

本设计所使用的是XTARK-AKMX2移动机器人, 该移动机器人采用阿卡曼转向模型。后面两个轮为主动轮, 通过左右两个电机差速控制以进行驱动和协调电机。前面两个轮为从动轮, 通过舵机控制移动机器人的转角。该机器人的主要硬件部分图1所示。



图1 XTARK-AKMX2 移动机器人实物图

直流减速编码器电机：采用塔克机器人公司的大扭矩有刷直流减速编码器电机，纯铜线圈，全金属齿轮，高强度磁瓦。额定电压为 12V，减速比 1 : 30，额定功率 5W，编码器分辨率 1440 脉冲 / 圈。

思岚 A1 激光雷达：采用思岚 A1 TK 版激光雷达，DC5V 输入，测量范围 12M、360°，扫描频率 5.5HZ，针对 A1 雷达扫描频率低的缺点，做出改进提升了雷达的扫描频率到 15HZ，建图精度有了明显的提升。

ROS 机器人控制器：机器人处理器主板采用 Jetson Nano，配置有 Linux 操作系统，具体为 Ubuntu 18.04 系统，安装有 ROS 分布式机器人操作系统，ROS 版本为 Melodic 版本，作为机器人端 ROS 节点控制器。

驱动控制器：采用 STM32 作为机器人的底层驱动控制器，通过端口配置对电机实现闭环控制，可与 Jetson Nano 主板实现串口通信。

### 3 移动机器人运动模型

#### 3.1 阿克曼运动模型应用背景

移动机器人不同的运动模型对应了不同的控制方式，其运动的特点和灵活性也有较大差异。常用的移动机器人运动模型有两轮差速模型，全向行进模型，阿克曼模型。本设计所采用的基于阿克曼运动模型的机器人采用舵机通过连杆控制前轮的转角，后面两轮为驱动轮，采用主动差速设计，即根据转向角度通过 PID 控制左右两电机的实际转速差。所以为了使机器人达到准确的转向精度，需要舵机和电机更好的协同配合。

#### 3.2 机器人正向运动学解析

机器人正向运动学解析，就是已知每个轮子的转速，计算得出机器人的位姿。具体做法为通过编码器读取到每个轮子的速度信息，通过驱动控制器的计算获得机器人在 X、Y、Z 三个坐标轴上的运动学参数，也就是我们经常用到的里程计计算。

##### 3.2.1 阿克曼运动模型正向解析理论推导

在使用阿克曼模型时，前两轮在使用舵机控制的时候形成的转角是不同的，使得所有车轮做无滑动的纯滚动时车轮的前进方向垂直于车轮于 ICR 之间的连线。（ICR 为旋转移心）为此我们将运动模型向小车的中轴线上进行简化，设定输入为前轮转角和线速度，如图 2 所示。

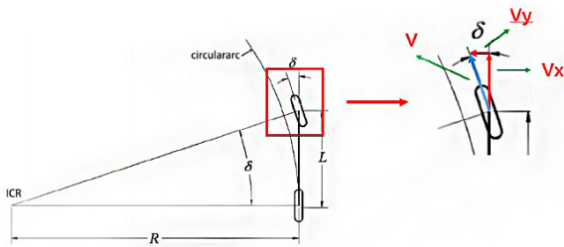


图 2 运动模型简化图

因此，我们得到机器人的运动学方程：

$$\begin{cases} \dot{x} = v \cos \delta \\ \dot{y} = v \sin \delta \\ \dot{\theta} = v \frac{\tan \delta}{L} \end{cases}$$

#### 3.2.2 机器人正向运动学解析代码实现

机器人正向运动学的解析计算主要由 STM32 底层驱动器承担，程序代码主要内容如图 3 所示。

```
//将编码器值转化为前进的距离，单位m
for(int i=0;i<2;i++)
{
    delta_count[i] = (double)(recv_count[i] * wheel_mul[i]*(ENCODER_MAX-ENCODER_MIN)
    - current_count[i])/ticks_per_meter;
    current_count[i] = recv_count[i];
}

//计算底盘x轴变化距离m与Yaw轴转向变化(弧度)
delta_u_ave[0] = (delta_count[0]+delta_count[1])/2.0;
delta_u_ave[1] = delta_u_ave[0]*tan(ANGLE_CEN*((double)(TIHQ>CCR1+(int16_t)(servo_bias*1.111)))
/ ACKMMH_WHEEL_DISTANCE);

//和计算里程计坐标系(odom_frame)下的机器人x,y,Yaw轴坐标
output[0] += (int16_t)((delta_u_ave[0]*cos((double)output[2]/1000)+1000));
output[1] += (int16_t)((delta_u_ave[0]*sin((double)output[2]/1000)+1000));
output[2] += (int16_t)(delta_u_ave[1]*1000);

//Yaw轴坐标变化范围控制-2PI -> 2PI
if(output[2] > PI*1000)
    output[2] -= 2*PI*1000;
else if(output[2] < -PI*1000)
    output[2] += 2*PI*1000;

//发送机器人Yaw轴速度反馈
output[3] = (int16_t)(delta_u_ave[0]*1000);
output[4] = 0;
output[5] = (int16_t)(delta_u_ave[1]*1000);
}
```

图 3 正向运动学解析部分实现代码

该程序主要实现了两个驱动轮的编码器脉冲采集，通过计算输出机器人的里程计信息，包括上述的机器人在各个轴向上的运动学参数。

#### 3.3 机器人逆向运动学解析

机器人逆向运动学的求解，即是期望机器人达到一定的速度和姿态，计算出两个驱动轮的转速以及舵机所应达到的转角。

##### 3.3.1 阿克曼运动模型逆向解析理论推导

逆向解析问题中，期望速度和转角为控制器的输入，后面两驱动电机既要受控达到期望的速度，又要配合舵机达到期望的转角，故须计算出两轮具体的速度，。假设我们的期望速度为，期望转角为。那么经计算左右两轮在满足转角的条件下应满足：

$$\begin{cases} V_l = V \left( 1 - \frac{T \tan \delta}{2L} \right) \\ V_r = V \left( 1 + \frac{T \tan \delta}{2L} \right) \end{cases}$$

##### 3.3.2 机器人逆向运动学解析代码实现

机器人正向运动学的解析计算主要由 STM32 底层驱动器承担，程序代码主要内容如图 4 所示。

```
void AK_Kinematics_Inverse(int16_t* input, int16_t* output)
{
    float x_speed = ((float)input[0])/1000;
    float turn_angular = -((float)input[2])/1000;
    static float wheel_velocity[2] = {0};
    static float servo_angle = 0;

    //计算舵机转角，弧度制转换为角度制
    servo_angle = (turn_angular*180/3.1415);

    //舵机转角限幅
    if(servo_angle>SERVO_MAX)
        servo_angle = SERVO_MAX;
    else if(servo_angle < -SERVO_MAX)
        servo_angle = -SERVO_MAX;

    //线速度限幅
    if(x_speed > SPEED_MAX)
        x_speed = SPEED_MAX;
    else if(x_speed < -SPEED_MAX)
        x_speed = -SPEED_MAX;

    //计算后轮电子速度
    wheel_velocity[0] = x_speed*(1+ACKMMH_WHEEL_TRACK*tan((float)servo_angle/180*3.1415)/2/ACKMMH_WHEEL_DISTANCE);
    wheel_velocity[1] = x_speed*(1-ACKMMH_WHEEL_TRACK*tan((float)servo_angle/180*3.1415)/2/ACKMMH_WHEEL_DISTANCE);

    //转化为实际电机/舵机控制值
    output[0] = (int16_t)(wheel_velocity[0] * ticks_per_meter/PID_RATE);
    output[1] = (int16_t)(wheel_velocity[1] * ticks_per_meter/PID_RATE);
    output[2] = servo_angle*18 - servo_bias;
}
```

图 4 逆向运动学解析部分实现代码

该程序主要实现了控制器输入各轴期望的角速度和线速度，计算舵机和电机的输出状态。

## 4 移动机器人自主导航

### 4.1 定位功能实现

定位功能作为移动机器人自主导航系统设计的基础，对于整个系统运行的精度起着至关重要的作用。本设计使用带有 6D 模型信息的扩展卡尔曼滤波器来整合来自车轮里程计，IMU 传感器和视觉里程计的数据信息。基本思路就是利用松耦合方式融合不同传感器信息实现位姿估计。

定位算法采用自适应蒙特卡洛定位算法 (AMCL, Adaptive Monte Carlo localization)，是蒙特卡洛定位方法的一种升级版，使用自适应的 KLD 方法来更新粒子，而 mcl (蒙特卡洛定位) 算法使用的是粒子滤波的方法来进行定位的。而粒子滤波可以说就是一开始在地图空间很均匀地撒一把粒子，然后通过获取机器人的 motion 来移动粒子，如机器人向前移动了一米，所有的粒子也就向前移动一米。使用每个粒子所处位置模拟一个传感器信息跟观察到的传感器信息作对比，从而赋给每个粒子一个概率。之后根据生成的概率来重新生成粒子，概率越高的生成的概率越大。通过迭代之后，所有的粒子会慢慢地收敛到一起从而推算出机器人的位置，如图 5 所示。

```

1:  Algorithm MCL( $\mathcal{X}_{t-1}, u_t, z_t, m$ ):
2:     $\mathcal{X}_t = \mathcal{X}_{t-1} = \emptyset$ 
3:    for  $m = 1$  to  $M$  do
4:       $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
5:       $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$ 
6:       $\tilde{\mathcal{X}}_t = \tilde{\mathcal{X}}_t + \{x_t^{[m]}, w_t^{[m]}\}$ 
7:    endfor
8:    for  $m = 1$  to  $M$  do
9:      draw  $i$  with probability  $\propto w_t^{[i]}$ 
10:     add  $x_t^{[i]}$  to  $\mathcal{X}_t$ 
11:    endfor
12:    return  $\mathcal{X}_t$ 

```

图 5 AMCL 算法步骤

### 4.2 环境地图构建

常用 2D 激光雷达开源 SLAM 栅格建图算法主要有 HectorSLAM、Gmapping、Cartographer 三种，其中 HectorSLAM 不需要里程计即可建图，但初值的选择对结果影响很大，要求雷达帧率较高，主要适用于空中或者地面不平整的情况，不适用于低成本激光雷达室内建图<sup>[2]</sup>。Gmapping 是基于 RBPf 算法的，将定位与建图分离，按照先定位后建图的原则运行。为了保证精准的地图细节，每个粒子都携带一幅地图，构建大地图需要更多的粒子，导致计算量和内存占用增加，不适合构建大场景、多特征地图。同时由于缺少回环检测，在回环闭合时可能会造成地图错位<sup>[3]</sup>。

Cartographer 是一个基于图优化的 SLAM 算法，采用多传感器融合的方式，如激光雷达、IMU 惯性测量单元等。结合大量的传感器的数据，搭建周围的环境栅格地图，代码

内包含 cartographer 和 cartographer\_ros 两个功能包。每次激光雷达的扫描数据需要融入当前的子图，并进行扫描匹配处理，从而将当前的扫描数据插入到子图中的最优估计位置，然后再用新的扫描数据去更新子图。所以子地图是通过与当前扫描数据通过邻近的多帧的扫描数据更新产生。在单位时间内由扫描匹配得到的位姿估计误差较小，但随着时间的增加会产生累积误差，Cartographer 算法为了优化这个误差会有规律地执行位姿优化。当不再有新的扫描数据映射到子图，就认为这个子图已经创建完成了。当完成所有子图创建，后端处理子图所含的位姿信息，最终实现全局的闭环检测<sup>[4]</sup>。

### 4.3 导航与路径规划

在移动机器人通过激光 SLAM 技术得到了地图信息和空间定位信息后，机器人要想实现智能化的移动和避障，还需要进行路径规划和运动控制。其中，路径规划作为移动机器人技术的热点研究问题与核心难题，对移动机器人的正常稳定运行至关重要。路径规划分为全局和局部路径规划，全局路径规划依靠已知的地图信息为机器人规划出一条从起点到终点的无障碍路径，局部路径规划则依靠传感器获取周围环境的信息，结合机器人自身属性和当前状态，为机器人规划出一条临时的局部路径，帮助机器人躲避动态障碍物<sup>[5]</sup>。

#### 4.3.1 全局路径规划算法研究

目前主流的算法由有 Dijkstra 算法、A\* 算法。Dijkstra 算法采用贪心算法和广度优先搜索的思想，从起点开始向外逐层扩展，直到扩展到终点为止，然后正向遍历比较所有节点得到最短路径。Dijkstra 算法十分简洁，可有效解决单源最短路径问题，但当环境地图较大时，算法所需保存的节点繁多，耗费了大量内存空间与计算时间，严重影响了寻路的速度和效率<sup>[6]</sup>。本设计采用 Dijkstra 算法作为移动机器人的全局路径规划方法。

#### 4.3.2 局部路径规划算法研究

目前，主流的局部规划搜索算法主要有动态窗口法 (DWA) 和 TEB 轨迹规划算法。DWA 是在给定的时间间隔  $\Delta$  内计算出轮式移动机器人的速度 ( $v, w$ ) (线速度和角速度)，这些可行速度对构成速度空间集合 ( $v, w$ )。在得到速度空间的路径规划簇后，可以通过构建路径规划目标评价函数，从这些路径规划簇中选出最优局部路径，即速度 ( $v, w$ )<sup>[7]</sup>。该方法在复杂环境中也具有较好的适应性和较高躲避障碍物效率，但是也具有一定的缺陷：

①由于速度采样的随机性，DWA 算法并不能保证能够得到可行路径 (可能陷入局部最优解)，即使得到可行解也不能保证其最优性。

②由于速度采样的随机性，DWA 算法没有导向目标位置的驱动，因此算法的收敛速度会受到较大影响。

TEB 轨迹规划算法在传统 EB 算法的基础上引入时间间隔信息，以路径偏离惩罚、障碍物靠近惩罚、运动学约束等多个因素进行多目标优化，并在 g2o 框架下求解得到局

部最优路径,帮助移动机器人能够安全快速地躲避障碍物,顺利达到目标位置并完成规定任务<sup>[8]</sup>。TEB 轨迹规划算法是基于多目标优化的图搜索算法,可以有效地避开动态障碍物,而且考虑的约束条件也更全面和实际。因此,本设计采用 TEB 轨迹规划算法作为移动机器人的局部路径规划方法。

## 5 系统构建与实验结果

### 5.1 实验平台与系统架构

论文所使用的实验设备为第 1 节所提到的 XTARK-AKMX2 移动机器人,所搭载的硬件设备见论文第 1 节。在软件平台上,ROS 作为开源机器人软件操作系统平台,集成了大量的库,工具以及协议,能够提供类似操作系统的程序间消息传递、程序发行包管理,底层驱动管理的功能。ROS 提出了一种基于发布/订阅的灵活通讯机制,极大的便利了移动机器人各系统复杂任务下的协调运行。机器人自主导航系统框架<sup>[9]</sup>如图 6 所示

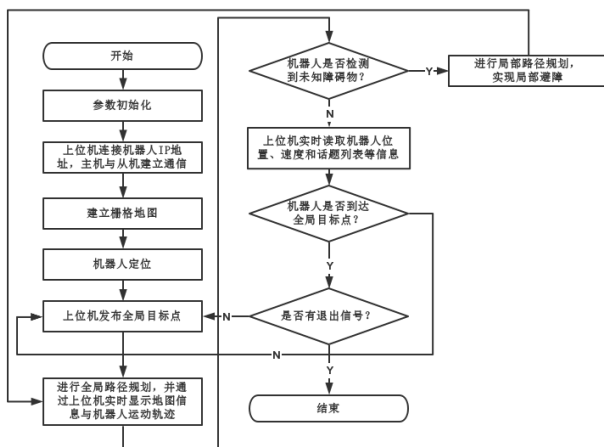


图 6 机器人自主导航系统框架

### 5.2 移动机器人定位与建图

通过个人电脑获取机器人端 Nano 主机的 IP,建立通信获得机器人的控制权。在 ROS 系统中运行机器人驱动节点和手柄控制节点,加载 cartographer 算法并通过手柄控制机器人进行运动建图。通过在 ROS 系统中的可视化界面 Rviz 可实时观测到地图的生成过程,在遍历整个工作空间之后,可以得到完整的二维栅格环境地图,实际环境与建立的栅格地图如图 7、8 所示。



图 7 实际环境



图 8 栅格地图

### 5.3 移动机器人自主导航

完成栅格地图的创建并加载后,基于 AMCL 算法的定位功能节点通过订阅静态地图,里程计数据,激光雷达数据,确定机器人在地图中的位姿。此时我们给出目标点的坐标信息,机器人便可根据当前状态信息规划一条前往目标点的路径。当我们在其运行路径上放置障碍物后,移动机器人将实时更新地图数据,通过局部路径规划绕开障碍物继续向目标点运动。其全局路径规划和局部路径规划如图 9 所示。

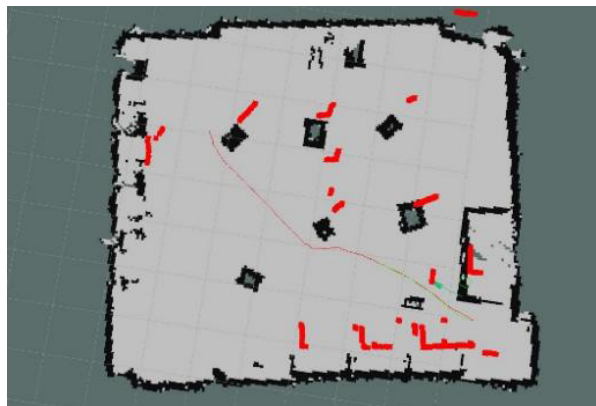


图 9 路径规划的实现

## 6 结语

论文首先对移动机器人运动模型进行了运动学研究,然后对多种 SLAM 算法和路径导航算法的优缺点进行了分析并选择,搭建了一套基于激光 SLAM 的阿克曼移动机器人自主定位与导航系统。实验结果表明:

①多传感器的数据融合并通过 AMCL 算法的加持可以实现较高精度的机器人定位,同时有利于高精度的建图和导航工作。

②基于 Dijkstra 的全局路径规划和基于 TEB 局部路径规划的算法可以在动态变化的环境中实时规划最优路径,抵达目标终点。

### 参考文献

[1] 周旭龙,赵言正,杨玥旻.室内移动机器人自主定位导航系统设计

- [J].中国民航大学学报,2020,38(5):55-59.
- [2] 李杨.基于ROS的室内全向自主导航机器人研究[D].北京:中国矿业大学,2020.
- [3] Hendeby, Gustaf, Karlsson, et al. The Rao-Blackwellized Particle Filter: A Filter Bank Implementation[J]. *Eurasip Journal on Advances in Signal Processing*,2010(1):724087.
- [4] 王忠立,赵杰,蔡鹤皋.大规模环境下基于图优化SLAM的图构建方法[J].哈尔滨工业大学学报,2015,47(1):75-85.
- [5] 李思亮.基于ROS与激光SLAM的自主导航移动机器人研究[D].南京:东南大学,2019.
- [6] Schulz F, Wagner D, Weihe K. Dijkstra's Algorithm On-Line: An Empirical Case Study From Public Railroad Transport[J].*Lecture Notes in Computer Science*, 2000,5(2):12.
- [7] 何壮壮,丁德锐.基于dstar和dwa的改进机器人导航方法[J].电子测量技术,2019,42(12):122-128.
- [8] 赵文瑜.移动机器人SLAM与路径规划研究[D].北京:华北科技学院,2020.
- [9] 崔奇,夏浩,滕游,刘安东.移动机器人自主导航系统及上位机软件设计与实现[J/OL].*计算机测量与控制*:1-7[2021-11-11].<http://kns.cnki.net/kcms/detail/11.4762.TP.20210826.1044.030.html>.