

基于 ARM-Linux 的爬壁机器人远程自动控制系统*

何 宏¹ 赵 磊¹ 张志宏^{1,2}

(1. 天津理工大学自动化学院 天津市复杂系统控制理论及应用重点实验室 天津 300384;

2. 天津广播电视台传输发射部 天津 300072)

摘要:目前,船舶清洗系统的爬壁机器人普遍采用无线工控机进行控制,为了改进无线工控机不能实时显示现场工况和状态信息以及自动寻径控制的缺点,研制了一种控制系统以实现控制的远程可视化,并能自动寻径。系统使用 ARM-Linux 平台,外部电路包括摇杆电路、串口-ZigBee 电路、视频解码电路等。自动控制算法采用基于图像识别的跟踪迹 $Q(\lambda)$ 算法,能够通过图像实现算法快速收敛,完成自动寻径。经过测试,系统能实现对爬壁机器人的自动控制,并且跟踪迹 $Q(\lambda)$ 算法比传统的 Q-learning 算法或 $Q(\lambda)$ 算法收敛速度分别提高 22.8% 和 13.8%。

关键词:爬壁机器人; Linux; 远程控制; 可视化

中图分类号: TN108.4 **文献标识码:** A **国家标准学科分类代码:** 510.8040

Remote automatic controller of climbing robots based on ARM and Linux

He Hong¹ Zhao Lei¹ Zhang Zhihong^{1,2}

(1. Tianjin Key Laboratory of Complex System Control Theory and Application, School of Electrical, Tianjin

University of Technology, Tianjin 300384, China; 2. Transmission and Launch Department,

Tianjin Radio and TV Station, Tianjin 300072, China)

Abstract: At present, the robot of the ship cleaning system is controlled by the wireless industrial controller in common. To solve the problem of that the wireless controller cannot display the working condition or route automatically, a control system which can display the working station remote and route automatically. This system, based on ARM and Linux, includes rocker circuit, serial-port ZigBee circuit and video recording circuit. The automatic control algorithm, the tracking $Q(\lambda)$ algorithm based on image recognition, is to control the robot tracking automatically. From the test results, this system can control the robot automatically, and the speed of convergence of tracking $Q(\lambda)$ algorithm is 13.8% faster than Q-learning algorithm or 22.8% than $Q(\lambda)$ algorithm.

Keywords: climbing robot; Linux; remote controller; visual

1 引言

在目前大型船舶清洗的工作中,使用高压水激流进行清洗船舶壁上的水锈、泥垢以及附着的海洋微生物已经成为了主流。但船舶壁较高的高度和高压水激流的巨大冲击力,使得该项工作具有较高的危险性,所以在这一类特殊领域,爬壁机器人逐步代替了人工进行工作^[1]。在船舶清洗工作过程中,操控者通过远程控制器进行控制。

在船舶清洗自动化领域,国外较早的开展了一系列的

研究,在船舶清洗爬壁机器人控制方面,采用工业无线控制器进行人工远程控制,这种控制器在传输数据方面有较高的可靠性。在国内,该领域虽然起步较晚,但是起点较高,吸取了国外的经验,在技术上与国外的差距逐步减小,爬壁机器人的远程控制使用无线工控机类似的控制平台^[2]。而现在使用的无线工控机只具备数据传输功能,不能够进行远程工况图像传输和爬壁机器人的自动寻迹控制,因此在一定程度上会影响到操控的实时性与操控者决策的准确性。因此,亟待需求一种不仅能够实时自动控

收稿日期:2016-09

* 基金项目:天津市科技支撑重大科技工程专项基金项目(14ZCDGSF00028)、“天津市高等学校创新团队培养计划资助”项目(TD12-5015)资助

制,并且能够将工况实时回传的控制。

工况实时回传可通过将摄像头图像回传至控制端实现,而自动控制则需要一种比较高效的控制算法以实现爬壁机器人的实时自动控制。在传统 Q-Learning 算法和 $Q(\lambda)$ 算法的基础上,提出了一种跟踪迹 $Q(\lambda)$ 算法,该算法相比 Q-Learning 算法和 $Q(\lambda)$ 具有更快速的收敛和寻迹速度。

2 可视化远程控制系统整体架构

这种控制系统针对的船舶清洗爬壁机器人如图 1 所示。

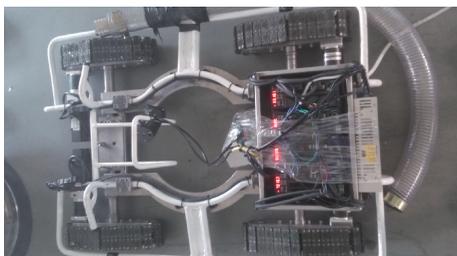


图 1 船舶清洗爬壁机器人

该控制系统的架构可分为硬件、软件和算法 3 个层面,如图 2 所示。

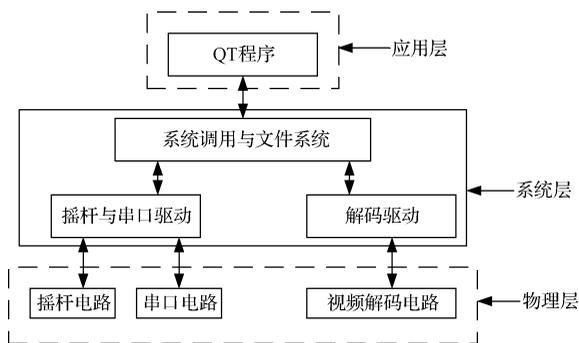


图 2 控制器架构

硬件上以 ARM 9 核心的 S3C2440 为主控芯片,主要包括控制爬壁机器人运动的摇杆电路(移动、转向和系统急停)、与爬壁机器人信号接收端通信的串口-ZigBee 电路、将回传的视频信号进行解码的视频解码电路、实时显示回传数据和工况视频监控的 LCD 显示屏。

软件分为系统驱动层和应用层,系统驱动层为各个硬件电路模块进行驱动,主要有摇杆-串口-ZigBee 驱动程序、LCD 驱动程序、视频解码驱动。应用层程序则是基于 QT 的软件显示程序,将爬壁机器人的状态信息、实时数据和视频监控信息实时显示在 LCD 显示屏上。

自动控制算法则采用使用基于图像识别跟踪迹 $Q(\lambda)$ 算法,该算法是基于马尔可夫决策过程的经典 Q-learning 算法改进,并且结合跟踪迹提取和瞬时差分算法多步回报算法。多步回溯能够将后续产生的数据及时反馈,跟踪迹

能够加快算法收敛。基于此算法来实现对爬壁机器人的自动控制。

3 基于 ARM-Linux 的爬壁机器人远程可视化控制系统的硬件电路

本文中摇杆使用的是 JH-D400X-R2 型摇杆,该摇杆是一种电位器型的摇杆,能够输出不同的阻值,而本项目中爬壁机器人运行过程中以固定速度进行,所以通过基于 LM339 芯片的比较器电路将摇杆作为开关型摇杆,比较电压通过 10 k Ω 、10 k Ω 和 5 k Ω 的电阻并联产生比较电位。电位转换为开关量的电路原理如图 3 所示。

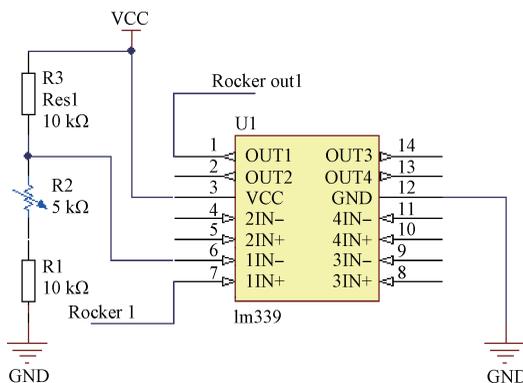


图 3 摇杆电位转换开关量原理

S3C2440 具有 3 个 UART 接口,本方案中使用 UART1 通过 max202 芯片将 TTL 电平转换为 RS232 电平,用 RS232 串口与 ZigBee 模块相连与爬壁机器人的接收器进行通信,电路如图 4 所示。

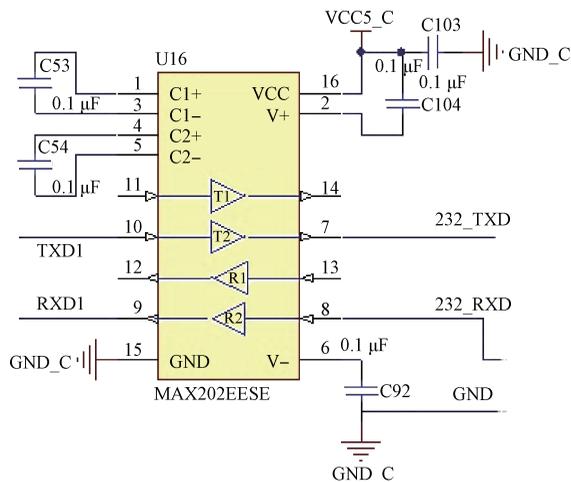


图 4 串口驱动电路

LCD 显示屏采用 7 寸群创 LCD 显示屏,通过 40 pin 排线与 S3C2440 相连,电路如图 5 所示。

视频解码芯片采用 TI 的 TVP5150A 芯片进行解码,该芯片的作用是输入模拟 PAL 信号,输出 ITU BT656 格式的数字视频信号。

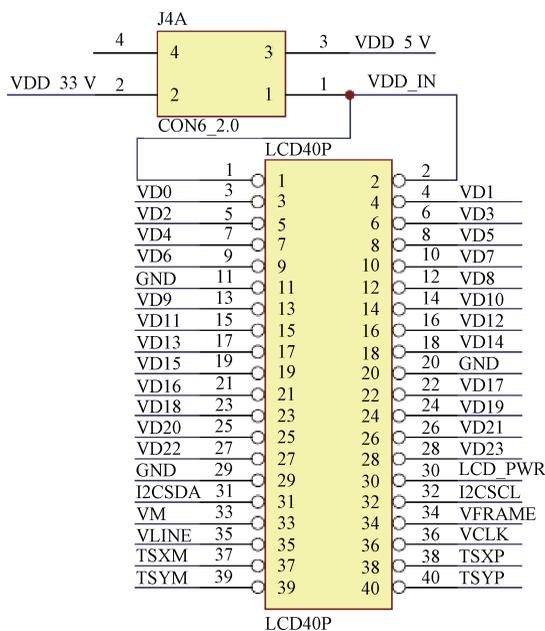


图5 LCD驱动电路

4 基于 ARM-Linux 的爬壁机器人远程可视化控制系统的软件程序

爬壁机器人的控制平台基于 ARM-Linux 操作系统,则首先要对 Linux 系统进行在 ARM 平台上移植。驱动程序和上层的 QT 程序都是运行在 Linux 平台上,整个系统中主要包括:摇杆-串口-ZigBee 驱动、视频解码驱动和 QT 显示程序等子程序。

4.1 摇杆-串口-ZigBee 驱动

在本方案中,控制系统应优先处理手持控制端对爬壁机器人的控制和系统急停,尤其要避免系统进程阻塞时影响对爬壁机器人和急停的控制。本方案采用将摇杆和串口使用同一个内核模块进行驱动,并且在内核模块中以中断方式来处理摇杆状态的检测和串口通信。具体方式如下,在内核模块中申请 EINT1 外部中断(非快速中断)和 UART1 中断(快速中断)。申请中断代码如下:

```
request_irq(eint1_rise, rocker_act_irq, IRQF_DISABLED, "rocker", &rocker_cdev); //摇杆动作
request_irq(eint2_fall, rocker_act_bck_irq, IRQF_DISABLED, "rocker", &rocker_cdev); //摇杆恢复
request_irq(uart1, tvp5150_uart1_irq, IRQF_DISABLED, "rocker", &rocker_cdev); //串口中断
```

当摇杆状态发生变化时会产生一个外部中断,中断处理程序会检测摇杆所发生的变化,然后将摇杆状态变化信息写入 UART1 的发送 FIFO,打开串口发送,退出 EINT1 中断。

此后 S3C2440 会将 UART1 发送 FIFO 中的数据通过串口和 ZigBee 发送到相应的控制端。

通过串口接收系统状态信息,当爬壁机器人发送系统

状态信息时将 S3C2440 的 UART1 接收 FIFO 深度设置为 8,且当 3 个时钟信号后没有接收到后续的数据,则产生一个 UART1 接收中断,在中断处理程序中对接收到的数据进行检测处理,然后将系统状态信息保存到数组中,等待应用层 QT 程序读取。

4.2 视频解码驱动

视频解码驱动主要完成对 S3C2440CAMIF 的驱动和 TVP5150 芯片的驱动。TVP5150 芯片的驱动通过 IIC 对芯片内部的寄存器进行读写,这些寄存器主要包括 0x00(视频源信号选择)、0x30(负责控制芯片信号输出端状态)、0x0d(芯片输出信号选择和比率选择)等。视频接口 CAMIF 的设置初始化包括源视频信号格式、目标视频格式、视频压缩格式、时钟等初始化,然后申请内存将解码后的视频数据存入内存,上层 QT 程序会对该段内存进行读取,用于在 LCD 显示屏上显示。

4.3 应用层 QT 显示程序

应用层 QT 程序实现对系统状态信息的显示,通过触摸屏实现对系统的非紧急控制。QT 程序分为 3 个线程:主线程实现对系统状态的显示,串口读子线程实现对系统状态的读取,触摸屏按键处理子线程实现当触摸屏按键被按下时的处理。

QT 主线程要完成主界面的构建、界面图形的绘制(由若干虚拟指示灯图片表示系统状态,在系统状态发生变化时,由串口读子线程发送变化信号,在槽函数中实现图形变化绘制)、视频图像显示(由帧定时信号和帧显示槽函数实现)和子线程的创建。主要代码如下:

```
MainWindow::MainWindow(QWidget * parent) :
QMainWindow(parent), ui(new Ui::MainWindow)
{
    pp = (unsigned char *) malloc(640 * 480 /
    * QWidget::width() * QWidget::height() * 3 *
    sizeof(char)); //申请保存视频数据的内存
    frame = new QImage(pp, 640, 480, QImage::
    Format_RGB888);
    vd = new VideoDevice(tr("/dev/video0"));
    rs = vd->open_device(); //打开摄像头
    qDebug("cam open fd= %d\n", rs);
    ui->setupUi(this); //建立主界面
    this->setWindowFlags(Qt::FramelessWin-
    dowHint);
    timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this, SLOT
    (play_lcd()));
    timer->start(30); //30 μs 刷新一次视频图像
    QObject::connect(&wig_rs_qt, SIGNAL(para
    _change(int,int,int,int)), this, SLOT(wt_lcd(int,
    int,int,int)));
    QObject::connect(&sig_slot_main, SIGNAL
```

(sig_rvd(int)), this, SLOT(sig_rvd_slot(int)) ;//连接子线程的信号与主线程的槽函数

串口读子线程从摇杆和串口驱动中读取串口接收到的系统状态信息,将读取到的系统状态信息与已有信息做比较,当系统状态发生变化时则发送信号给 QT 主线程,由 QT 主线程的槽函数实现状态图形的绘制。QT 程序中图形绘制主要通过 setPixmap()函数在 lable 上绘制实现,如下:

```
ui->label_1->setPixmap(QPixmap("./blue_light.jpg")); //blue_light.jpg 为图片的路径
```

触摸屏按钮处理子线程用于触摸屏上的按钮按下时调用串口发送相应信息。

5 爬壁机器人自动运行算法

爬壁机器人在清洗过程中会在船舶壁上形成两种不同的区域:已清洗区域和未清洗区域。两种区域有不同的图像特征,通过高斯平滑滤波和二值化分割可以将图像变换为具有清晰边界的二值化图像。这个边界就可以作为爬壁机器人的运行轨迹。

规定爬壁机器人移动时有上、下、停、左、右 5 个动作,在这里用离散矩阵表示 5 个动作值:

$$AS = [0,1;0,-1;0,0;-1,0;1,0] \quad (1)$$

而爬壁机器人在移动时的状态有 3 种:在已清洗区域、在未清洗区域和在分界线上。那么根据这 3 种状态可以设置一个非线性函数作为奖励函数 R :

$$R = \begin{cases} 0.2 & \tau s = 0 \\ -0.2 & \tau s = 1 \\ 1 & boundary \end{cases} \quad (2)$$

式中: τs 为小车所处环境。

跟踪迹 $Q(\lambda)$ 算法中通过使用记忆矩阵来实现记忆使用的参数,定义记忆矩阵:

$$M(t) = [s_t, a_t, r_t, \lambda_t] \quad (3)$$

式中: (s_t, a_t) 用来表征 Q 值更新中 s_t 状态需要更新在 Q 值表中的索引, r_t 表示奖励值, λ_t 表示学习效率。

利用之前的记忆矩阵获取状态链,用迭代公式进行依次迭代变换。表示为:

$$\begin{aligned} & \text{for } k = t-1, t-2, \dots, 2, 1 \\ & Q_{k+1}(s_k, a_k) \leftarrow (1 - \lambda_k)Q_k(s_k, a_k) + \lambda_k \{r_k + \gamma \max_{a \in A} Q_{k+1}(s_{k+1}, a_{k+1})\} \\ & \text{Until } k = 1 \text{ end} \end{aligned}$$

在实现记忆状态被访问时,跟踪迹会有一个“遗忘”过程,以用来对跟踪状态—动作对进行更新。这个“遗忘”过程就是渐进遗忘。一般渐进遗忘会程指数衰减。跟踪迹有累计迹和替换迹,累积迹的计算公式为:

$$e_t(s, a) = \begin{cases} \lambda \gamma e_{t-1}(s, a) & s \neq s_t \\ 1 & s = s_t \end{cases} \quad (4)$$

式中: $e_t(s, a)$ 为 s 在 t 时刻的迹,对应的替换迹应该为:

$$e_t(s, a) = \begin{cases} \lambda \gamma e_{t-1}(s, a) & (s, a) \\ \lambda \gamma e_{t-1}(s, a) + 1 & s = s_t \end{cases} \quad (5)$$

本方案中将跟踪迹的数学模型定义为:

$$e_t(s, a) = \begin{cases} \lambda \gamma e_{t-1}(s, a) & \\ 1 & s = s_t, a = a_t \\ 0 & t = 0 \end{cases} \quad (6)$$

那么,当状态-动作对被遗忘时,有:

$$e(s, a)(\lambda \gamma)^n < \epsilon \quad (7)$$

由于 $e(s, a) > 0, 0 < \lambda \gamma < 1, 0 \leq \epsilon \leq 1$, 所以当

$$n > \left\lceil \frac{\lg(\epsilon/e(s, a))}{\lg(\lambda \gamma)} \right\rceil \quad (8)$$

时对应的状态-动作对将会被遗忘。重新进行状态-动作对选择。一次寻迹结束。

6 测试

自动寻迹过程中跟踪迹 $Q(\lambda)$ 的优势在于算法的快速收敛,为了验证跟踪迹 $Q(\lambda)$ 算法的优越性,设计了一种环境来对比传统的 Q -learning 算法、 $Q(\lambda)$ 算法和跟踪迹 $Q(\lambda)$ 算法在同一环境下的进行学习和自主寻迹的过程。该环境是将一定区域划分为 8×8 栅格的待清洗区域,如图 6 所示。爬壁机器人最初的位置为 $(1, 7)$, 当逐步运行到 $(4, 4)$ 的位置时认定为达到终点目标。

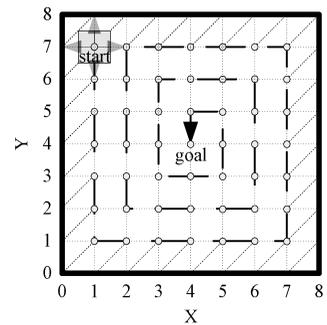


图 6 8×8 栅格待清洗区域示意

这里选择首次状态动作对探索次数、算法收敛探索次数和系统完成整个寻迹过程时合计的行为数量来对比 3 种算法。3 种不同的算法在环境中的循迹数据如表 1 所示,通过表 1 中的数据可知,跟踪迹 $Q(\lambda)$ 算法比 $Q(\lambda)$ 算法在收敛速度上提高了 13.8%,比 Q -learning 算法提高了 22.8%。由此可以看出跟踪迹 $Q(\lambda)$ 算法的适用性和优越性。

表 1 3 种算法的优越性比较

算法	8×8 栅格环境		
	N_f	N_c	N_{ts}
Q-learning	178	181	38 040
$Q(\lambda)$	75	71	34 098
跟踪迹 $Q(\lambda)$	53	60	29 376

7 结 论

综上所述,针对现在爬壁机器人无线工控机不能实时显示现场工况和状态信息的现状,研制了一种可远程可视化控制器,该控制器硬件基于 ARM-Linux 平台,包括摇杆电路、串口-ZigBee 电路、视频解码电路,软件包含各硬件电路的驱动程序与实时显示状态信息和工况信息的基于 QT 的 GUI 程序。通过具有更快收敛速度的基于图像识别的跟踪 $Q(\lambda)$ 算法,实现了对爬壁机器人的自动控制。

参 考 文 献

- [1] 朱志宏,李济泽,彭晋民,等. 微小型壁面检测爬壁机器人移动平台研究[J]. 机械工程学报, 2011, 47(3): 49-54.
- [2] 衣正尧,弓永军,王兴如,等. 船舶除锈爬壁机器人控制系统研究[J]. 机床与液压, 2010, 38(5): 64-67.
- [3] 路青起,习丹丹. 嵌入式 Linux 系统移植[J]. 国外电子测量技术, 2014, 33(12): 78-81.
- [4] 陈小贝,汪志成,叶宝安. 基于嵌入式系统的 PLC 触摸屏控制系统设计[J]. 电子测量技术, 2015, 38(3): 54-56.
- [5] 周泉,马俊,董亚男,等. 基于低功耗微控制器的图像采集系统设计[J]. 电子测量技术, 2014, 37(3): 77-81.

- [6] 陈龙,何成,陶雪娇,等. Linux 下基于 QT 和 OpenCV 的红外图像伪彩色处理研究[J]. 电子测量技术, 2014, 37(2): 44-46, 49.
- [7] 周美丽,延小进,白宗文. 智能车路径信息图像采集控制系统的设计[J]. 国外电子测量技术, 2015, 34(8): 87-89.
- [8] 王金秋,孙晓松,秦华. 基于强化学习的爬壁机器人路径规划方法[J]. 计算机测量与控制, 2013, 21(11): 3093-3095.
- [9] 于占龙. 爬壁机器人控制系统设计[J]. 微计算机信息, 2010, 26(7-2): 200-201.
- [10] 何宏,李宇,张志宏. 基于图像识别的爬壁机器人的路径规划[J]. 工业技术创新, 2015, 2(3): 267-271.

作 者 简 介

何宏, 1960 年出生, 女, 博士后, 硕士生导师, 主要研究方向为检测技术与自动化装置、电磁兼容、控制系统理论等。

E-mail: heho604300@126.com

赵磊, 1989 年出生, 男, 硕士研究生, 主要研究方向为检测技术与自动化装置、嵌入式操作系统。

E-mail: zhaoleichn@126.com

(上接第 55 页)

- [8] 武奕楠,江洁,张广军. 动态 CCD 星敏传感器驱动成像系统设计与实现[J]. 电子测量技术, 2009, 32(12): 17-20.
- [9] 贾传良. 数字式太阳敏传感器研究[D]. 哈尔滨: 哈尔滨工业大学, 2007.
- [10] 杜志强,席红霞. 数字太阳敏传感器高精度电激励信号源设计[J]. 科学技术与工程, 2013, 13(19):

5575-5579.

作 者 简 介

王文靖, 1991 年 11 月出生, 工程硕士, 主要研究方向为检测技术及自动化装置。

E-mail: wen_jingwang@sina.com