doi:10.3772/j.issn.1002-0470.2021.07.007

基于 ROS 和全向舵轮驱动的移动机器人系统设计①

徐建明② 吴小文 蔡奇正 倪洪杰

(浙江工业大学信息工程学院 杭州 310023)

摘 要 本文设计了一种基于机器人操作系统(ROS)和 EtherCAT 的双舵轮全向移动机器人控制系统。首先建立双舵轮对角驱动移动机器人运动学模型;采用 Xenomai 实时内核和 IgHEtherCAT 主站技术设计机器人底层驱动;在 ROS 平台下,以 move_base 为核心设计机器人导航规划层,导航数据源由惯性测量单元、编码器和激光雷达获取;利用 Qt 设计远程客户端人机界面,主要包含控制窗口和数据库等界面;最后,针对构建的全向移动机器人进行自主导航实验,实验结果验证了所设计系统的实用性。

关键词 机器人操作系统(ROS);全向移动机器人;激光导航;EtherCAT;舵轮

0 引 言

随着智能制造业的推进及工业物流行业的扩张,物流机器人技术得到不断的发展。移动机器人作为工业物流机器人的一员,其系统正朝着控制智能化、多机协同化、设计模块化方向发展,人们对其功能需求也越来越多,因此加快物流机器人技术的发展对推动智能装备制造产业具有重要意义[1-2]。

当前国内外工业移动机器人主要面向搬运、配送等任务,此类机器人主要由传统的大型装备制造商生产,并通过专用软件和编程语言实现控制,存在控制系统开放程度低、不易扩展的缺点[3]。同时也有基于商用平台开发移动机器人控制系统,如文献[4]基于 CoDeSys 开发了两轮移动机器人系统,但这种商用平台往往需要收取大量授权费用,且功能块在不同平台之间难以移植。移动机器人在消防、安保、巡检等领域的应用[5-8],使得移动机器人应用场景越来越广、功能需求越来越多,研发一个开放通用、易扩展的移动机器人控制系统具有重要的应用价值。

移动机器人控制系统的设计涉及的技术众多,包括底层驱动、自主导航以及避障等机器人应用技术,同时还需要根据实际需求设计人机交互等功能模块。如何把功能模块低耦合、高内聚[9]地整合到系统中是机器人系统研究领域的一大难点。近年来,具有高度开放性的开源机器人操作系统(robot operating system,ROS)愈发受到研发人员的青睐,其分布式、松耦合的控制节点(Nodes)框架[10],使得机器人模块设计不再依赖于特定编程语言,并且提高了代码复用率和开发效率[11]。如文献[12]在 JetsonTK1 上基于 ROS 设计三轮全向移动机器人系统,采用模块化的设计方式整合移动机器人基本功能模块。文献[13]结合 ARM 开发板设计机器人控制系统。文献[14]则在 ROS 系统的基础上结合 Windows 系统设计导游机器人功能模块。但由于 ROS 系统为非实时操作系统[15],因此它的应用多集中在服务机器人上,难以在控制精度要求较高的工业机器人系统中发挥作用。近年来,开源实时内核如 PREEMPT_RT、Xenomai、RTAI 等的推出,其线程的优先性极大地提高了操作系统的实时性[16]。EtherCAT 是德国 Beckhoff 研发的高性能工业以太

① 国家自然科学基金-浙江省自然科学基金联合基金两化融合项目(U1709213)和国家自然科学基金面上项目(61374103)资助。

② 男,1970 年生,博士,教授;研究方向:迭代学习控制,电机伺服控制技术,机器人控制技术;联系人,E-mail:xujm@zjut.edu.cn

(收稿日期:2020-07-16)

网技术,广泛运用在工业控制系统中,其与实时内核的结合应用[17]能有效地解决 ROS 系统在工业移动机器人控制系统的实时性问题。

本文以双舵轮移动机器人底盘为研究对象,以 ROS 作为开发平台设计了一种双舵轮全向自主导航移动机器人。首先对机器人底盘进行运动学分析;利用 Xenomai 实时内核和 IgHEtherCAT 主站技术设计驱动层控制节点;基于 ROSmove _ base 设计机器人导航规划层,实现移动机器人自主导航和避障功能;通过 Qt 设计远程客户端人机界面,包含控制窗口、数据库等模块;最后通过机器人的自主导航实验验证所设计系统的有效性。

1 控制系统架构设计

1.1 控制系统硬件架构

全向移动机器人系统总体结构如图 1 所示。底盘采用双舵轮对角驱动方式,舵轮由牵引电机和转向电机组成,由 RoboteQ 双通道驱动器驱动。传感器采用 Sick-lms111 激光雷达、Razor-IMU 惯性测量单元和增量式编码器。控制系统主控单元由搭载 Intel 赛扬处理器的工控机构成,用于接收调度信息、处理传感器数据以及底盘的运动规划。远程客户端通过局域网访问底盘控制系统,并根据任务要求下发目标位姿,控制系统根据目标实现机器人自主导航。

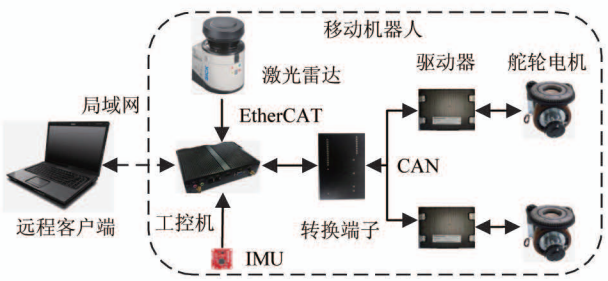


图 1 控制系统硬件架构图

1.2 控制系统软件架构

基于模块化设计思想,分别设计远程客户端和机器人底盘控制模块,并通过数据处理节点实现数据交互。系统软件采用分布式控制节点的方式,分布在底盘控制系统和远程客户端中。按层次可划分为交互层、导航规划层以及驱动层,结构如图 2 所示。交互层包含数据处理节点及基于 Qt 框架设计的人机界面和数据库,通过 ROS 消息机制关联不同系统机器人模块接口,提供人机交互功能。导航规划层基于 ROSmove _ base 设计,提供地图构建、即时定位、路径规划、速度规划以及避障等功能。驱动层结合 EtherCAT 主站技术设计实现并封装在 ROS 节点中,工控机通过 EtherCAT 转 CAN 端子实现与驱动器通信,EtherCAT 主站用户层运行在 Xenomai 实时内核中并利用线程共享内存方式与 ROS 节点进行数据交互,其运行周期可达 1 ms。

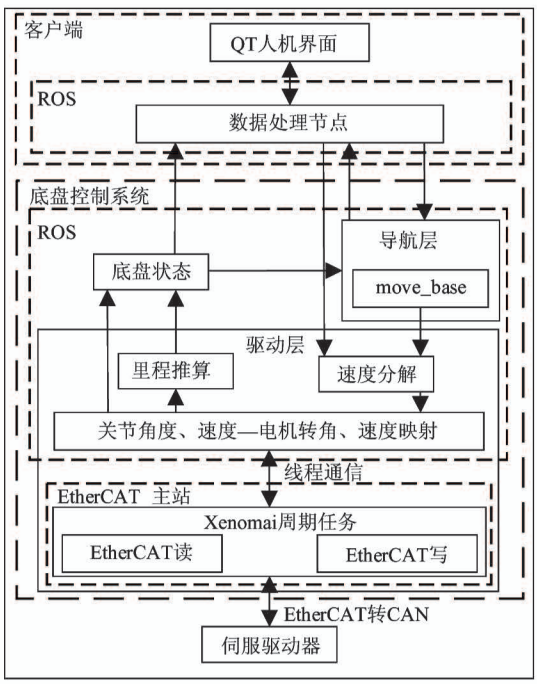


图 2 控制系统软件架构图

系统基本控制流程如下:用户通过远程客户端 QT 人机界面发出控制指令或者导航目标,经数据处理节点解析后,点动控制指令直接将底盘速度发送至驱动层,导航指令则发送目标位姿至导航层,经过 move _ base 路径规划、速度规划后将底盘速度下发至驱动层。驱动层将获取的速度进行分解得到舵轮电机的速度和舵角,通过线程通信由 EtherCAT 主站将控制指令经总线发送至电机驱动器。

2 移动机器人运动学建模

移动机器人底盘结构形式如图 3 所示。

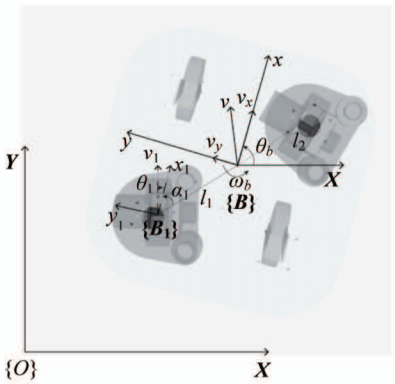


图 3　移动底盘结构图

两个驱动舵轮位于底盘对角放置，配有两个对角支撑随动轮。相较于常见的两轮差速或 Ackermann[18] 移动底盘，舵轮对角放置的驱动方式可实现机器人的原地旋转和横向平移等动作，具有高度的灵活性。

以底盘舵轮中心连线的中点，即底盘的几何中心作为车体基座标参考点，设机器人坐标系 $\{B\}$、舵轮坐标系 $\{B_i\}$。机器人相对世界坐标系 $\{O\}$ 姿态 θ_b 及其角速度 ω_b、线速度 v、线速度分量 v_x、v_y，v_i、θ_i 为舵轮的速度和舵角。已知量 l_i、$\alpha_i(i=1,2)$ 分别表示舵轮中心到底盘中心的距离和夹角。根据底盘在 $\{B\}$ 坐标系中的牵引速度 v 和转角速度 ω_b 推导舵轮速度 v_i 和舵角 θ_i，由几何关系可以推出机器人的逆运动学方程。

$$v_i\begin{bmatrix}\cos\theta_i\\ \sin\theta_i\end{bmatrix}=\begin{bmatrix}v_x\\ v_y\end{bmatrix}+\omega_b l_i\begin{bmatrix}\sin\alpha_i\\ \cos\alpha_i\end{bmatrix} \tag{1}$$

给出速度控制量 v_x、v_y 和 ω_b，通过逆运动学即可算出 v_i 和 θ_i。根据舵轮编码器反馈的实际电机牵引速度 v_i 和转角 θ_i，由运动约束可得正运动学关系：

$$\begin{bmatrix}1 & 0 & l_1\sin\alpha_1\\ 0 & 1 & l_1\cos\alpha_1\\ 1 & 0 & -l_2\sin\alpha_2\\ 0 & 1 & -l_2\cos\alpha_2\end{bmatrix}\begin{bmatrix}v_x\\ v_y\\ \omega_b\end{bmatrix}=\begin{bmatrix}v_1\cos\theta_1\\ v_1\sin\theta_1\\ v_2\cos\theta_2\\ v_2\sin\theta_2\end{bmatrix} \tag{2}$$

令式(2)为 $\boldsymbol{B}[v_x v_y \omega_b]^{\mathrm{T}}=\boldsymbol{A}$，关系式有 4 个方程和 3 个未知数，根据最小二乘法可得：

$$\begin{bmatrix}v_x\\ v_y\\ \omega_b\end{bmatrix}=(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{A} \tag{3}$$

由姿态角 θ_b 得到底盘速度相对世界坐标系 $\{O\}$ 的转换矩阵。

$${}^{B}_{O}\boldsymbol{R}(\theta_b)=\begin{bmatrix}\cos\theta_b & -\sin\theta_b & 0\\ \sin\theta_b & \cos\theta_b & 0\\ 0 & 0 & 1\end{bmatrix} \tag{4}$$

则移动底盘在世界坐标系下的相对速度为

$$\begin{bmatrix}v_{ox}\\ v_{oy}\\ \omega_b\end{bmatrix}={}^{B}_{O}\boldsymbol{R}(\theta_b)=\begin{bmatrix}v_x\\ v_y\\ \omega_b\end{bmatrix} \tag{5}$$

假设轮子不打滑，T_{sample} 为系统采样周期，则 k 时刻机器人相对世界坐标系的实际位姿为

$$\begin{cases}x(k)=x(k-1)+v_{ox}\times T_{sample}\\ y(k)=y(k-1)+v_{ox}\times T_{sample}\\ \theta_b(k)=\theta_b(k-1)+\omega_b\times T_{sample}\end{cases} \tag{6}$$

3　移动底盘控制系统设计

导航规划层和驱动层分布在底盘控制系统中，系统功能以 ROS 节点的形式展现，并通过 topic 或 action 的消息机制实现节点间通信，节点关系如图 4 所示，功能如表 1 所示。

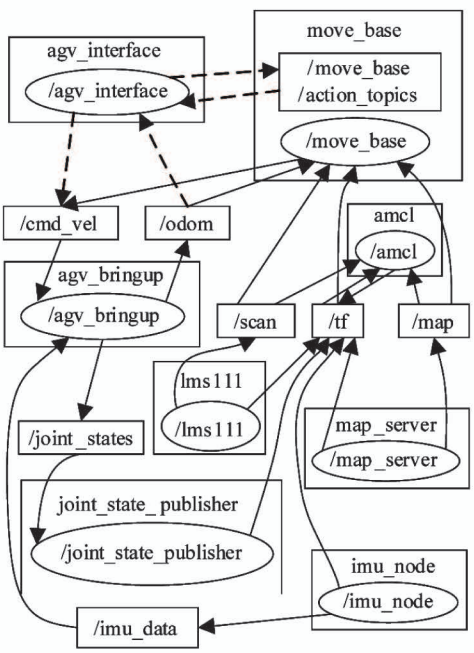


图 4　节点关系图

表 1　节点功能表

节点名称	节点功能
agv _ interface	数据处理节点,分布在远程客户端,实现人机界面与底盘控制系统的数据交互,接收里程消息和导航状态信息,发布目标位姿和速度消息
move _ base	实现底盘自主导航的核心,接收底盘位姿消息/odom、雷达数据消息/scan 和坐标间转换关系/tf,并根据目标规划路径和速度
amcl	接收雷达数据消息/scan,实现底盘定位,并发布里程坐标系和世界坐标系的/tf 坐标转换关系
map _ server	加载静态地图,发布地图消息/map
lms111	启动激光雷达并发布雷达数据消息/scan
imu _ node	启动惯性测量单元并发布单元的姿态和速度消息/imu _ data
state _ publisher	根据关节状态映射发布底盘舵轮关节速度、位置状态/joint _ states
agv _ bringup	接收/cmd _ vel 速度消息进行速度分解、控制底盘运动,并反馈里程消息/odom

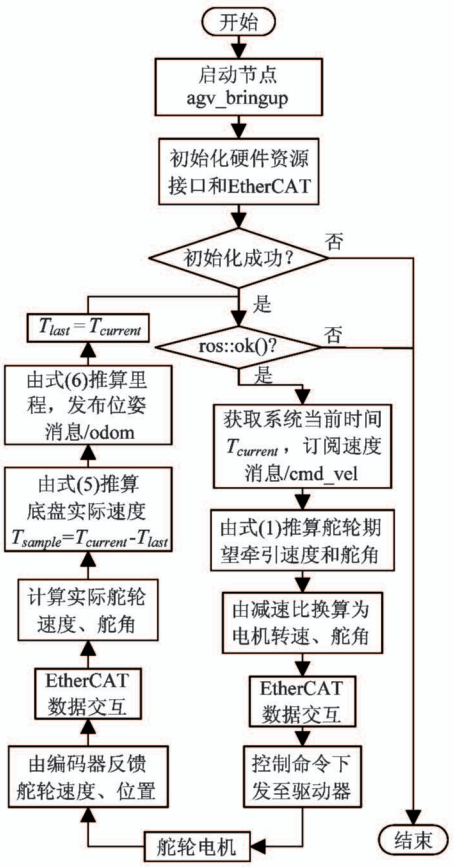


图 5　agv _ bringup 节点工作流程图

3.1　驱动层

驱动层运行于 agv _ bringup 节点中并随工控机同时启动,agv _ bringup 节点主要负责初始化硬件资源接口、EtherCAT 主站、关节状态映射、ROS 与 EtherCAT 的数据交互以及发布机器人里程消息,其设计流程如图 5 所示。

当节点收到来自上层的速度控制消息/cmd _ vel 后,经运动学方程式(1)将其转化为舵轮的牵引速度和舵角,经换算后最终转化为电机转速和转角。电机转速 V_{Li} 和转角 P_i 与舵轮的牵引速度 v_i、舵角 θ_i 的转换关系为

$$v_i = \frac{V_{Li} \times 2\pi \times R}{J \times 60} \tag{7}$$

$$\theta_i = \frac{(P_i - P_0) \times 2\pi}{N} \tag{8}$$

式中,R 表示舵轮半径,J 表示牵引电机减速比,P_0 表示转向电机初始位置,N 表示舵轮旋转一圈电机的位置。系统根据反馈速度和采样周期推算里程计,里程消息通过/odom 发送,包含底盘的位姿及速度信息。

3.2　导航规划层

导航规划层基于 ROS move _ base 设计,move _ base 提供了各类接口,开发人员可根据需求设计导航算法。导航规划层根据驱动层反馈的里程信息/odom、激光雷达数据消息/scan 和代价地图(costmap)实现底盘的导航规划,主要分为全局路径规划(global _ planner)和局部路径规划(local _ planner)。首先,系统根据移动底盘的安全活动半径将全局静态地图上的障碍物膨胀,生成全局代价地图(global _ costmap)。全局路径规划由 A* 算法[19]实现,通过在全局代价地图上规划可行路径,并将规划出的路径信息发送至局部路径规划。局部路径规划则根据激光雷达扫描的机器人周围障碍物生成动态的局部代价地图(local _ costmap),并通过动态窗口法(dynamic window approach, DWA)[20]进行局部避障的路径和运动规划。导航规划层工作流程如图 6 所示。

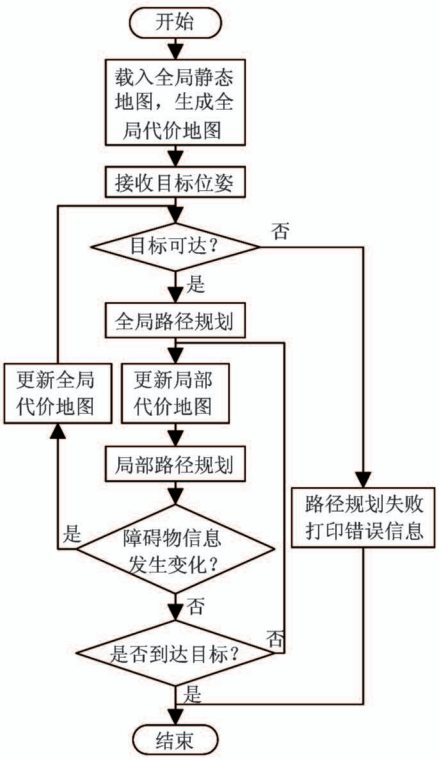


图 6　导航规划层工作流程图

4　交互模块设计

ROS 提供了一款 3D 可视化工具 Rviz，可显示传感器、机器人状态等图像信息，是开发人员调试时必不可少的工具。但是 Rviz 并不直接显示数值信息且缺少易操作的人机界面，因此本文基于 ROS-Qt 插件 ros _ qtc _ plugin[21] 设计客户端人机界面，整体架构如图 7 所示。

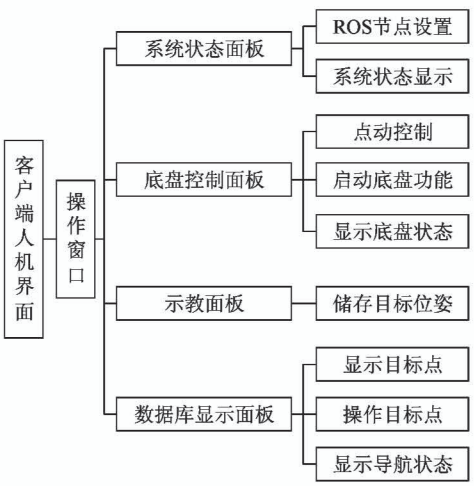


图 7　人机界面架构图

操作界面利用 Tab Widget 控件实现多页面切换。针对移动机器人的功能需求主要完成控制窗口界面和数据库设计，底盘控制系统通过 ROS 分布式节点通信机制与远程客户端实现数据交互。底盘状态信息和界面控制命令处理通过数据处理节点 agv _ interface 完成，并由中间变量完成数据传递，模块化的设计方式方便后续的功能拓展。

4.1　控制窗口设计

控制窗口设计如图 8 所示。其中功能指令、动作指令和速度指令由按钮 QPushButton 控件实现、按钮槽函数响应，状态栏显示位姿消息/odom、舵轮状态信息和导航状态。功能指令通过 system 函数调用控制节点启动指令启动使能、导航等功能。动作指令实现底盘点动控制，按钮槽响应函数根据按键内容将对应线速度、角速度赋值给中间变量，再由 agv _ interface 节点通过/cmd _ vel 消息下发至 agv _ bringup 节点，“控制切换”按钮可将旋转功能按钮切换为平移功能按钮以实现底盘横向平移动作。“回零”功能则发送导航起点位姿，令底盘回到地图初始点。

图 8　控制窗口界面

4.2　数据库设计

数据库用于储存导航目标位姿，通过开源

SQLite 数据库实现,SQLite 具有轻量化、占用资源低等优点,常用于嵌入式设备中[22]。人机界面通过调用封装好的 API 函数 QSqlDatabase 建立数据库、QSqlTableModel 设置数据模型并关联数据。控制界面“位姿示教”功能按钮可以记录底盘当前位姿,位姿示教窗口可手动输入目标位姿并保存。数据库设计流程如图 9。

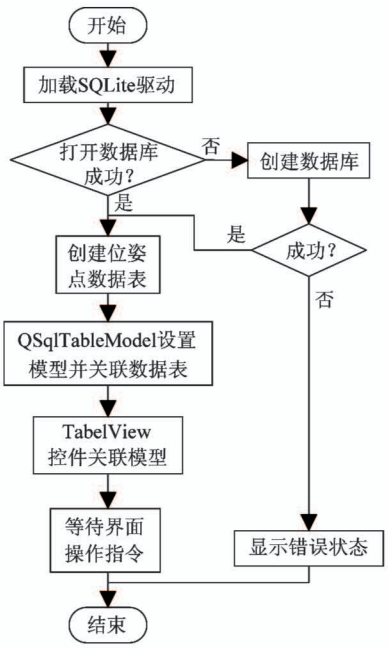


图 9　数据库存储流程图

数据库界面包括发送目标、升降序、删除等功能。发送目标为手动发送选中的目标导航位姿;升降序能改变位姿点顺序;删除选中位姿点。

5　实　验

自主导航实验主要测试移动机器人执行任务时的路径跟踪能力、避障能力和导航精度,实验环境选择教学楼走廊。通过人机界面点动控制底盘完成地图构建,机器人 SLAM 由开源算法 Gmapping[23] 实现。教学楼走廊地图如图 10 所示,地图中网格大小为实际 1 m×1 m。以地图坐标原点为导航起点,远程客户端将保存在数据库中的目标位姿远程下发至底盘控制系统(图 11),控制系统根据给定目标位姿实现机器人自主导航。

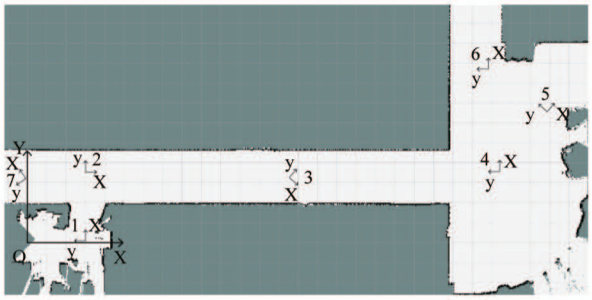


图 10　教学楼走廊地图

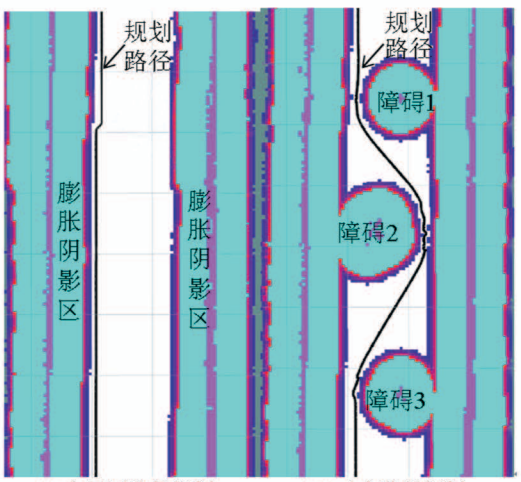
图 11　位姿点数据库

实际导航中需要根据底盘半径设置障碍物膨胀半径,导航层根据系数在地图中无法通行的区域生成阴影区,底盘中心进入阴影区域视为发生碰撞,以表 2 设定的参数执行导航规划。

表 2　自主导航主要参数表

参数	数值
底盘半径/m	0.630
障碍物膨胀半径/m	0.645
机器人质量/kg	180
x 方向牵引速度/m/s	(-0.1, 0.2)
y 方向牵引速度/m/s	(-0.2, 0.2)
最大转向速度/rad/s	0.5
允许位置误差/m	0.05
允许转角误差/rad	0.05

图 12(a)为从地图 4 号点到 2 号点导航的直线走廊全局路径规划,此时机器人并未检测到动态障碍物,因此规划路径为直线。如图 12(b)所示,当出现新的障碍物时,全部路径规划实时更新地图信息并规划出新的可行路径,避开障碍物。图 13(b)为移动机器人实际行进路径图,图中路径由一系列单箭头构成,箭头指向表示当前时刻底盘航向角方向,箭头尾端表示底盘中心所在位置。



(a) 全局面路径规划　(b) 动态路径规划

图 12　直线走廊路径图

(a) 实际导航效果　(b) 实际行进路径

图 13　直线走廊导航效果图

通过 agv _ interface 节点记录底盘 3 个方向的速度,其变化曲线如图 14 所示。

直角走廊导航实验测试移动底盘的转向和过弯能力,由于激光雷达存在视野盲区,当目标位姿在机器人后方时,底盘需要完成转向动作。从地图 6 号点到 3 号点的导航过程中底盘先完成转向动作再前往下一目标点,其导航效果和速度变化曲线如图 15 ~ 图 17 所示。

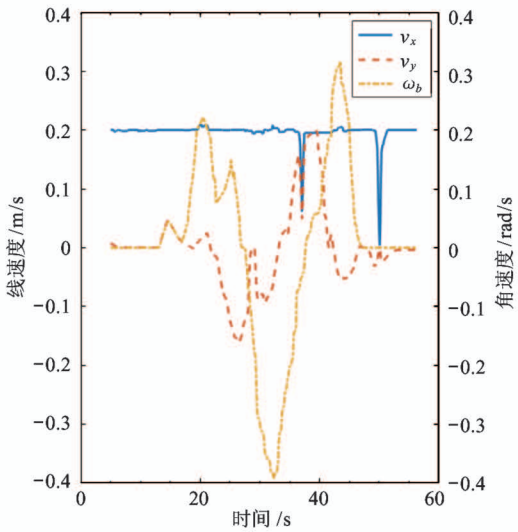
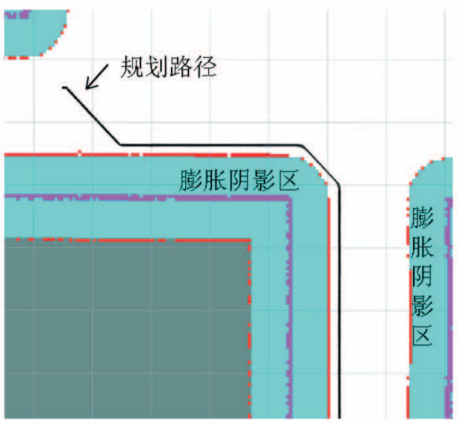
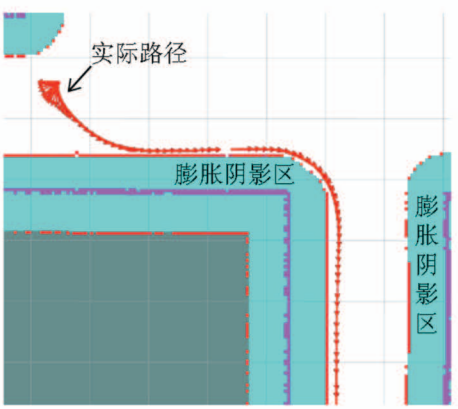


图 14　直线走廊导航速度曲线



(a) 全局规划路径



(b) 实际行进路径

图 15　直角走廊路径图

图 16　直角走廊导航效果

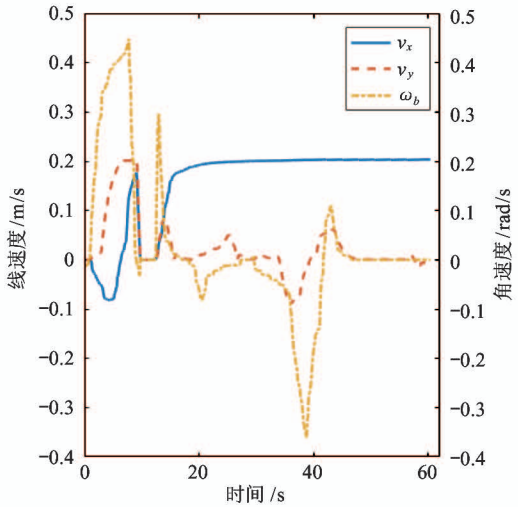


图 17　直角走廊导航速度曲线

将图 10 中的目标位姿储存在数据库中并依次发送至底盘控制系统导航层,实现连续导航。测量 10 次实验的结果,平均位姿误差如表 3 所示。

表 3　导航误差

位姿编号	期望位姿 (x/m, y/m, θ/rad)	平均误差 (x_e/m, y_e/m, θ_e/rad)
1	(3.00, 0.10, 1.57)	(0.02, 0.03, −0.01)
2	(3.00, 3.50, 0.00)	(0.01, −0.02, 0.00)
3	(13.60, 3.20, −1.00)	(−0.02, −0.03, −0.01)
4	(24.40, 3.50, 1.57)	(0.02, 0.02, 0.03)
5	(26.80, 6.60, 1.00)	(−0.04, 0.01, 0.01)
6	(23.10, 13.40, −1.57)	(−0.02, 0.03, 0.01)
7	(0.00, 3.20, 2.00)	(−0.02, 0.00, −0.04)

6　结　论

本文研究了一种全向移动机器人系统,基于 IgHEtherCAT 主站技术、ROS move _ base、Qt 界面等技术框架实现机器人的交互、自主导航功能。控制系统采用平台和框架均为开源,具有低成本的优势;模块化的设计方式降低了系统设计周期,同时易于系统功能的扩展,满足工业机器人控制的基本需求,对开发具有通用易扩展的移动机器人控制系统具有一定的参考价值。

参考文献

[1] Wang T, Tao Y, Liu H. Current researches and future development trend of intelligent robot: a review[J]. *International Journal of Automation and Computing*, 2018, 15(9):1-22

[2] 谭建荣. 智能制造与机器人应用关键技术与发展趋势[J]. 机器人技术与应用, 2017(3):18-19

[3] 白亮亮, 平雪良, 陈盛龙, 等. 分布式移动机器人控制系统设计与实现[J]. 机械设计与制造, 2015(10): 180-183

[4] 管观洋, 陈广锋, 席伟. 基于 Codesys 的两轮机器人控制研究[J]. 自动化与仪表, 2018, 33(4): 16-19

[5] 史兵, 段锁林, 李菊. 基于无线传感器网络的室内移动灭火机器人系统设计[J]. 计算机应用, 2018, 38(1):284-289

[6] Dsouza R, Bainagri P, Dicholkar A. Intelligent security robot[J]. *International Journal of Scientific and Technical Advancements*, 2016, 2(1): 221-224

[7] 于振中, 闫继宏, 赵杰, 等. 9DOF 全方位移动机械臂运动学及其控制系统研究[J]. 高技术通讯, 2011, 21(1): 88-94

[8] Silva F F A, Adorno B V. Whole-body control of a mobile manipulator using feedback linearization and dual quaternion algebra[J]. *Journal of Intelligent and Robotic Systems*, 2017,91(1):1-24

[9] 程春蕊, 刘万军. 高内聚低耦合软件架构的构建[J]. 计算机系统应用, 2009, 18(7): 19-22

[10] 恩里克. 费尔南德斯, 路易斯. 桑切斯. 克雷斯波, 亚伦. 马丁内斯, 等. ROS 机器人程序设计[M]. 北京: 机械工业出版社, 2014: 1-2

[11] 胡春旭. ROS 机器人开发实践[M]. 北京:机械工业出版社, 2018: 1-4

[12] 张鹏. 基于 ROS 的全向移动机器人系统设计与实现[D]. 合肥:中国科学技术大学信息科学技术学院, 2017: 5-7

[13] 韩昊旻，张崇明，陈志红. 基于 ROS 和激光雷达的 AGV 导航系统设计与实现[J]. 电子测量技术，2018，41(8)：112-117

[14] 禹鑫燚，朱峰，柏继华，等. 基于 Win-ROS 的公共服务机器人人机交互系统设计[J]. 高技术通讯，2018，28(11-12)：954-963

[15] 高美原，秦现生，白晶. 基于 ROS 和 LinuxCNC 的工业机器人控制系统开发[J]. 机械制造，2015，53(614)：21-24

[16] Bhaira A，Rudra G，Bag T，et al. Development and implementation of a Linux-Xenomai based hard real-time device driver for PCI data acquisition system (DAS) Card[J]. *International Journal of Computer Applications*，2013，81(12)：24-28

[17] 徐建明，吴蜀魏，吴小文，等. 基于 ROS 和 IgH EtherCAT 主站的 SCARA 机器人控制系统[J]. 高技术通讯，2019，29(9)：876-885

[18] Hrbacek J，Ripel T，Krejsa J. Ackermann mobile robot chassis with independent rear wheel drives[C]// Proceedings of the 14th International Power Electronics and Motion Control Conference，Ohrid，Macedonia，2010：46-51

[19] 王殿君. 基于改进 A * 算法的室内移动机器人路径规划[J]. 清华大学学报：自然科学版，2012，52(8)：1085-1089

[20] Fox D，Burgard W，Thrun S. The dynamic window approach to collision avoidance[J]. *IEEE Robotics and Automation Magazine*，1997，4(1)：23-33

[21] Armstrong L. ROS _ qtc _ plugin[EB/OL]. https://github.com/ros-industrial/ros _ qtc _ plugin：ROS.org，2015

[22] Lv J，Xu S，Li Y. Application research of embedded database SQLite[C]// 2009 International Forum on Information Technology and Applications，Chengdu，China，2009：539-543

[23] Gerkey B. Gmapping[EB/OL]. http://wiki.ros.org/gmapping：ROS.org，2019

Design of mobile robot control system based on ROS and omnidirectional steering wheel

Xu Jianming，Wu Xiaowen，Cai Qizheng，Ni Hongjie

(School of Information Engineering，Zhejiang University of Technology，Hangzhou 310023)

Abstract

An omnidirectional mobile robot control system is designed based on robot opertating system (ROS) and EtherCAT. First，the kinematic model of the two-steering-wheel mobile robot is established，and using Xenomai real-time kernel and IgHEtherCAT master to achieve the low-level driver of the robot. Under the ROS platform，the navigation layer of control system is constructed based on move-base，and the navigation data resource is obtained by inertial measurement unit，motor encoder and lidar. Based on Qt framework，remote client-sidehuman-machine interface is developed，mainly including the control window and database interface. Finally，the autonomous navigation experiment is carried out for the omnidirectional mobile robot，and the experimental results verify the practicability of the designed system.

Key words：robot opertating system (ROS)，omnidirectional mobile robot，autonomous navigation，EtherCAT，steering wheel