基于ROS的移动机器人平台系统设计

智能移动机器人对我国先进制造技术的发展起着关键作用，利用机器人技术和信息技术的相互结合，移动机器人将成为未来数字化信息化社会的关键角色。随着移动机器人的发展，传统按键、摇杆等远程操作方式由于其操作繁琐等因素渐渐淡出人们的视野，基于视觉跟随、语音控制、自动导航等新型控制方式的机器人由于具有增强人机交互、提高参与感等优点越来越受到人们的关注。然而，随着机器人技术的发展，为不同机器人编写软件这一工作也变得越来越繁重。不同的机器人设计方案不同，底层接口也大不相同，编写软件也不同。虽然不同型号的机器人其基本算法是一致的，但是由于硬件板卡的不一致，导致了大量代码冗余、通用性较差等缺点。为了改善这些缺点，机器人操作系统RO S应运而生。RO S很好地解决了软件开发过程中代码冗余、移植性差等问题。本文在RO S平台的基础上，搭建移动机器人平台。该移动机器人平台以模拟人类自我学习的过程为目的，理解自己的状态和外部环境信息，从而实现实时运动控制决策、避障、找到最优路径；在栅格地图表征环境，采用栅格法进行路径规划自主移动和轨迹跟踪。

1 系统框架

本课题基于开源的 ROS 机器人操作系统框架，提出一种层次化的平台架构如图3-4所示。架构自顶向下分为应用层、分布式系统框架层、硬件层三个层次。该架构层次分明，结构清晰，采 用了模块化设计，具有良好的适应性和可拓展性，同时支持多机器人的组网控制和远 程控制。只要遵循相关的开发协议和规范，任何机器人、智能设备或者应用都可以通 过统一的接口接入该平台。



图3-4 平台架构

### 2 硬件层设计

硬件层是机器人全部实体的组成部分，如图3-5所示根据硬件部分的功能分类，硬件层又可以进一步划分为传感器系统、执行系统、信息处理系统。

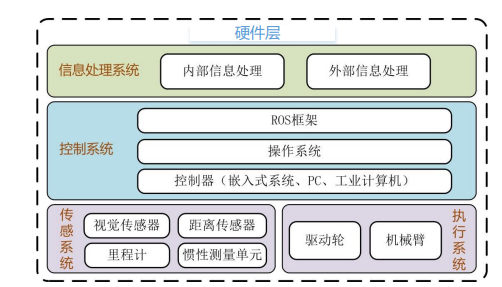


图3-5 硬件层组成

2.1 感知器系统

感知器系统是机器人对外界环境进行感知的系统。类似于人类利用多种感官对环境的进行感知方式，机器人也可以利用多种类型的传感器来对获取环境信息。机器人上常用的传感器包括视觉传感器、距离传感器、里程计、陀螺仪、加速度计等等。视觉传感器可以获得图像信息，机器人通过视觉传感器可以识别人脸，识别特定物体，识别运动物体，也可以将周围环境情况实时发送到远程端，实现远程监控的功能。深度传感器和距离传感器主要用来进行环境感知，计算环境中障碍物距离机器人的距离，在机器人地图构建和智能导航阶段都起着至关重要的作用。陀螺仪、加速度计等用来感知机器人自身的位姿和状态，提高对复杂地形的适应能力。

2.2 执行系统

执行系统是指机器人身上的机械执行机构。一般机器人上最为常见的执行机构就是驱动轮，机器关节等。一个完整地执行机构又可以细分为驱动装置、检测装置和控制系统这三部分。驱动装置是驱使执行机构运动的装置，一般底层的执行部件都是电机或者舵机；通过外围的检测装置和控制系统，接受来自上层的命令并实现精确的运动控制和动作执行。如图3-6。



图3-6 机器人执行系统

2.3 信息处理系统

信息处理系统相当于机器人的大脑，一般由高性能的pc主机或者嵌入式主机构成，当然也可以通过网络连接远程主机作为机器人的中央处理系统，相当于机器人的远程大脑。传感系统和执行系统都要通过通讯接口与信息处理系统相连。信息处理系统会根据当前应用的需要，获取多路传感器的数据、处理，并向执行机构发出执行指令，控制机器人完成指定的任务。同时，信息处理系统还应具有互联网接入、使用云端资源的能力，可以为机器人提供远程升级、远程监控、机器学习等智能化服务，机器人不再只是单一的个体，互联网和云计算使得每台机器人都可以拥有超级计算机级别的大脑。

### 3 分布式系统框架层设计

分布式系统框架层处于整个系统的中间位置，向上为应用层提供数据共享和传输服务，向下负责机器人硬件层的抽象化处理、数据封装等功能。如图3-7所示分布式系统框架层基于ROS，其分布式系统的特点，使得不同设备在同一局域网内就可以把整个系统看做一个整体，在系统层级可以不区分设备，相当于在同一设备上可以调用任意系统资源。PC、移动端设备和其它机器人都可以以节点的形式接入整个系统，通过向主节点订阅和发布消息，系统中的任意节点之间都可以实现数据的共享和指令的发送。



图3-7 分布式系统框架

3.1 底层驱动部分。

这部分主要由机器人控制节点对底层硬件的抽象，涉及机器人的移动控制和传感器原始数据的采集，底层硬件框架如图3-8所示，



图3-8 底层硬件框架

主要包括核心处理器、嵌入式控制板、电机驱动、编码器、惯性测量单元(IMU)和机器人控制节点这几个部分。其中，惯性测量单元负责机器人初始化航向角的校正，提供实时的姿态角数据，电机驱动和编码器部分则负责机器人移动，这些均由一个嵌入式控制板进行控制和数据采集。这些数据包括左右驱动轮编码器数据和IMU数据，再由嵌入式控制板以串口的方式传输给机载计算机，由机器人控制节点进行处理并将数据以“话题”的方式发布。

3.2 人机交互部分。

人机交互的设计目标主要是方便用户操作，是对底层控制命令的封装，用户可直接输入期望轨迹，它是由一个个坐标点组成，由轨迹跟踪控制节点计算并转化成机器人所需的期望线速度u和期望角速度以将新的期望速度通过话题发布，再由机器人控制节点订阅并更新将其转化成自定义命令发送给嵌入式控制板。同时，除了命令行的方式外，只要将其转化成机器人的移动命令，还可以通过一些控制设备进行控制，例如利用手柄来控制机器人的移动。

3.3 数据可视化。

数据可视化可带来更直观的感受，方便开发人员调试或者展示效果。ROS系统提供了数据的可视化工具Rviz，可以帮助机器人记录和现实运行轨迹以及位姿状态。这部分主要由里程计计算节点节点和Rviz节点组成。里程计计算节点订阅由机器人控制节点发布的“话题”，接收左右驱动轮实时速度信息，进行计算并发布里程计话题，再由Rviz节点订阅该话题就可以实时更新显示机器人的运动轨迹以及位姿状态。

4 基于 ＲOS 的自主导航实现

ＲOS 下的经典导航框架如图 8 所示［4］。首 先需要激光雷达建立实验环境地图，采用较为常 用的 2D SLAM 算法 Gmapping ［5］。导航过程中， 自适应蒙特卡洛定位算法( adaptive Monte Carlo localization，AMCL) 实现机器人在地图坐标系的 定位，ＲOS 里的 laser\_scan\_matcher 功能包可以通 过比对激光雷达的前后数据帧来得到激光雷达位 姿变化的估计，进而得到机器人的里程信息( 激光 雷达固定在机器人的前面，二者之间是简单的平移 关系) 。这样就得到了地图坐标系( map) 、里程计 坐标系( odom) 和机器人坐标系( base\_link) 三者 之间的坐标转换关系，通过 tf( 坐标变换树) 统一 发布相应的话题( topic) 。

