

基于 Qt 和 rviz 的自动驾驶调试操作界面改进设计

王法龙, 温长青

(安徽安凯汽车股份有限公司, 合肥 230051)

摘要:针对自动驾驶系统调试过程中指令下发需求,提出一种基于 Qt 和 rviz 的操作界面改进设计。通过自定义指令面板和 rviz 图形化,实现指令的可视化输入;并关联 rviz 与指令消息,实现指令实时传递。经测试,界面改进效果良好,对类似界面开发有参考价值。

关键词:Qt; rviz; 自动驾驶; 调试; 操作界面

中图分类号:U471.15; TP31

文献标志码:A

文章编号:1006-3331(2024)03-0045-05

Improved Design of Operating Interface for Autonomous Driving Debugging

Based on Qt and rviz

WANG Falong, WEN Changqing

(Anhui Ankai Automobile Co., Ltd., Hefei 230051, China)

Abstract: Aiming at the demand for instruction issuing during the debugging process of the autonomous driving system, an improved operating interface design based on Qt and rviz is proposed. By creating a custom instruction panel and utilizing rviz graphics, visual input of instructions is achieved. Additionally, the rviz is associated with instruction messages to enable real-time transmission of instructions. The testing shows that the interface improvements have a good effect and can provide a valuable reference for similar interface development.

Key words: Qt; rviz; autonomous driving; debugging; operation interface

开发自动驾驶系统时,开发人员要通过下发相关自动驾驶指令来测试和标定系统的功能,确保车辆能按预期自动运行。自动驾驶程序通常在 Linux 系统中应用 ROS(机器人操作系统)开发,而 rviz(ROS Visualization)是 ROS 中一个重要的可视化开发工具。由于自动驾驶系统的操作指令通常涉及多个参数和选项,在实际项目中仅使用 rviz 提供的界面及工具可能较为繁琐或难以达到预期效果^[1]。本文运用应用程序开发框架 Qt 对 rviz 的操作界面进行改进后封装在车内自动驾驶系统中,可实现开启和暂停自动驾驶模式、紧急停车、选择自动驾驶区域、选择行车路线、启动全局规划等功能的一键式操作^[2],主要用于开发自动驾驶车辆时的功能测试和参考标定等场景。

1 操作界面改进需求

操作界面是控制指令和自动驾驶核心控制程序之间的媒介,是具有一定顺序的计算机指令和数据集合,可实现人与计算机之间信息、数据的传递和交换^[3]。rviz 作为 ROS 中的强大可视化工具,其原操作界面主要包含视图区、工具栏、状态栏等,允许开发人员自定义显示内容和操作指令^[4]。其中,视图区能实时显示传感器数据、车辆状态信息、环境地图、障碍物检测结果等;工具栏可提供实时向自动驾驶车辆发送交互指令的功能;状态栏可实时显示 ROS 节点状态。但直接使用 rviz 进行系统的调试和标定,需要开发人员通过 Linux 系统的命令窗口进行控制和配置,较为复杂,降低了自动驾驶系统的开发效率。

收稿日期:2024-01-04。

第一作者:王法龙(1989—),男,工程师;主要从事智能驾驶研究工作。E-mail:wangfalong@ankai.com。

为解决此问题,本文提出在 rviz 中关联自定义(根据调试需求)接口消息与指令面板,包括开启和暂停自动驾驶模式、紧急停车、选择自动驾驶区域、选择行车路线、启动全局规划等功能,对 rviz 原操作界面进行改进设计,为自动驾驶的调试和标定提供可视化的一键式操作界面。改进后的操作界面可以让自动驾驶系统的开发人员更深入地考察系统运作,迅速识别并解决潜在问题,同时准确验证算法性能,从而大幅地提升了自动驾驶系统研发的效率和精准度。其软件整体结构如图 1 所示。

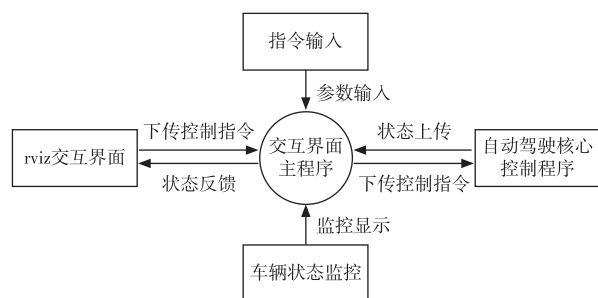


图 1 操作界面改进设计的软件整体结构

2 界面改进设计

2.1 总体思路

本文的操作界面改进设计是基于 Qt 平台。Qt 是跨平台的 C++ 软件开发框架,提供丰富的界面控件、布局管理及多线程、网络通信等功能,助力开发者高效构建稳定、功能全面的图形用户界面(GUI)^[5]。开发人员可通过拖拽方式构建页面,便于 Qt 界面的可视化编辑和修改,以所见即所得的方式完成 Qt 界面的样式编辑和排版^[6]。

ROS 系统中,rviz 主要负责可视化和交互,不直接处理通信、取值和定时发送消息等任务(这些任务通常由 ROS 其他节点来完成)。因此,使用 Qt 对 rviz 进行自动驾驶操作界面的集成式改进设计时,需要确保改进后的界面能与 ROS 其他节点进行通信、获取数据、发送指令等。

2.2 界面改进设计原则

应用 Qt 进行 rviz 界面的改进设计,就是设计指令面板代替原来的 Linux 指令文本。为了使操作者使用指令面板进行自动驾驶可视化指令下发时获得更好的体验感,需要考虑 UI 界面设计的诸多原则。

1) 易用性和统一性原则。设计简单、直观的界面,使操作人员能够轻松理解 and 操作^[7]。使用清晰的布局和可识别的图标,以及易于理解的控件,降低开发人员的学习成本和操作难度。保持相似的控件样式和布局,使操作者能够更快地熟悉和使用界面^[8]。保持界面元素的一致性,使界面整体上具有统一的外观和行为。将同类型的功能选项按钮合并,采用打开后显示下拉菜单的方式选择该种类功能。

2) 操作灵活性和可定制性原则。允许操作者根据需要进行个性化设置。提供灵活的选项和参数,使操作者能够自定义界面,以满足不同的需求。某些操作指令的数据可手动输入,如设置自动驾驶车辆的起始及终止点位、搜索路线和车道索引号等。

3) 导航和操作流畅性原则。界面应具有良好的导航和操作流畅性,操作者能够快速找到所需功能和操作控件。提供直观的菜单、按钮和导航结构,以及友好的界面响应速度,确保能够轻松地导航和操作界面^[9]。

4) 完备性原则。界面应满足操作者的功能需求,提供各项下发自动驾驶指令的操作和功能。确保这些功能在设计中充分体现界面涵盖,不仅包括自动驾驶的相关操作指令,还包含越过自动驾驶程序对车辆进行直接控制指令的操作,如直接对车辆行驶方向、油门、刹车、喇叭、车灯等的直接控制指令^[10]。

2.3 界面改进设计流程

首先应用 Qt 进行用户指令面板设计,以实现直观的操作界面。然后将该指令面板与 ROS 的自定义消息相关联,将指令消息下发给自动驾驶核心控制程序。通过该设计,实现了自动驾驶开发人员与自动驾驶系统之间的有效交互和指令下发。界面改进设计流程如图 2 所示。

2.3.1 设计 Qt 指令面板

指令面板上集成的指令主要包括驾驶模式切换、紧急制动、路径选择、地图切换、全局规划、车辆限速、转向控制等。根据 Qt 界面设计原则对指令面板进行设计,使用 Qtcreator 创建指令面板,将所需的面板空间添加到界面中,并设计相应的属性^[11]。指令面板设计完成后保存 UI 文件,使用 Qt 的 uic 工具将其编译为 C++ 代码文件。将生成的 C++ 代码集成到 user_

panel 的 ROS 功能包中。

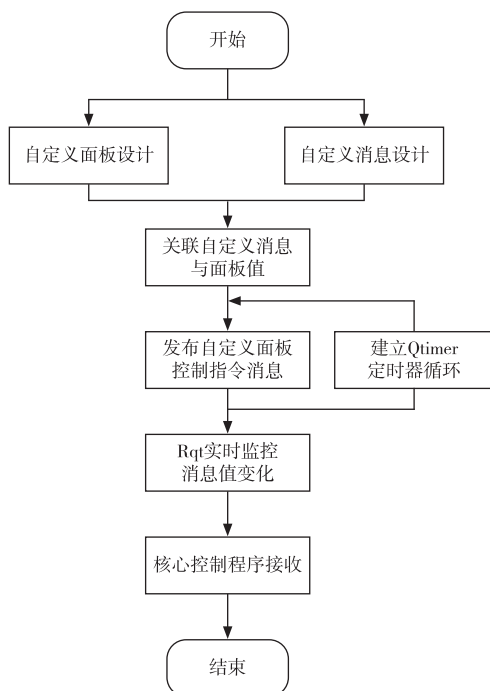


图2 界面改进设计流程图

2.3.2 嵌入自动驾驶指令消息

首先创建自定义消息类型 `Msg_MovingCmd`, 在 ROS 中以“.msg”文件形式定义, 如图3所示。文件中包含多个字段, 每个字段由字段类型(如 `int`、`float`、`string` 等)以及字段名称(`time_stamp`(时间戳)、`region_ID`(地区编号)、`lane_index`(道路索引)、`target_moving_task`(目标运动任务)等组成。



图3 关于自动驾驶指令的自定义消息

完成自定义消息类型后, 需编辑软件包下的 `CMakeLists.txt` 文件和 `package.xml` 文件, 以此增加编译依赖和执行依赖, 从而保证生成相应的源代码文件。在工作空间目录下使用 `catkin_make` 命令, 将编

译自定义消息类型, 生成包含自定义消息类型的头文件 `Msg_MovingCmd.h`, 后续可使用该消息类型进行发布与订阅, 达成发送自动驾驶控制指令的功能。

2.3.3 关联 rviz 面板与自动驾驶指令消息

操作改进后的 `rviz` 界面中的 Qt 指令面板时, 确保有相应的消息指令下发是非常重要的。下面以一个自定义消息类型 `Msg_MovingCmd` 的对象 `rvizCmd` 为例, 详细介绍其过程。将 `rviz` 指令面板中组合框的操作值赋给 `rvizCmd`, 从而获取与自动驾驶操作指令相关的自定义消息。其主要代码如下: `rvizCmd. reg_ID. area ← comboBox_Regin_ID -> cIndex()`; // 将面板内控件的值赋给 `rvizCmd`。 `rvizCmd. target_moving_task ← comboBoxmoving_task -> cIndex()`; `rvizCmd. start_autodrive ← pushButton_Start_autodrive -> isChecked()`;

通过这种方式, 可将 `rviz` 指令面板上的用户操作与指令下发进行有效的关联。因此, 在自动驾驶系统中就能通过解析 `rvizCmd` 对象来获取相应的操作指令, 并将其用于控制车辆的行为。最终自动驾驶开发人员能够利用 `rviz` 界面进行直观、灵活的操作, 从而达到更高效、精准的自动驾驶控制。

2.3.4 发布自动驾驶指令消息

ROS 有 3 种通信机制可供使用, 本文采用话题通信的方法, 通过在话题上发布和订阅消息, 实现不同节点之间的通信。仍以 `rvizCmd` 为例, 在 ROS 节点初始化和创建节点句柄之后^[12], 需要创建一个 `Publisher` 函数, 用于发布名为“`/user_panel/rvizCmd`”的话题, 消息类型为“`autoware_msgs::Msg_MovingCmd`”。

在创建 `Publisher` 函数后, 需要初始化一个“`autoware_msgs::Msg_MovingCmd`”类型的对象 `rvizCmd`。该对象包含了与自动驾驶操作相关的各种成员变量和属性。然后, 从 `rviz` 指令面板中获取操作指令, 并将其赋值给 `rvizCmd` 对象的相应成员后, `rvizCmd` 对象将包含来自 `rviz` 指令面板的操作指令数据。

最后, 利用 `Publisher` 函数将 `rvizCmd` 对象发布到话题“`/user_panel/RvizCmd`”上。通过发布该消息, 其他节点(例如自动驾驶核心控制节点)就可通过订阅该话题来获取 `rvizCmd` 对象中的操作指令数据, 从而

进行相应的控制操作。主要代码如下: void InitializeAllVariables(void); ros::Publisher_pubRvizCmd ← node. advertise < autoware_msgs::Msg_MovingCmd > ("/user_panel/RvizCmd", 1); //赋值 ros::Publisher_pubImuAlignRequest ← node. advertise < std_msgs::UInt8 > ("sensor_interface/ImuAlignRequest", 1); autoware_msgs::Msg_MovingCmd rvizCmd; //初始化对象。

2.3.5 创建 Qt 定时器

QTimer 是 Qt 框架中的一个类,用于创建和管理定时器。它提供了一种机制,可在预定的时间间隔内触发特定的操作。本文利用 QTimer 来实现定时任务,以确保 rviz 指令面板能够定时发送自动驾驶的操作指令消息。通过这种方式,能够确保 rviz 指令面板在预定的时间间隔内定时发送 rvizCmd 等自动驾驶的操作指令消息。这种机制可提供一种稳定可靠的方式,以确保操作指令按照预定的时间间隔发送,从而实现更加准确和可控的自动驾驶控制。主要代码如下: Time_T() //创建一个定时器,用来定时发布消息 { QTimer * output_timer ← new QTimer(this); //定时器的回调函数,按周期调用 output() connect(output_timer, SIGNAL(timeout()), this, SLOT(output())); //设置定时器的周期,100 ms output_timer->start(100); }

3 测试验证

3.1 改进界面测试

改进前的 rviz 界面显示区域如图 4 所示,包括开启和暂停自动驾驶模式、紧急停车、选择自动驾驶区域、选择行车路线、启动全局规划等功能指令按钮。

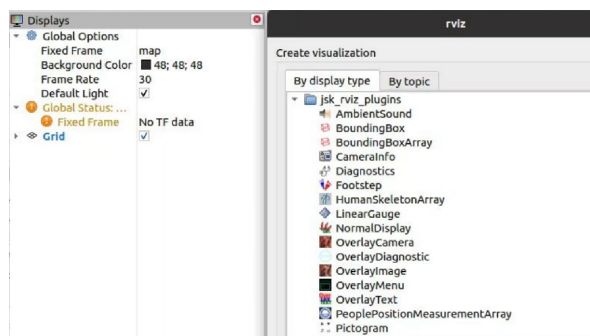


图 4 改进前的 rviz 界面显示区

改进后的界面上的消息发布控件允许选择或输入自动驾驶控制指令的各个属性值,如时间戳、地区编号、道路索引和目标运动任务等。可通过界面上的文本框或下拉菜单来选择或输入这些属性值。当进行指令操作时,界面将根据在自动驾驶指令下发控制界面中输入的属性值,生成相应的自动驾驶控制指令消息,并将其发布到指定的话题上。改进后的 rviz 界面显示区如图 5 所示。

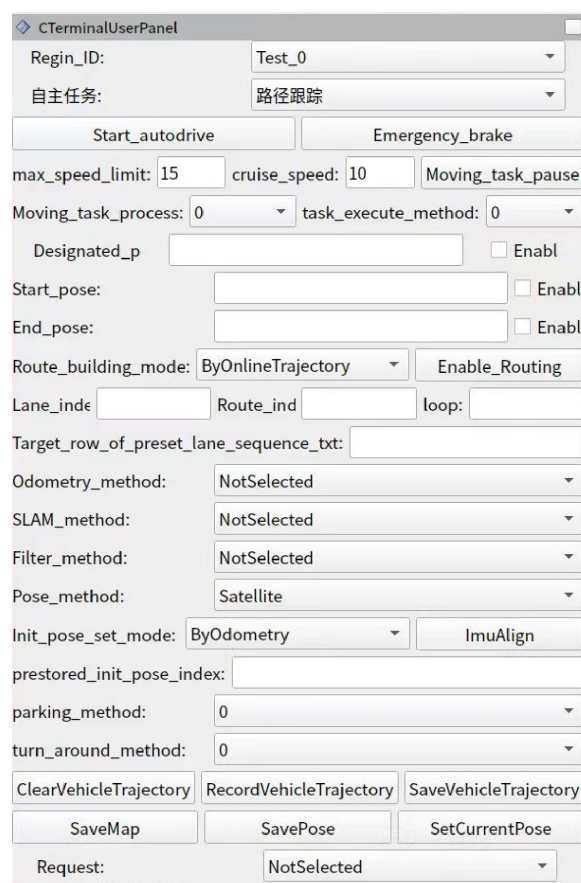


图 5 改进后的 rviz 界面显示区

此外,可通过 rqt(ROS 提供的一个图形化用户接口框架,允许以插件的形式来实现各种图形工具和接口)工具观察消息的变化。界面上的消息发布可用 rqt 来验证是否发送成功,一旦有新的自动驾驶控制指令消息发布到该话题上,rqt 中该消息的值将发生变化,表示已成功发送相应的指令消息,分别对 rviz 中自动驾驶指令界面的操作和 rqt 显示消息值进行记录,测试操作指令消息是否发送成功,如图 6 和图 7 所示。

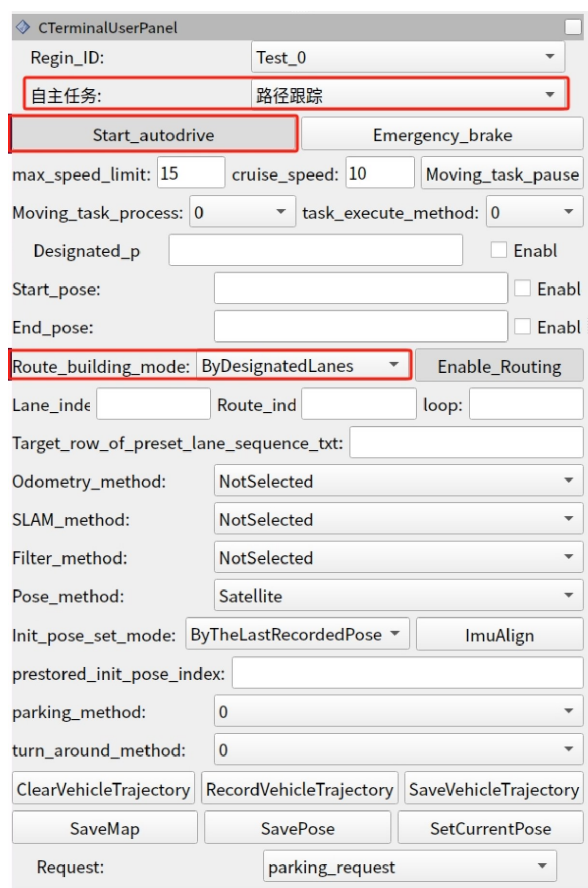


图6 rviz 改进后的界面操作

Default - rqt				
File Plugins Running Perspectives Help				
Topic Monitor				
Topic	Type	Bandwidth	Hz	Value
reserve_data	float32[]		0	
route_building_mode	uint8		5	
route_index	int16		0	
SLAM_method	uint8		0	
start_autodrive	uint8		1	
start_pose	autoware_msgs/MyPose			
state_num	uint16		0	
target_lane_indices	int16[]		0	
target_moving_task	uint16		0	
target_row_of_preset_lane_sequence_txt	uint8		0	
task_execute_method	uint8		0	
time_stamp	float64		1688274524.9496737	

图7 rqt 显示操作指令值

3.2 实车验证

在实车上对改进后的自动驾驶系统操作界面进行自动驾驶系统与整车握手请求、车辆转向控制、紧急制动等指令控制测试。

图8是在自动驾驶系统控制车辆进入自动驾驶状态下握手指令后,整车接收到指令进入自动驾驶模式过程图。图9是在自动驾驶模式下,自动驾驶系统下发转向指令后,整车转向系统相应变化过程。图10是在自动驾驶模式下,自动驾驶系统下发紧急制动指令后整车车速变化过程图。

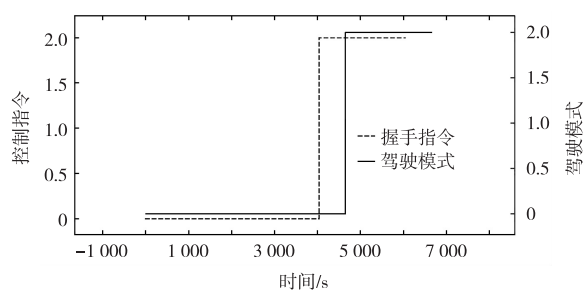


图8 握手指令下发后整车控制模式变化

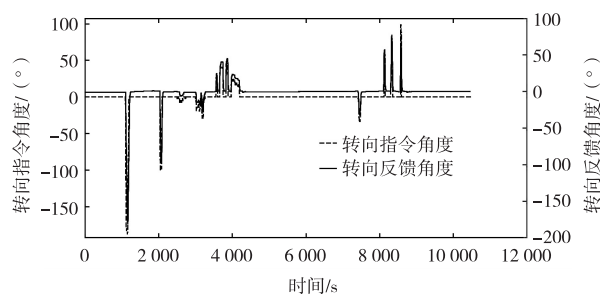


图9 转向指令下发后车辆转向角度跟随

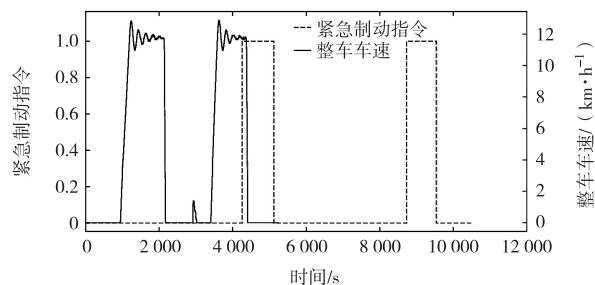


图10 紧急制动指令下发后车辆车速变化

4 结束语

本文测试结果表明,基于 Qt 和 rviz 改进后的自动驾驶操作界面可准确地进行自动驾驶系统中操作指令的可视化输入。界面设计遵循易用性、统一性、操作灵活性和可定制性的原则,以带给更自动驾驶开发者好的用户体验。通过与自定义消息类型相结合,将界面操作与自动驾驶指令下发进行关联,实现了自动驾驶开发人员通过界面进行直观、灵活的操作。通过可视化下发自动驾驶指令,来实现精准的自动驾驶控制。

参考文献:

- [1] 刘思毅. 面向高铁客站的巡检机器人路径规划算法研究 [D]. 南昌: 华东交通大学, 2023.

(下转第 53 页)

布置空间,为整车总布置提供更优方案。

参考文献:

- [1] 中华人民共和国公安部. 机动车运行安全技术条件:GB 7258—2017[S]. 北京:中国标准出版社,2017:4-46.
- [2] 全国汽车标准化技术委员会客车分技术委员会. 营运客车安全技术条件:JT/T 1094—2016[S]. 北京:人民交通出版社,2017:6.
- [3] 刘亚欧,魏仲文,陶仕佳. EBS 在商用车上的应用及性能研究[J]. 重型汽车,2020(1):25-26.
- [4] 郭志军,岳东东,聂彦鑫,等. 纯电动汽车再生制动控制策略研究[J]. 机械设计与制造,2018(1):173-176.
- [5] 樊一鸣. 铰接式城市客车空压机排量与贮气筒容积的匹配[J]. 江苏广播电视大学学报,2000(4):94-95.
- [6] 张伟,郝海生,王领. 客车底盘制动系储气筒容积的计算与选择[J]. 客车技术与研究,2012,34(2):25.
- [7] 唐瑜亮,杨晓云. 纯电动城市客车电动空压机与储气筒容积匹配分析[J]. 汽车电器,2020(10):6-10.
- [8] 梁会仁,王恩前,常颖,等. 电动城市客车制动系统气体容积匹配计算[J]. 客车技术与研究,2017,39(4):49-51.
- [9] 中华人民共和国工业和信息化部. 商用车辆和挂车制动系统技术要求及试验方法:GB 12676—2014[S]. 北京:中国标准出版社,2015:12.
- [10] 刘惟信. 汽车制动系的结构分析与设计计算[M]. 北京:清华大学出版社,2004:125.
- [11] 王陆林,刘贵如. 汽车制动距离估算模型和安全车距控制算法[J]. 农业装备与车辆工程,2019,57(12):23-28.
- [2] 吴勘,胡鑫苑,门龙龙. 无人驾驶电动汽车人机交互界面设计探索[J]. 湖南包装,2021,36(3):100-104.
- [3] 支政. 基于 ROS 和 Qt 的七自由度机械臂人机交互界面的设计和实现[D]. 青岛:中国石油大学(华东),2019.
- [4] 邵刚,谢世琪,段国宁. 基于 Qt 编写的可配置的监控系统[J]. 工业控制计算机,2022,35(3):69-70.
- [5] 范志昊,程跃. 基于 Qt 平台的室内定位系统设计[J]. 长江信息通信,2023,36(8):207-209.
- [6] 李华君. 一种基于 Qt 的应用软件开发与集成方法[J]. 雷达与对抗,2022,42(4):46-52.
- [7] 胡双,吴瑜佳,王语瑛,等. 基于 ROS 的智能机器人可视化自主交互系统设计[J]. 实验室研究与探索,2023,42(4):54-59.
- [8] 廖熹,易克非. 基于嵌入式 Linux 系统下的 Qt 测试软件开发[J]. 兵工自动化,2013,32(8):94-96.
- [9] 吴连港. 基于 Qt 的嵌入式水质检测系统界面软件设计[J]. 农业装备与车辆工程,2021,59(11):140-142.
- [10] 秦艳丽. 试谈移动 UI 设计原则与方法[J]. 电脑编程技巧与维护,2018(5):64-65.
- [11] 黄艳芳. 基于 Qt4 的图形用户界面程序设计与游戏开发[J]. 电子设计工程,2011,19(17):49-53.
- [12] 张静,褚丽莉,陈佳. 浅谈智能家居 ROS 机器人语音控制实现[J]. 信息系统工程,2020(1):16-17.

(上接第49页)