

基于 Qt 的移动机器人上位机软件设计与实现

杜召辉, 刘安东

(浙江工业大学 信息工程学院, 杭州 310023)

摘要: 为了实现移动机器人运行过程状态数据的可视化, 以面向对象的 C++ 编程语言, 设计了一款基于跨平台的 Qt 图形库框架的上位机监测软件; 该软件对接收到的状态数据进行解码、分离并绘制移动机器人实时运行轨迹和状态信息波形曲线等功能; 首先概述了自行设计搭建的移动机器人硬件平台的结构和原理以及软件的实现流程; 然后简要介绍了上位机软件的开发环境, 并阐述了软件的各项功能以及其设计与实现过程; 最后以移动机器人路径跟踪运动控制作为实验对象, 验证了所设计上位机软件的有效性。

关键词: Qt; 上位机; 移动机器人; 路径跟踪; 数据可视化

Design and Implementation of Mobile Robot Upper Monitor Software Based on Qt

Du Zhaohui, Liu Andong

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: This paper studies the design problem for a host computer monitor software which is used to realize the visualization of data real-time during the moving of the mobile robot. The software is designed based on the crossing platform framework of Qt graphics library and programmed with the object-oriented C++ programming language. The main functions of the software are such that decode and separate the received data, and draw the real-time trajectory and the state curves of the mobile robot. First, we give a brief introduction of the hardware, software, the self-designed mobile robot platform and development environment. Then, the functions of the host computer software and its design process are stated. Finally, a path following experiment of the mobile robot is given to verify the effectiveness of the designed host computer monitor software.

Keywords: Qt; upper monitor; mobile robot; path tracking; data visualization

0 引言

移动机器人是一类能够借助一些运动部件, 如轮子、履带或机械足等装置, 在较大范围和空间内自由移动的机器人。由于自身结构灵巧, 环境适应能力强等特点, 移动机器人的应用已经渗透到了社会生活的各个领域。如军事领域排雷排爆、敌军情报侦察、有害危险物品处理以及复杂作战条件下的军械物资搬运的机器人; 工业领域作为运输工具的 AGV, 通常可以运送数十吨到几千克不等的钢材、生产模块和电子元器件等物料, 尤其是在物流快件的分拣运送方面, AGV 正得到广泛的投入使用, 极大提高了物流行业的运输效率; 在民用领域, 如家庭机器人中的扫地机器人, 餐厅里专门负责点餐以及膳食的配送工作的移动机器人, 还有一些高级公寓使用移动机器人来执行安保以及自动调度泊车等任务^[1]。此外, 高智能的移动机器人, 无人驾驶汽车, 近几年来作为一个热门项目也正得到越来越多的高科技公司和一些汽车生产厂商的青睐。移动机器人之所以能够得到如此迅速的发展应用, 离不开计算机技术以及控制技术的发展。尤其是移动机器人运动控制技术的发展在其中起到了关键的作用, 有效的运动控制技术为移动机器人能够进一步实现更为复杂的任务打下了坚实的基础。

为了实现移动机器人良好的运动控制效果, 需要对相应控制算法的有效性做出评估。在实验过程中, 通过肉眼观察移动机器人的运行效果往往不够精确, 很难从客观上分析该算法的实际性能表现^[2]。另一方面, 由于环境限制使得研究人员不能直接感知移动机器人的运动控制情况。因此, 如何实现有效的数据可视化则成为一种必要的需求^[3]。本文提出了基于跨平台的 Qt 图形库框架, 以面向对象的 C++ 编程语言, 设计一款人机交互软件来实现移动机器人运行状态数据的可视化。

1 移动机器人结构及原理

本文所提上位机软件的实验对象是一辆自主设计的以视觉导航为主的智能小车, 其底板结构由两个驱动轮以及一个万向轮组成, 实物对象如图 1 所示。该结构的设计既简洁有效又稳定可靠, 保证了智能车平稳运行的前提。

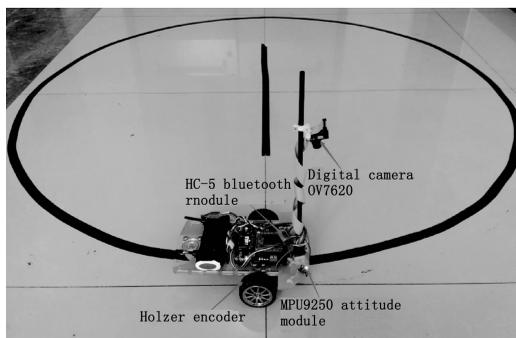


图 1 智能车系统硬件实物图

收稿日期: 2017-09-02; 修回日期: 2017-10-23。

基金项目: 浙江省自然科学基金项目 (LZ15F030003; LY17F030019)。

作者简介: 杜召辉(1991-), 男, 河南南阳人, 硕士生, 主要从事移动机器人, 预测控制方向的研究。

外围硬件电路模块的主要功能及其工作原理如下：

1) 采用拨码开关作为用户输入媒介。

由于拨动开关所引起的电路通断会产生高低电平的转换，处理器通过检测其 I/O 口电平状态而得知用户意图，以次实现了智能车相关初始化参数的设置，避免了程序的多次编译及烧录。

2) 采用霍尔传感器设计智能车的里程计。

霍尔传感器是一种基于霍尔效应而制作的磁传感器，由于其反应灵敏等特点，能够及时响应电机快速运转情况下的检测任务。其工作过程主要由 HoutA 及 HoutB 引脚输出两路测速方波，再通过处理器的定时器输入捕获引脚在单位时间内所捕获的方波数，以确定智能车的实际运行速度。进一步，在一定时间段内，利用处理器所检测到的方波数以及其与驱动轮运行转数的关系，便可设计出智能车的虚拟里程计模块。

3) 采用 OV7620 数字摄像头作为视觉检测模块。

OV7620 是一款 P 制标准的 COMS 摄像头，每秒可输出 30 帧图像数据。处理器通过检测其引脚 VSYN 及 HREF 的电平变化而获取图像数据的采集情况，从而及时存储该摄像头所采集到的图像数据。

4) 采用 MPU9250 作为智能车的姿态解算模块。

MPU9250 是一种集成了三轴加速度计和三轴陀螺仪的 IMU 惯性传感器^[4]，处理器通过对各项惯性数据的解算，便可得到智能车的航向角信息。

5) 采用 STM32F407 单片机为系统主控单元。

STM32F407 是一款基 ARM Cortex™—M4 处理器内核的高性能控制器，常用于工业控制领域。其拥有最高工作频率为 168Mhz，以及高达 1 M 的可编程存储容量，有效满足了智能车系统实时处理大量传感器数据以及进行高负荷运算的需求。

6) 采用 HC-5 蓝牙模块与上位机通信。

蓝牙模块的特点是简单易用，只需将其与处理器的 UART 串口连接，通过对处理器的 UART 接口编程即可实现对蓝牙收发数据的控制。本文所设计的主控软件与智能车间的数据通信，便是以配对的两个蓝牙模块为媒介进行的。

此外，通过对处理器 SPI 接口外接无线通信模块 nRF24L01，实现了多机通信基础，提高了系统的可扩展性。通过 IIC 接口外接显示器模块，可实时显示相关实验数据，便于开发调试。所设计的智能车系统硬件架构如图 2 所示。

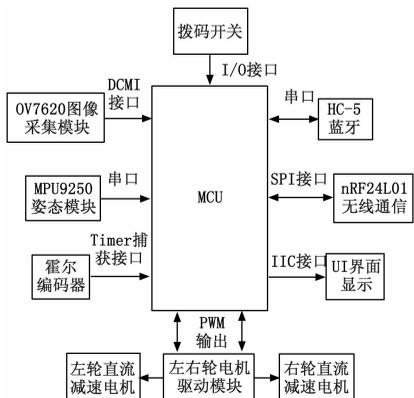


图 2 智能车系统硬件架构图

2 移动机器人软件实现

路径跟踪^[5]是移动机器人运动控制研究的 3 个基础领域之

一。本文以移动机器人路径跟踪这一典型运动控制技术作为上位机软件的测试用例，其中路径跟踪算法控制器采用预测控制方法设计^[6-8]。智能车路径跟踪软件系统实现的简化流程如图 3 所示。

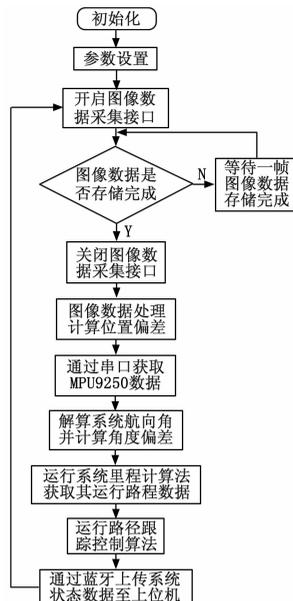


图 3 智能车系统软件流程图

启动过程，将智能车放置在参考轨迹附近处，主控系统完成对附属控制单元以及各外部传感器模块的初始化工作。视觉传感器开始以每秒 30 帧的采集速率实时检测运行路径，并将采集到的图像信息通过 DMA 通道运送至控制器内存，软件系统的运行周期便是以一幅图像的存储完成时间为基准。这样，系统周期性的运行过程总是以一幅图像数据的保存完毕为始。一方面，处理器通过对图像信息进行二值化处理，从而得到智能车与参考轨迹偏差距离的像素级表示，并根据测试整理的像素偏差与实际物理位移偏差的对应关系便可得到智能车偏离参考轨迹的实际距离。另一方面，在每个系统周期内，处理器在中断处理程序中采样一幅 IMU 惯性模块的测量数据，并解算出航向角信息。同时，由测速编码器所设计的动态里程计可实时反馈智能车的运行里程。通过结合智能车的位置偏差、航向角以及运行里程等信息，可以确定智能车在参考轨迹下的唯一位置和姿态。从中提取智能车与参考轨迹的距离偏差 Δx （距离偏差的 x 轴坐标分量）、 Δy （距离偏差的 y 轴坐标分量）和航向角偏差 $\Delta \alpha$ 作为预测控制算法的 3 个状态分量，代入所设计的控制器便可计算出当前状态的控制律。

本文设计的预测控制算法的控制律参数由智能车的线速度 v 和角速度 ω 组成，处理器将上述计算所得的数字量控制律转化为对应的模拟量 PWM 脉冲以驱动智能车的运行。不同的 PWM 脉冲宽度对应不同大小的驱动电流，从而驱动电机的转速大小亦不相同。针对智能车的两个驱动轮，控制器分别给出了对应大小的 PWM 脉冲宽度，实现了智能车线速度和角速度的改变，使其路径跟踪效果逐渐趋于稳定。

实验测试过程，该智能车以恒定的参考速率沿一条半径为 75cm 长的圆形参考轨迹做路径跟踪运动。同时，在每一个运行周期内，将智能车的实时线速度 v 、角速度 ω 以及其与参

考轨迹的距离偏差 Δx 、 Δy 和航向角偏差 $\Delta \alpha$ 三个姿态偏差量编码为一数据帧, 发送至上位机做数据可视化处理。

3 移动机器人上位机设计

3.1 Qt 开发环境简介

Qt 是由 Trolltech 公司基于面向对象 C++ 语言而开发的一种跨平台图形用户界面框架, 使用 Qt 开发的软件, 同样的源代码不需要修改便可以在多数主流的平台编译运行, 且会自动依平台的不同而表现出该平台特有的图形界面风格。另一方面, 使用传统 callback 进行对象间的信息传递无法确定传递参数的正确性, 而 Qt 利用其特有的信号与槽 (signals/slots) 机制有效地解决了这一问题。经过多年的发展, Qt 已经具备了完善的 C++ 图形库, 并且逐步集成了数据库、OpenGL 库、脚本库、XML 库等, 其核心库也加入了进程间通信、多线程等模块, 极大的丰富了 Qt 开发大规模复杂应用程序的能力^[9]。而 Qt Creator 则是一款基于 Qt 的跨平台集成开发环境, 它可以运行于 Windows、Linux/X11 以及 Mac OS X 等桌面操作系统, 其开发界面如图 4 所示。

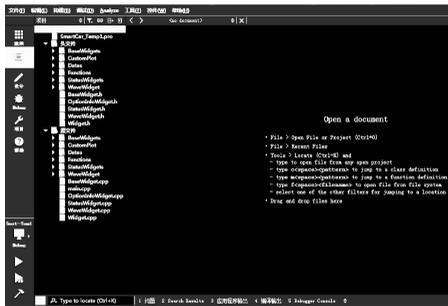


图 4 Qt Creator 开发环境界面

Qt Creator 的开发界面简洁友好, 主要包括代码编辑器、编译器以及一个可视化调试工具和外形设计师 Qt Designer。本文所设计的移动机器人上位机软件是在 Qt Creator 4.0.2 版本下开发, 它基于近期发布的 Qt 5.7.0 框架, 可稳定有效运行, 能够满足项目开发需要。

3.2 上位机设计与实现

QWidget 的是 Qt 图形用户界面的基础控件, 在不指定父类控件的条件下, QWidget 可作为一个独立的窗口或一个顶层窗口显示。本文所设计上位机软件的主界面便是以 QWidget 作为独立的容器窗口, 并将具体的功能部件放入该容器内得以实现。同时, 使用了 QTabWidget 标签来切换各主界面, 以实现不同子任务的切换, 软件的运行初始界面如图 5 所示。



图 5 上位机软件运行界面

该软件主要有 3 个主界面组成: 串口设置和参数命令下发

的基本功能界面 BaseWidget、移动机器人实时运行轨迹的显示界面 StatusWidget 以及移动机器人实时运行状态数据的波形显示界面 WaveWidget。其工作方式可分为在线和离线两种, 两者的工作流程分别由图 6~7 简述。

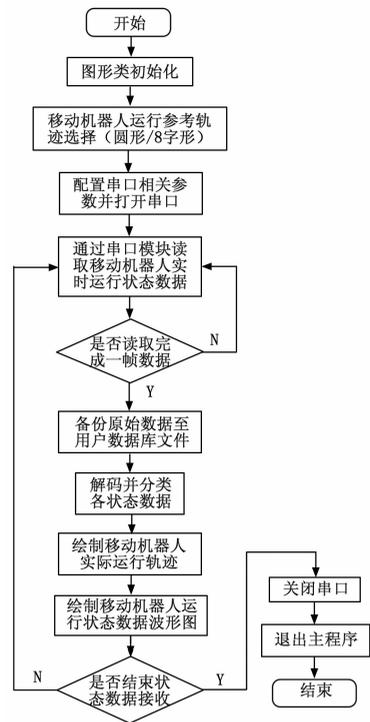


图 6 上位机在线方式工作流程图

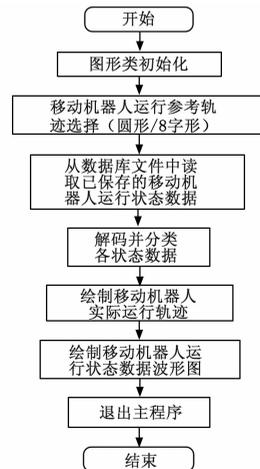


图 7 上位机离线方式工作流程图

在线和离线两种工作方式的唯一区别是两者所处理的数据来源不同。在线方式通过串口读取智能车无线传输的状态数据^[10], 而离线方式则是直接读取保存在用户数据文件中的状态数据。该上位机软件所实现的主要功能如下:

- 1) 解码并分离状态数据帧。

本文所设计的主机软件主要处理的数据对象是移动机器人运行过程中的 5 个状态量, 它们分别是移动机器人的线速度 v 、角速度 ω 以及其与参考轨迹的距离偏差 Δx 、 Δy 和航向角偏差 $\Delta \alpha$ 等。无论在线还是离线的工作方式, 这 5 个原始状态

量总是需要在智能车运行过程，通过无线传输的方式发送至上位机。因此，需要把这些状态量封装为一数据帧，以便于传输以及上位机软件的解码识别。该软件所设计的数据帧格式如表 1 所示。

表 1 移动机器人状态数据帧结构

帧头		距离偏差 x 坐标分量		距离偏差 y 坐标分量		航向角 偏差量 $\Delta\alpha$		线速度 v	角速度 ω
0X FF	0X FF	0XFE/ 0XFD	Δx	0XFE/ 0XFD	Δy	0XFE/ 0XFD	$\Delta\alpha$	v	ω

从该数据帧结构可知，一帧状态数据由 10 个字节组成。其中，帧头由两个字节的相同数值 0xff 标识，3 个姿态偏差量的数值符号各由一个字节表示，0xfe 表示该数值为正，而 0xfd 表示为负， Δx 、 Δy 、 $\Delta\alpha$ 三个字节则分别表示 3 个姿态偏差量的数值部分。字节 v 表示智能车行驶的线速度值，字节 ω 则表示其行驶的角速度值。软件运行过程，通过识别数据帧头标志，即可判定上一帧数据的结束以及下一帧数据的到来，从而解码每一帧有效数据。然后，通过 5 个有效状态数据在帧结构中的固定位置标号以及数值符号标志对其进行分割操作，为之后的数据可视化处理做好前期工作。

2) 基于参考轨迹绘制移动机器人实际运行轨迹。

qcustomplot 是基于 Qt 的一个功能强大的数据可视化图表控件类库，其主要由一个 .cpp 源文件以及一个与其对应的 .h 头文件组成。所述的上

位机软件采用了 qcustomplot 的基本坐标控件类 QCustomPlot 作为智能车路径跟踪的参考坐标系，并以此坐标系使用图形曲线绘制控件 QCPCurve 绘制参考轨迹。同时使用标签选框控件 QRadioButton 实现了不同参考轨迹的切换。以圆形参考轨迹和“8”字形参考轨迹为例，其效果分别如图 8 和图 9 所示。

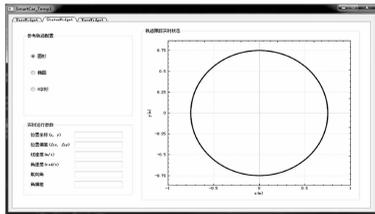


图 8 圆形参考轨迹

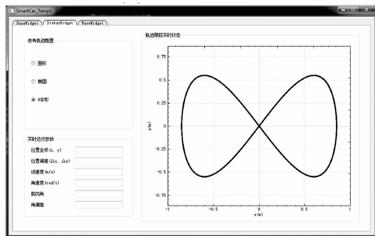


图 9 “8”字形参考轨迹

其中，不同参考轨迹切换的设计方法使用了 Qt 信号与槽 (signals/slots) 的机制，信号端添加在选框控件 QRadioButton 的触发按钮，槽函数则实现了参考轨迹的绘制功能。这样，点击不同的选框控件便会生成不同的请求信号，以此触发不同参

考轨迹的绘制。然后，使用新的 QCPCurve 控件，将功能模块 (1) 中分离出的距离偏差 Δx 、 Δy 坐标分量绘制在对应的参考轨迹中，便实现了移动机器人实际运行轨迹的图形化描述。

3) 绘制移动机器人运行状态波形图。

该功能模块同样使用了控件 QCustomPlot 作为参考坐标系，另外，为了能够更为清晰的查看智能车运行过程的局部状况以及对其运行状态的全局把控，该模块的设计使用了 QCustomPlot 控件的图形缩放功能。同样的，使用 QCPCurve 控件将功能模块 (1) 中分离出的 5 个有效状态数据集分别绘制在对应的参考坐标系下，便实现了移动机器人各运行状态数据的图形化描述。图 10 展示了由移动机器人各状态波形曲线组成的主界面视图。

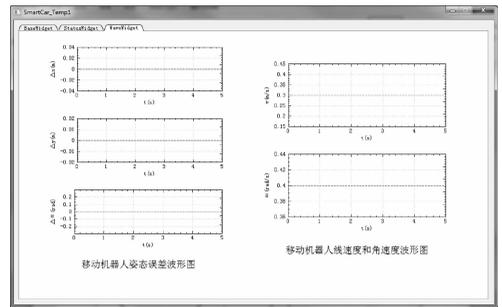


图 10 移动机器人状态波形图主界面

4 实验验证与分析

上位机软件的实验方案如前文所述，使用自行设计的智能车作为下位机平台，以其在圆形参考轨迹下的路径跟踪为例来展开实验。具体的圆形参考轨迹半径为 75 cm，智能车以 0.3 m/s 的线速度和 0.4 rad/s 的角速度作为参考速度，基于预测控制的方法实现路径跟踪的效果。同时，在每一个运行周期，智能车将一帧封装完好的状态数据通过蓝牙模块无线传输至上位机端。上位机软件分别以在线和离线两种方式实现了所述的功能模块，通过从数据帧中分离出智能车与参考轨迹的距离偏差 Δx 、 Δy ，软件绘制了如图 11 所示的相对于圆形参考路径智能车的真实运行轨迹。

路径跟踪轨迹显示

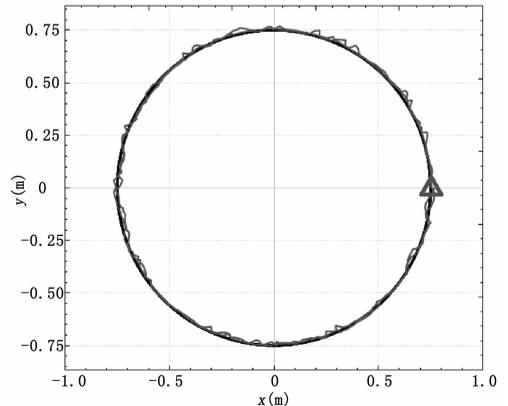


图 11 基于圆形参考路径的智能车真实运行轨迹

图 11 中，智能车以三角标志作为起点，沿圆形参考轨迹的逆时针方向匀速运行两周，其结果可以看出智能车的两次圆周运行轨迹与参考轨迹的拟合程度较高，偏离误差较小。另一

方面, 软件以波形曲线的方式绘制了智能车与参考轨迹的距离偏差 Δx 、 Δy 以及航向角偏差 $\Delta \alpha$ 的动态变化情况。其效果如图 12 所示。

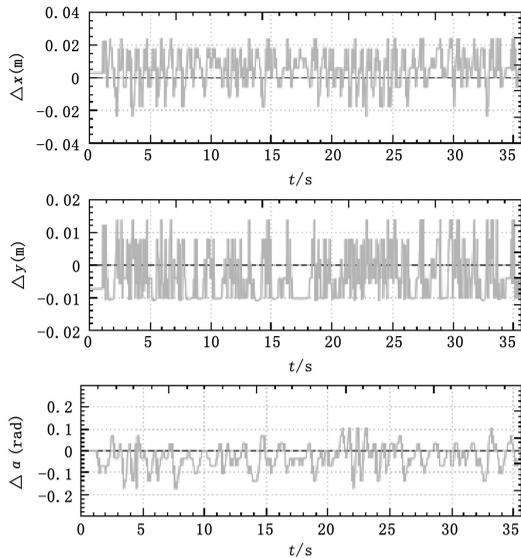


图 12 智能车姿态偏差波形图

该姿态偏差波形图则更为客观的反映了智能车路径跟踪的有效性。通过分析图 12 中的数据变化可知, 智能车运行过程与参考路径的距离偏差 Δx 基本保持在 ± 0.02 m 以内, 距离偏差 Δy 保持在 -0.01 m \sim $+0.014$ m 之间, 航向角偏差则大致保持在 -0.18 rad \sim $+0.1$ rad 之间。

综合图 11 和图 12, 上位机软件从智能车的实际运行轨迹以及其运行过程的态度偏差波形图两个方面, 有效反映了实验对象路径跟踪控制算法的良好表现。与此同时, 图 13 中智能车运行过程平稳的线速度与角速度曲线则表明了受控实验对象良好的动态特性。

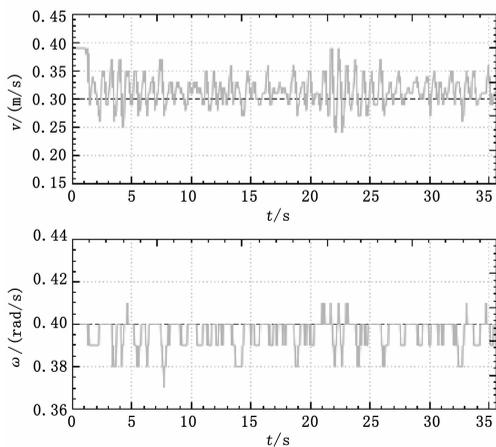


图 13 智能车运行过程的线速度与角速度

5 结论

本文基于跨平台的 Qt 图形库框架, 以面向对象的 C++ 编程语言, 设计和实现了一款界面简洁且功能丰富的人机交互上位机软件。该软件主要实现了以无线传输的方式接收移动机器人发送的状态数据并对其解码、分离移动机器人各状态数据信息并绘制其实时运行轨迹以及状态波形曲线等功能。文中详细阐述了所述上位机软件各功能模块的设计和实现原理, 并以智能车路径跟踪这一典型移动机器人运动控制为实验用例, 实现了其运行时状态数据的可视化, 实验能够清晰地反映出智能车的运动控制效果, 验证了该上位机软件所述功能的有效性。同时, 对于运算性能较弱的下位机控制器, 该上位机软件亦可作为主控单元, 并以无线传输的方式向下位机发送控制指令。此外, 该上位机软件所述功能仅以智能车路径跟踪实验为例验证了其有效性, 对于涉及移动机器人的其他运动控制研究同样有一定的适用性。

参考文献:

- [1] Zhang J, Xuan D, Kim J W, et al. A study of autonomous parking for a 4-wheel driven mobile robot [A]. Proceedings of the 26th Chinese Control Conference [C]. 2007: 179-183.
- [2] 陈端平. 轮式移动机器人监控系统研究与开发 [D]. 广州: 华南理工大学, 2016.
- [3] 吴免利. 锂离子动力电池检测系统上位机软件开发及电池成组算法研究 [D]. 长沙: 中南大学, 2009.
- [4] Zheng F, Liu Y H. A geometric model for fusing IMU into monocular visual localization of 3-D mobile robots [A]. IEEE International Conference on Real-Time Computing and Robotics [C]. 2016: 301-305.
- [5] Shiao Y S, Yang J L, Su D T. Path tracking laws and implementation of a vision-based wheeled mobile robot Proceedings of the Institution of Mechanical Engineers. Part I [J]. Journal of Systems and Control Engineering, 2013, 223 (6): 847-862.
- [6] Ramos C, Martinez M, Sanchis J, et al. Robust constrained receding-horizon predictive control via bounded data uncertainties [J]. Mathematics and Computers in Simulation. 2009, 79 (5): 1452-1471.
- [7] Krid M, Benamar F, Lenain R. A new explicit dynamic path tracking controller using generalized predictive control [J]. International Journal of Control Automation and Systems. 2017, 15 (1): 303-314.
- [8] Liu A D, Zhang W A, Chen M Z Q, et al. Moving Horizon Estimation for Mobile Robots With Multirate Sampling [J]. IEEE Transaction on Industrial Electronics, 2017, 64 (2): 1457-1467.
- [9] 文生平, 陈端平, 刘其信. 基于 Qt 的激光导引移动机器人监控系统设计 [J]. 自动化与仪表, 2016, 31 (8): 6-9.
- [10] 李鹏军, 兰殿星, 宁文斌, 等. 基于 Windows 下 Qt 与脉冲发生器的串口通讯实现 [J]. 计算机测量与控制, 2016, 24 (4): 206-209.
- [10] 徐 骞, VxWorks 下龙芯 3A 中断控制技术的研究 [J]. 现代电子技术, 2017, 40 (14): 36-39.
- [11] 龙芯中科技术有限公司, 龙芯 1E 1F 开发手册 [Z]. 北京: 龙芯中科技术有限公司, 2017.

(上接第 106 页)

- [8] Wind River. Device Driver Developer's Guide Volume 1: Fundamentals of Writing Device Drivers 6.8. Wind River [Z]. 2009.
- [9] Wind River. Device Driver Developer's Guide Volume 2: Writing Class-Specific Device Drivers 6.8. Wind River [Z]. 2009.