|  |
| --- |
| **智能系统技术实践报告** |
| 基于MATLAB/Simulink的ROS机器人运动控制 |
| |  |  | | --- | --- | | 姓　 名： | 张宇一 | | 学 科 专 业： | 自动化 | | 指 导 教 师： | 王亚午 | | 班 级： |  | | 学 号： |  | |
| 二○二二年十二月 |

# 实验目的及实验步骤

## 实验目的

设计移动机器人路径规划和轨迹跟踪控制算法，使用MATLAB或 Simulink进行代码化或图形化编程，在Gazebo环境下控制移动机器人，并实现其运动控制目标。

## 基于 PRM 算法的路径规划实验步骤

1. 使用rosinit命令创建环境；

2. 载入地图“officemap.mat”，并使用inflate命令对地图进行膨胀操作；

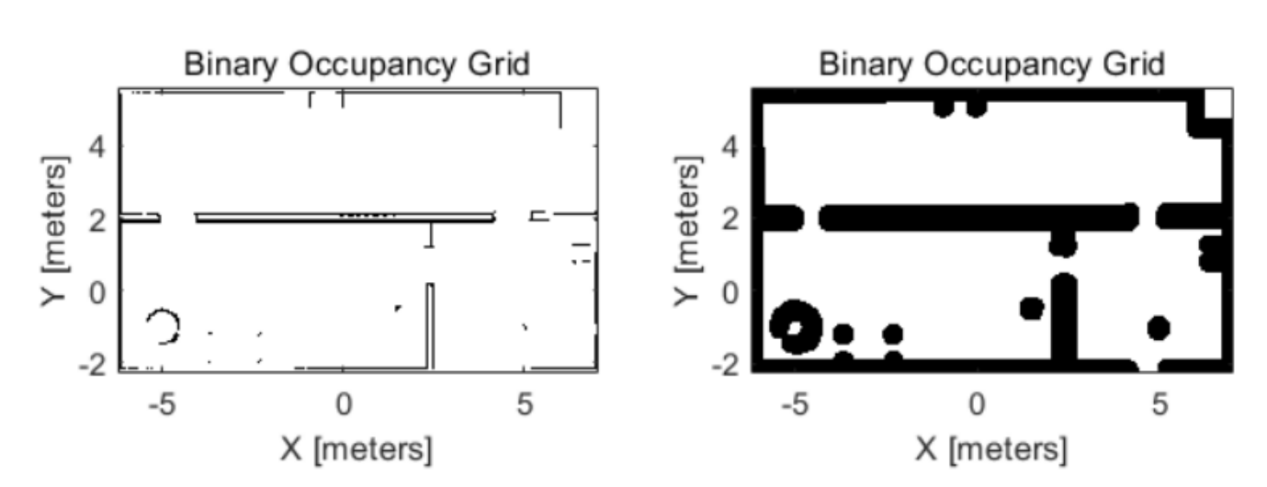


图 1-1 地图展示及膨胀操作

3. 选择路径规划算法 PRM 并设置参数，为移动机器人规划可行动路径；

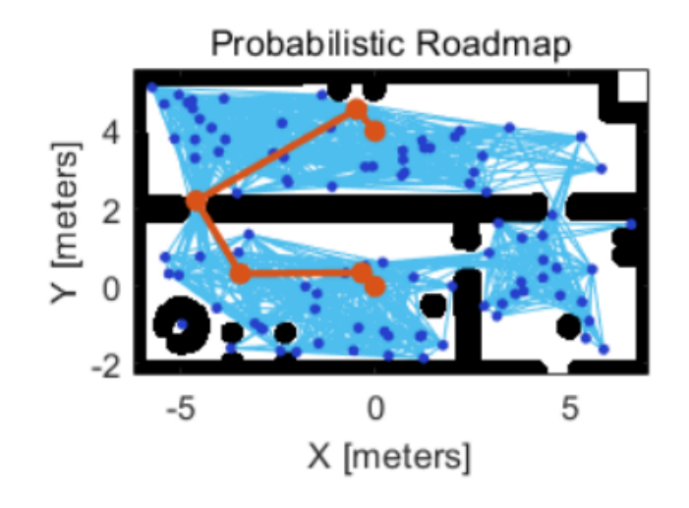


图 1-2 PRM路径规划展示

4. 定义机器人的初始和目标位置、初始朝向，从而获得机器人的当前位姿；

5. 为移动机器人创建 PurePursuit 控制器，用于轨迹跟踪控制，并定义目标

半径和初始偏差；

6. 定义 rospublisher 和 rossubscriber，分别用于向向移动机器人发送速度控

制指令和从机器人端接收位姿信息；

7. 进行运动控制命令，使小车能在 Gazebo 中按照规划路径即进行运动；

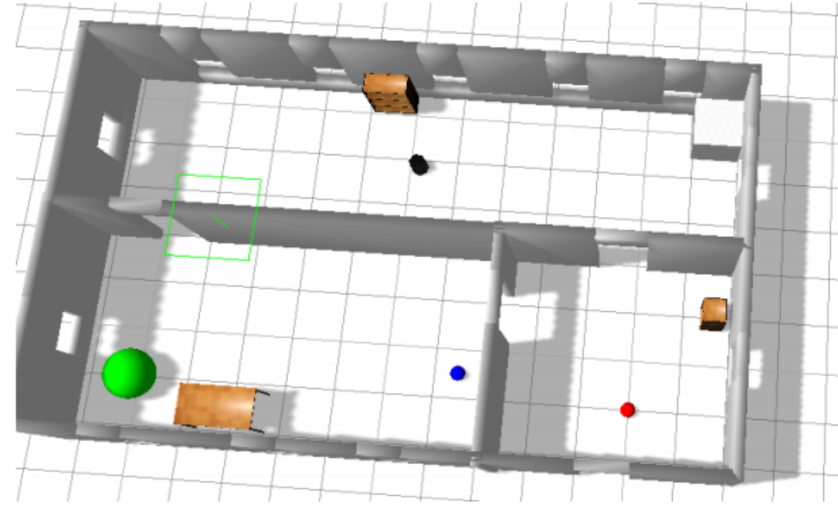


图 1-3 Gazebo环境下的turtlebot运动仿真

8. 使用rosshutdown命令结束 ROS 交互。

完成了PRM算法的实现之后，还会进行其他算法的创新和实践。

# ROS 基本简介

## 2.1 ROS 的基本原理

ROS是一个用于开发机器人应用程序的、类似操作系统的机器人软件平台。

ROS提供开发机器人应用程序时所需的硬件抽象、子设备控制，以及机器人工程中广泛使用的传感、识别、绘图、运动规划等功能。此外 ROS 还提供进程之间的消息解析、功能包管理、库和丰富的开发及调试工具。

使用rosinit命令可以初始化ROS，并创建一个ROS master和一个ROS node。其中master是主节点。主节点负责节点到节点的连接和消息通信，类似于名称服 务器。roscore是它的运行命令，当运行主节点时，可以注册每个节点的名字，并 根据需要获取信息。没有主节点，就不能在节点之间建立访问和消息交流（如话 题和服务）。

当启动 ROS 时，主节点将获取用户设置的 ROS\_MASTER\_URI 变量中列出的 URI 地址和端口。除非另外设置，默认情况下，URI 地址使用当前的本地 IP，端口使用 11311。在Matlab中使用rosinit命令，如下图所示。

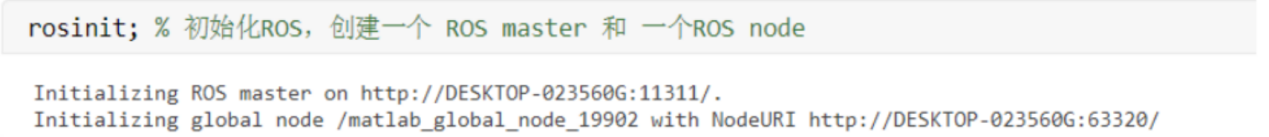


图 2-1 rosinit命令

节点（node）是指在ROS中运行的最小处理器单元。可以把它看作一个可执 行程序。在ROS中，建议为一个目的创建一个节点，建议设计时注重可重用性。 例如，在移动机器人的情况下，为了驱动机器人，将每个程序细分化。也就是说，使用传感器驱动、传感器数据转换、障碍物判断、电机驱动、编码器输入和导航 等多个细分节点。

节点在运行的同时，向主节点注册节点的名称，并且还注册发布（publisher）、 订阅者（subscriber）、服务服务器（service server）、服务客户端（service client） 的名称，且注册消息形式、URI 地址和端口。基于这些信息，每个节点可以使用话题和服务与其他节点交换消息。

rosnode list命令可以用来查看ros网络中所有的node。

节点之间通过消息（message）来发送和接收数据。消息是诸如 integer、floating point和boolean等类型的变量。用户还可以使用诸如消息里包括消息的简单数据结构或列举消息的消息数组的结构。使用消息的通信方法包括TCPROS等，根据情况使用单向消息发送/接收方式的话题（topic）和双向消息 请求（request）/响应 (response）方式的服务（service）。

话题（topic）就是“故事”。在发布者（publisher）节点关于故事向主节点注 册之后，它以消息形式发布关于该故事的广告。希望接收该故事的订阅者节点获得在主节点中以这个话题注册的那个发布者节点的信息。基于这个信息，订阅者节点直接连接到发布者节点，用话题发送和接收消息。

rostopic list 命令可以查看ROS网络中所有话题，rostopic info命令可以查看某个话题的信息，rostopic type 命令可以查看数据类型，而rosmsg show命令可以查看数据的属性。最后使用rosshutdown命令可以关闭整个ROS网络。

## 2.2 ROS 的通讯机制

发布（publish）是指以与话题的内容对应的消息的形式发送数据。为了执行 发布，发布者（publisher）节点在主节点上注册自己的话题等多种信息，并向希 望订阅的订阅者节点发送消息。发布者在节点中声明自己是执行发布的个体。单 个节点可以成为多个发布者。

首先需要创建一个带有topic的rospublisher，并创建一个ROS message。设定msg的参数值，并用send命令发送消息，最后使用rosshutdown命令可以关闭整个 ROS 网络。

订阅是指以与话题内容对应的消息的形式接收数据。为了执行订阅，订阅者节点在主节点上注册自己的话题等多种信息，并从主节点接收那些发布此节点要订阅的话题的发布者节点的信息。基于这个信息，订阅者节点直接联系发布者节点来接收消息。订阅者在节点中声明自己执行订阅的个体。单个节点可以成为多个订阅者。

首先需要创建一个同样带有topic的rossubscriber，使用receive函数接收msg，并用showdetails函数显示message的详细内容，最后使用rosshutdown命令可以关闭整个 ROS 网络。

# 第三章 算法设计及 Turtlebot 控制仿真

## 3.1 移动机器人路径规划算法设计

### 3.1.1 PRM 算法实现

王老师给出的算法采用了PRM算法。PRM（probabilistic roadmap method）是随机路标图的英文缩写，是一种基于图搜索的方法，它将连续空间转换为离散空间，再利用 A\*等搜索算法在路线图上寻找路径，以提高搜索效率。这种方法能用相对少的随机采样点来找到一个解，对多数问题而言，相对少的样本足以覆盖大部分可行的空间，并且找到路径的概率为 1（随着采样数增加，P（找到一条路径）指数的趋向于 1）。

代码实现非常简单，只需要对膨胀操作后的题地图选择mobileRobotPRM 函数，并设置算法参数，如采样点为100，步长限制为5。然后设置好轨迹的起点和终点位置，最后使用findpath函数进行路径规划就实现了。如下图所示为使用PRM 算法进行路径规划的结果。

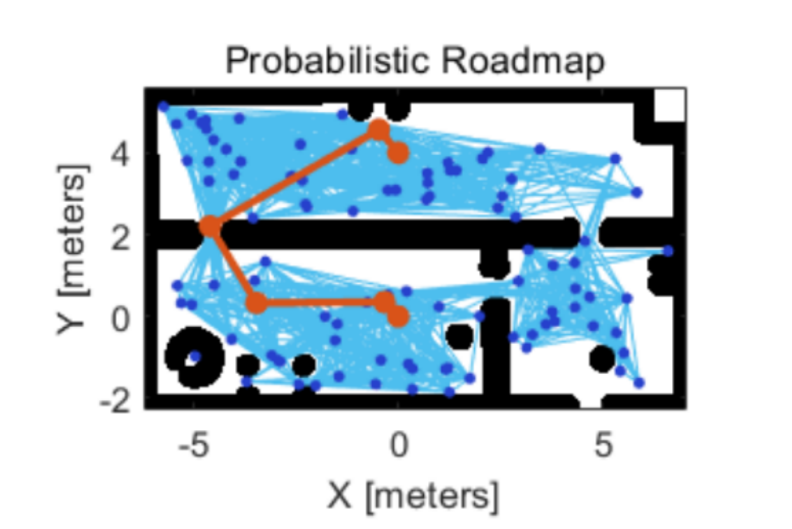


图 3-1 PRM路径规划

经过与虚拟机的连接也实现了 Turtlebot3 机器人在仿真环境中的路径规划实现，较好的完成了移动机器人的路径规划。

### 3.1.2 RRT\*算法实现

RRT\*算法（Rapidly-exploring Random Trees\*）。RRT 是一种多维空间中有效率的规划方法。它以一个初始点作为根节点，通过随机采样增加叶子节点的方式，生成一个随机扩展树，当随机树中的叶子节点包含了目标点或进入了目标区域，便可以在随机树中找到一条由从初始点到目标点的路径。

RRT\*算法的主要特征是能快速的找出初始路径，之后随着采样点的增加，不断地进行优化直到找到目标点或者达到设定的最大循环次数。通过在mathwork 官网上找到的plannerRRTStar文档可知，plannerRRTStar对象创建了一个渐近最优的 RRT 规划器，RRT\*。RRT\*算法旨在收敛到状态空间距离方面的最优解。此外，它的运行时间是 RRT 算法运行时间的常数因素。RRT\*用于解决几何规划问题。几何规划问题要求从状态空间中抽取的任意两个随机状态可以连接。

在实际编程中，规划器需要stateSpace和stateVal两个参数进行建立。前者是状态空间，这里使用stateSpaceSE2二维平移加旋转状态空间作为ss，并用创造的ss建立基于占据地图的状态验证器sv。导入地图map，设置sv参数和ss的状态边界，即可使用plannerRRTStar函数来构建规划器planner。设置规划器参数最大迭代次数2500次，最大连接距离为0.3，并给出起始和终点的位姿，最终使用plan函数进行路径规划，显示规划后的地图并将其进行膨胀处理。具体的图像如下图所示。

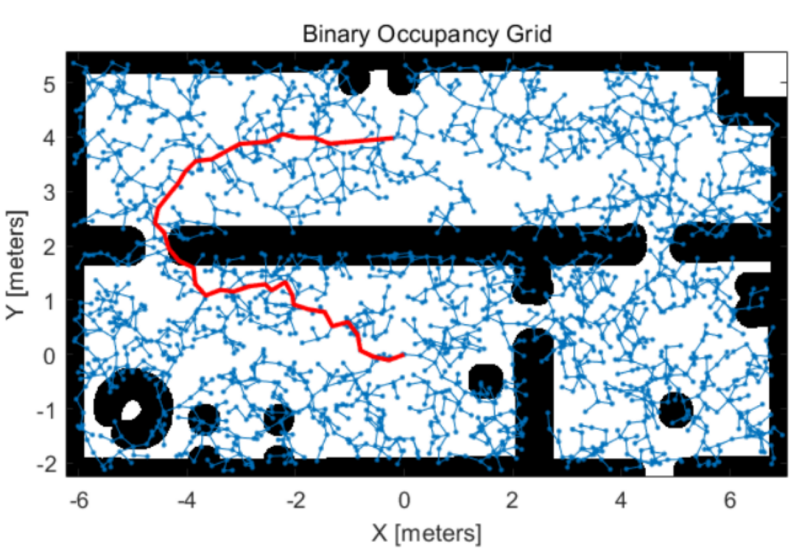


图 3-2 RRT\*路径规划

给出最初的机器人位姿，就可以进入下一步设计路径追踪的算法。

## 3.2 移动机器人轨迹跟踪控制算法设计

对于轨迹跟踪控制算法，采用纯追踪算法。这种算法是一种传统且经典的车辆横向运动控制算法，其基本思想是在每个控制周期，通过前方目标轨迹上的一个点，指导当前方向盘的动作，使车辆产生向目标点的运动。纯追踪算法重要的参数在于前视距离系数，一般而言，其前视距离与速度正相关。

在代码实现层面，我们直接给定控制器为 controllerPurePursuit，并且进一步设置控制器的参数，如期望线性速度为 0.3，最大角速度为 3，期望距离为 0.5 等。接着定义目标半径，并且根据移动机器人当前位姿和目标位姿之差定义初始偏差。

在运动控制中，当移动机器人距终点的偏差大于目标半径时，获取当前的位姿，通过控制器计算速度和转角，并且驱动机器人。当驱动完成后，同样获取当前的位姿，计算更新后的偏差，继续与目标半径进行对比，直到小于目标半径，算法结束。

## 3.3 Gazebo 环境下的移动机器人运动控制仿真

### 3.3.1 Gazebo 仿真器简介

Gazebo是一款3D仿真器，支持机器人开发所需的机器人、传感器和环境模型，并且通过搭载的物理引擎可以得到逼真的仿真结果。Gazebo是近年来最受欢迎的三维仿真器之一，并被选为美国 DARPA 机器人挑战赛的官方仿真器。

Gazebo同样是开源的仿真器，也有着高水准的仿真性能，因此在机器人工程领域中非常流行。不仅如此，负责开发和普及ROS，且担任社区的Open Robotics在开发ROS和Gazebo，因此ROS和Gazebo非常兼容。在linux系统中，如果已经安装 Gazebo，运行“gazebo”命令即可运行 Gazebo 进行相关的仿真工作。在本文中我们所使用的 Gazebo 版本为 8.0。如下所示为进入 Gazebo 的初始画面。

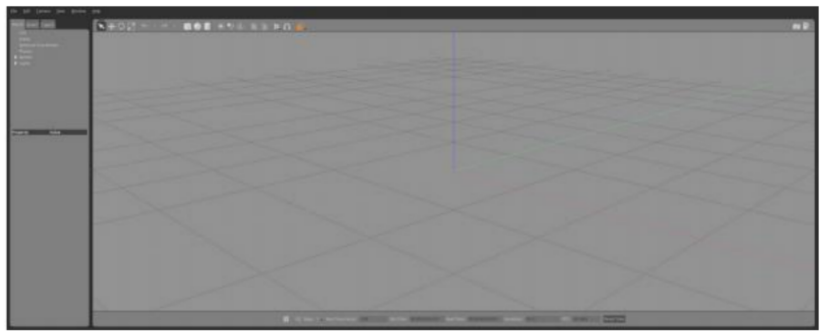


图 3-3 Gazebo初始画面

### 3.3.2 Gazebo 环境下的移动机器人运动控制仿真实现

根据上述步骤，首先在Matlab中创建ROS环境，然后将路径规划算法、轨迹跟踪控制算法的代码进行复用，最后创建发布者和订阅者，进行对移动机器人的运动控制。以上代码均在matlab中实现。打开虚拟机 ROS Indigo Gazebo v3。首先打开终端，输入命令roscore。roscore命令会运行主节点，主节点管理节点之间的消息通信中的连接信息。主节点是使用ROS时必须首先被运行的必要元素。然后打开Gazebo Turtlebot World，加载出地图界面与其中的移动机器人模型。接着可以打开Matlab，运行之前处理好的mlx文件，稍作等待，即可看到Gazebo中移动机器人从初始点到目标点的运动过程。

移动机器人最终达到目标点，如下图所示。

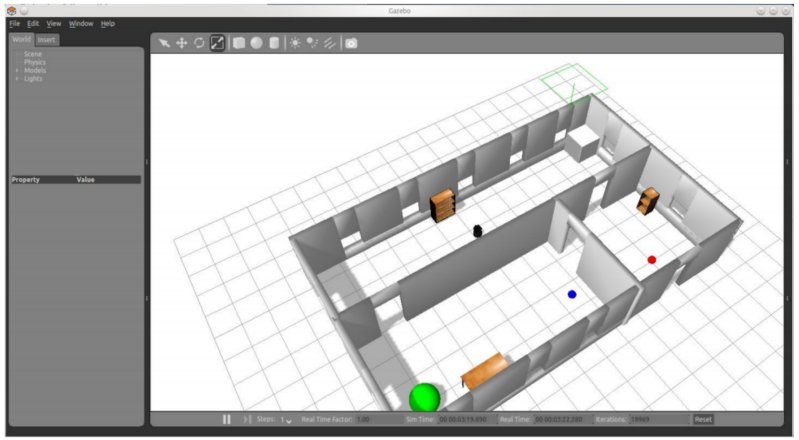


图 3-4 Gazebo环境下的Turtlebot3仿真

# 第四章 总结

本次实习主要是针对Turtlebot3机器人进行的MATLAB/Simulink的ROS机器人运动控制。从中我学习了关于ROS系统的基本情况、通信原理等相关ROS入门的基础知识。同时了解到了用MATLAB进行ROS机器人运动控制开发的简便性，了解了关于MATLAB中相关于机器人开发的工具包和基础使用。并且通过查询相关资料和MATLAB的官方文档，完成了对于路径规划算法和跟踪算法的新算法的实践。