

DOI: 10.13382/j.jemi.2017.09.017

基于 ARM-Linux 的爬壁机器人控制器研究^{*}

贾云辉¹ 张志宏² 何 宏¹

(1. 天津理工大学电气电子工程学院 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:针对人工船舶除锈的效率低、危险性高、成本高等问题,研制出爬壁机器人进行船舶除锈。采用了上、下位机的分布式控制方案,对基于 ARM-Linux 的爬壁机器人控制器进行研究,包括遥感器、比较器及其组成电路。通过引用强化学习算法 Q-Leaning 算法,实现了爬壁机器人的强化学习循迹,改进了传统 PID 等算法无法针对环境进行最优化动作策略选择的缺点,提高爬壁机器人在不同环境下循迹的准确性。实验结果表明,基于 ARM-Linux 的爬壁机器人控制系统性能较好,可以满足控制器要求。

关键词:爬壁机器人;下位机;ARM-Linux;Q-Leaning

中图分类号: TP108.4 **文献标识码:** A **国家标准学科分类代码:** 510.8040

Research on controller of wall-climbing robot based on ARM-Linux

Jia Yunhui¹ Zhang Zhihong² He Hong¹

(1. School of Electrical and Electronic Engineering, Tianjin University of Technology, Tianjin Key Laboratory of Complex System Control Theory and Application, Tianjin 300384, China; 2. Transmission and Launch Department, Tianjin Radio and TV Station, Tianjin 300072, China)

Abstract: Aiming at the low efficiency, high risk and high cost in manual ship descaling, the wall-climbing robot is developed. This paper adopts the distributed control scheme of host and slave machine to research the controller of the wall-climbing robot based on ARM-Linux, including the sensor, the comparator, and the composition of the circuit. By using the reinforcement learning algorithm Q-Leaning algorithm, the reinforcement learning tracking of wall-climbing robot is realized. The weakness that traditional PID and other algorithms that can't optimize the action strategy for environment are improved and the wall-climbing robot in different environments tracking accuracy is raised. The experiment results show that the control system of the wall-climbing robot based on ARM-Linux can perform better and meet the requirements of the controller.

Keywords: wall-climbing robot; host and slave computer; ARM-Linux; Q-Leaning

0 引 言

由于船舶长期在海水中浸泡,以及空气中的氧化,表面会生出很多锈迹,同时粘上浮藻等难以清理的垃圾。修船涂装的首要工作就是船舶除锈,清除钢板上的铁锈、油污、旧漆皮等,才能使涂层与钢板的无缝粘合,从而对

船进行保护。一般大型船只每五年必须对其表面进行除锈涂装保养工作,但是船舶每天进坞停放成本昂贵,所以除锈工作必须迅速^[1]。

本课题以爬壁机器人为除锈载体,利用永磁体产生的磁力使爬壁机器人吸附在船体表面上,采用环保无污染的自来水为除锈介质,通过超高压大功率泵机组获得除锈能量,利用真空除锈器进行除锈作业,并通过真空排

污装置回收除锈过程中产生的废水和锈渣^[1-3]。控制系统是船舶壁面除锈爬壁机器人的重要组成部分,是机器人的“神经中枢系统”,负责完成对除锈爬壁机器人的行走和转向,除污等工作。代替了人类的亲自工作,降低人类的危险系数,完全的机械自动化代替了人工省时省力,效率高,造成的环境污染可忽略不计。本课题主要对爬壁机器人控制器进行了详细研究。

1 爬壁机器人的控制系统

根据爬壁机器人控制系统的设计要求,从全局角度考虑,确立爬壁除锈机器人控制器整体方案和控制策略。根据智能爬壁机器人控制系统整体方案,搭建智能爬壁机器人控制系统上位机(手持操作器)和下位机(爬壁机器人端控制器)的硬件平台,其中上位机包括:基于 ARM9 内核的 S3C2440 微处理器及其外围电路、视频解码模块、图传模块接收端、ZigBee 通讯模块、LCD 显示模

块。下位机包括基于 ARM9 内核的 S3C2440 微处理器及其外围电路、ZigBee 通讯模块、图传模块接收端、摄像头、视频解码模块、气动阀门控制器模块和状态监测模块。

爬壁机器人的控制器是基于 ARM9 内核的 S3C2440 微处理器作为主控芯片,充分的实现了对除锈爬壁机器人控制与监控的稳定性和实时性^[4-5]。本文爬壁机器人的主要控制方式是机械摇杆和触摸屏,而实际的情况信息,则需要显示屏实时监控,达到实时性和可靠性。

实时信息和图像的采集以 PAL 制式的模拟摄像头为媒介,可以很好地减少失真。同时以模拟信号的压力传感器对爬壁机器人的外部信息进行采集,外部信息主要包括的高压水和真空压力,模拟信号和数字信号的转换则是通过 S3C2440 内部 A/D 转换器进行转换。现实中机器人运动则是通过摇杆器和触摸屏产生运动信号,传送给 S3C2440 进行 A/D 转换通过 ZigBee 的收发器传送给气动马达和阀门开关控制机器人的运动和清洗动作。爬壁机器人的控制过程原理如图 1 所示。

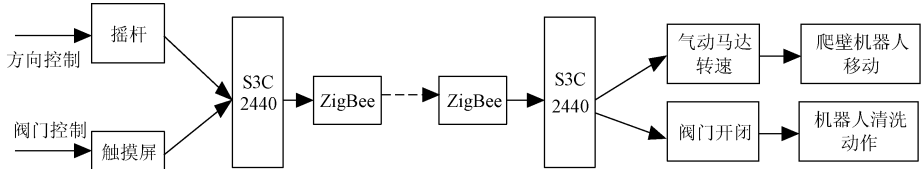


图 1 爬壁机器人控制系统原理

Fig. 1 Schematic diagram of wall-climbing robot control system

控制系统工况与状态信息采集、传送和显示过程原

理如图 2 所示。

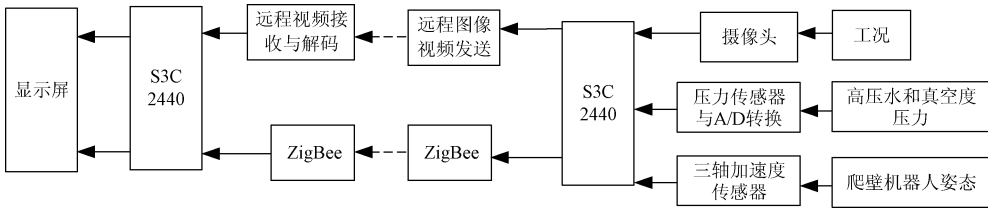


图 2 爬壁机器人图像传输过程

Fig. 2 Block diagram of the image transfer process of the wall-climbing robot

总体来说,通过对爬壁机器人控制指令转换、发送与实现和状态与工况信息的采集、发送与显示共同实现爬壁机器人控制系统的功能^[6]。

2 爬壁机器人的遥控器

爬壁机器人的遥控器是 JH-D400X-R4 型的六向摇杆。控制其前、后、左、右以及速度增加(顺时针旋转)和速度减小(逆时针旋转)。上端的按钮是切换开关,用来切换爬壁机器人的自动和手动控制。摇杆器如图 3 所示。



图 3 摇杆

Fig. 3 The rocker

遥控器内部是 3 个 10 kΩ 的滑动变阻器和一个按键。当滑动变阻器外加电压时,其滑动部分的电压会随着摇杆位置的移动产生相应的变化,从而可以作为控制爬壁机器人运行的指令信号发生机构。爬壁机器人的运行是由

气动马达推动的,速度调整较慢,所以这里的摇杆只用于爬壁机器人的移动方向控制,然后爬壁机器人将会自主寻迹完成接下来的清洗工作^[7]。摇杆驱动电路如图 4 所示。

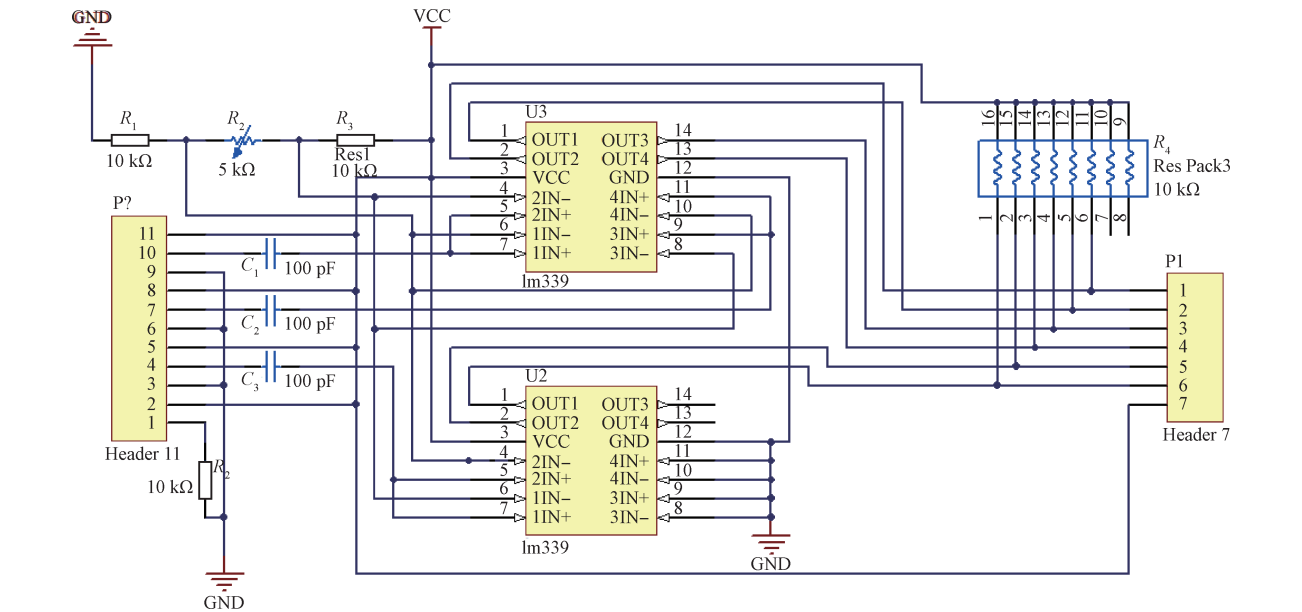


图 4 摇杆电路
Fig. 4 The rocker circuit

电路图中比较器采用的是 LM339 芯片,该芯片具有输入阻抗大,电平转换速度快等优点。两个比较器芯片将摇杆输出转换为开关信号后接入到 S3C2440 的 GPF0 ~ GPF6 7 个端口。这些端口作为信号采集端,采集摇杆状态变化,当摇杆变化时,S3C2440 端口采集到信息后产生外部中断信号,通过无线 ZigBee 模块发送给下位机控制系统,下位机产生相应的动作控制驱动机器人做出相应的运动。

3 爬壁机器人的自动控制算法

基于遥控器只能简单控制爬壁机器人的运动方向,所以还要研究机器人的自动运行。因为在船舶表面进行清洗工作,所以采用了一种基于图片的自主寻迹。在自动运行条件下,通过模拟摄像头监控清洗区域和未清洗区域间的分割线,作为路径的引导线,爬壁机器人能够按照引导线自动行进,以及清洗行进路线。在跟踪循迹引导线的过程中,本课题引入了一种强化学习算法——Q-Leaning 算法,更好的实现了循迹时的速度稳定的最优化。

3.1 Q-Learning 算法

Q-learning 算法是获取最优策略的一种方法,其进行策略估计时采用的是对它的动作值函数进行估计并不是状态值函数^[8-10]。当进行策略改进时,需要对 $Q(s,a)$ 值更新到最大的 $Q(s_{t+1},a)$ 值,与动作 a_t 无关。

Q-learning 的动作更新公式如下：
$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$
$$Q(s,a) \leftarrow r + \gamma \max_{a'} Q(s',a')$$
因为 Q-learning 算法是基于 $Q(s_{t+1},a)$ 最大值进行动作值更新,这样简化了算法分析和收敛性的难度,能够更好的实现算法的快速收敛。为了实验数据以及测试时的效果更佳理想,构建 LE reward 值矩阵 R ：

$$R = \begin{matrix} & \begin{matrix} \text{Action} \\ \text{State} \end{matrix} & \begin{matrix} 0 & 1 & \cdots & n \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ \vdots \\ n \end{matrix} & \begin{bmatrix} r_{00} & r_{01} & \cdots & r_{0n} \\ r_{11} & r_{11} & \ddots & \\ \vdots & & & \vdots \\ r_{n0} & \cdots & & r_{nn} \end{bmatrix} \end{matrix} \tag{3}$$

此外与 R 矩阵同阶次的 Q 矩阵,作用是用来表示 Agent 在很多次的试验中学到的知识,其 Q 矩阵的行为状态,列为行为。

$$Q = \begin{matrix} & \begin{matrix} \text{Action} \\ \text{State} \end{matrix} & \begin{matrix} 0 & 1 & \cdots & n \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ \vdots \\ n \end{matrix} & \begin{bmatrix} q_{00} & q_{01} & \cdots & q_{0n} \\ q_{11} & q_{11} & \ddots & \\ \vdots & & & \vdots \\ q_{n0} & \cdots & & q_{nn} \end{bmatrix} \end{matrix} \tag{4}$$

初始状态时,Agent 作为初学者没有任何经验,此时 Q 矩阵的值全部为 0,引入 Q-learning 算法目的是训练 Agent 通过探索外界环境,对 Q 矩阵进行不断的自我更新^[11-12]。所以,通常情况下,进行越多次的训练,记忆性能越强。

在 Q-learning 系统的学习过程中,学习步骤如下。

1) 初始化

$Q(s,a) \leftarrow$ 任意值,为方便计算一般初始化为 0,给定参数 α, γ 的初值。

2) Repeat

给定初始状态 s ,Repeat 对于每一步 (episode) 根据 ε 贪婪策略选择动作 a ,得到立即回报 r 和下一个状态 s'

$$\delta \leftarrow r + \gamma \max Q(s',a) - Q(s,a)$$
$$e(s,a) \leftarrow 1$$
$$s \leftarrow s'$$

For all s

$$Q(s,a) \leftarrow Q(s,a) + \alpha \delta e(s,a)$$
$$e(s,a) \leftarrow \gamma \lambda e(s,a)$$

Until s 是终止状态,而且 until 所有状态。

3) 输出最终策略

$$\pi(s) = \operatorname{argmax} Q(s,a)。$$

通过算法流程,详细迭代过程如图 5 所示。

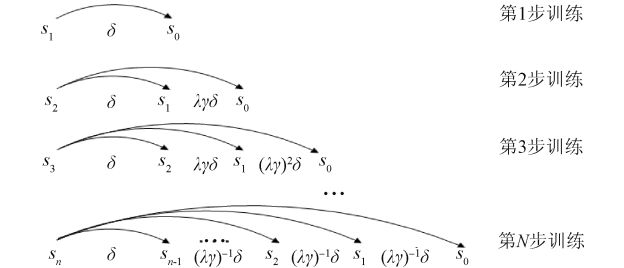


图 5 算法数据更新图

Fig. 5 Algorithm data update

如图 5 所示,当 Agent 每一次的训练结束,都会对经历过得所有的状态动作进行更新,因此又被称为多步 Q-learning 算法。

3.2 爬壁机器人自主循迹模型及参数设定

爬壁机器人的智能自我循迹是通过图像处理后的引导性的轨迹。同时引导线是船舶清洗区域和未清理区域的界限,经过灰度化,二值化等处理方式,使得得到一条清晰的循迹线。

本文的爬壁机器人则是靠网格的方法来模拟自主循迹模型,通过模拟摄像头采集网格内的两条相对较窄的线作为自主循迹引导线。同时任取引导线上的两点 A 和 B ,它们的坐标即所在位置,当两点均在图像中央时视为“吸收态”。如图 6 所示。

爬壁机器人的动作值是根据爬壁机器人双边气动阀

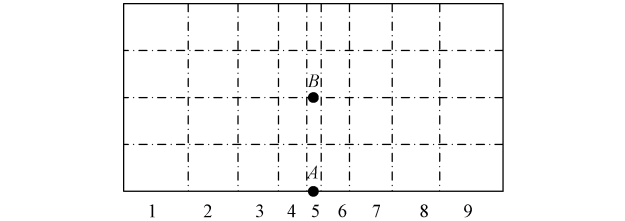


图 6 图像分区及 A, B 点所处状态示意图

Fig. 6 Schematic diagram of the image partition and A, B point state

的开度差值。实际测得动作值与开度差如表 1 所示。在 Q-Learning 算法中强化信号又称奖赏函数直接决定了机器人动作准确性和稳定性^[13]。而本文 Q-Learning 算法中的奖励函数则以离散函数表示。爬壁机器人为了能获得奖励必须达到预期设定的地点,也就是在吸收态时。此时奖励函数设定为式(5)。

表 1 动作值与开度差

Table 1 The difference of action value and opening (Mpa)

	动作值				
	1	2	3	4	5
开度差	右侧 -0.2	右侧 -0.1	无差	右侧 +0.1	右侧 +0.2M

$$R = \begin{cases} 100, & \text{目标状态(吸收态)} \\ 0, & \text{其他} \end{cases} \quad (5)$$

另外,其他的参数规定如下:折扣因子 $\gamma = 0.9$;行驶总距离标期 $Convgol = 0.01$;最大探索次数 $Trials = 500$;单次探索最高步数 $Maxiter = 2\ 000$;最低尝试次数 $Avgtials = 10$ 。

根据 Q-learning 算法原理,可以确定一旦 Q 矩阵经过训练,无论在什么状态,都可以迅速的找到最精确的策略使得 Agent 达吸收态^[14]。通过表 1 的显示可知,当机器人到达吸收态时,其下一步选取的动作一定是 3,即左右两侧的气动马达开度无差值。

本文的爬壁机器人设计的最终目的是它可以自主的跟随引导线并且保持稳定,确保在清洗过程中不产生偏移或偏移量达到最小,以此来实现除锈时的稳定和迅速^[15]。

4 基于 Linux 的软件架构设计

软件架构是指将遥杆的动作值转换成程序所识别的数字信号,同时将遥杆产生变化时的状态信息立即的传递给应用层程序。

遥杆的状态转换是指从无到有即可视为 0 和 1 信

号,对应了电平信号中的高低电平。每当摇杆状态发生变化时,0、1 信号会及时传送到应用层程序,通过中断信息和阻塞机制实现:应用层程序数字信息的更新是通过摇杆所产生的动作变化,当向前拨动摇杆,摇杆发出的指令视为前进,同时会触发一个外部的中断,数字信息相应的由 0 变为 1。机器人执行前进动作,当反方向动作摇杆,数字信息将反向变化由 1 变为 0,暂停机器人动作,原地不动进入等待状态。当下一次动作时,通过同样的中断和阻塞方式将摇杆的状态传递给应用层程序。

根据字符型设备的模型和摇杆设备的功能,本课题的摇杆驱动程序设计结构如图 7 所示。摇杆驱动程序中

定义的设备方法主要分为 open、read、release 3 种。Open 设备方法顾名思义是提供驱动做任何初始化的操作函数,在绝大多数函数中,open 执行的任务是打开设备,无论是设备还是文件首要处理就是用 open 打开,然后再进行之后的动作函数。而 Open 函数的通用语句为:

```
int open( const char * pathname,int flags)

Pathname 是个人用户所要建立的目标文件名,它包括 dev/ttyS0 这个路径。Flags 是每个文件的打开方式,按照读取的差别划分为 3 种不同的打开方式分别为只读方式打开文件、只写方式打开文件、和读写打开方式。而对应的函数为 O_RDONLY、O_WRONLY、O_RDWR。
```

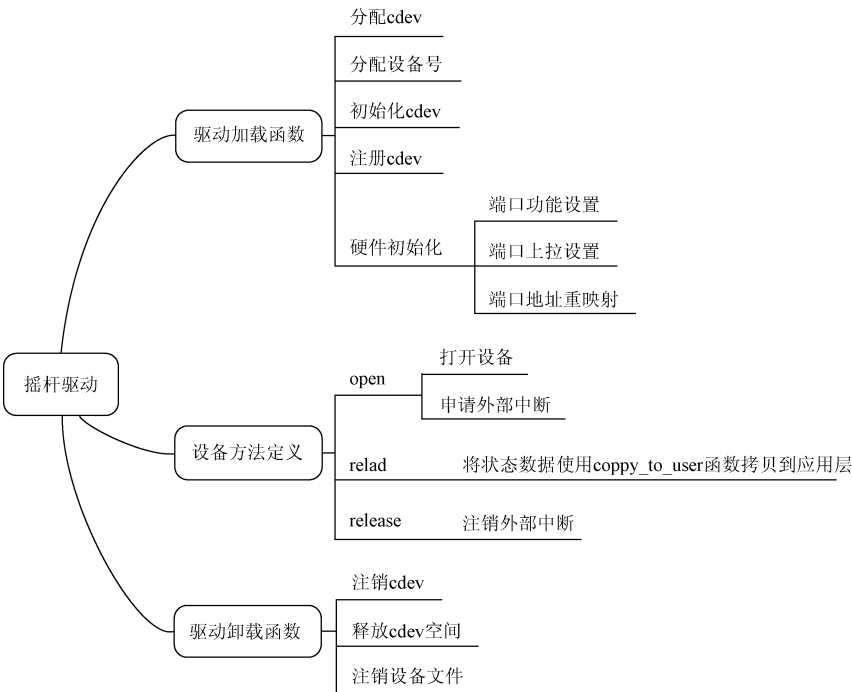


图 7 摇杆驱动模型结构

Fig. 7 The structure of rocker drive model

Read 函数即为读函数,亦可认为 copy,将状态信息 copy 到应用层程序。所用的函数就是 copy_to_user() 函数,它能完全的复制,而且不存在延时,因此摇杆的状态信息可以及时的传送到应用层程序。

Release 设备方法是结束函数,作用和 open 函数恰恰相反,是它的对立函数,release 函数会释放 open 分配在 filp -> private_data 中的任何东西。在 Linux 操作系统中一般用来进行中断和任务队列的注销。驱动任务注销要完成的任务有:从系统中注销 rocker_cdev、注销设备号、释放 rocker_eudev 的内存空间。读取摇杆数据的流程如图 8 所示。

4.1 摇杆读取线程

摇杆读取线程主要完成的任务是读取摇杆的状态信息,当读取到的摇杆发生变化时,发送信号到主线程的状

态显示槽函数。读取摇杆的状态信息的流程如图 9 所示。

摇杆操作的线程是用 rocker_qthread 类的 void run() 函数来执行遥的。首先在 rocker 设备文件中通过 open_() 函数进行系统的调用并初始化,然后通过 read 设备方法进行读数据,因为此时设备里的函数是阻塞的,换句话说当摇杆没有发生动作变化时,系统是阻塞的,阻塞态持续直到摇杆发生动作才会进入就绪态,以此系统进行调度。当系统执行完一次读摇杆数据之后,就会调用串口通信发送函数,发送摇杆变化后所示更新的数据信息。

4.2 摇杆驱动源代码

```
int port_tmp;
char rocker_data[7];
//摇杆的状态数组
```

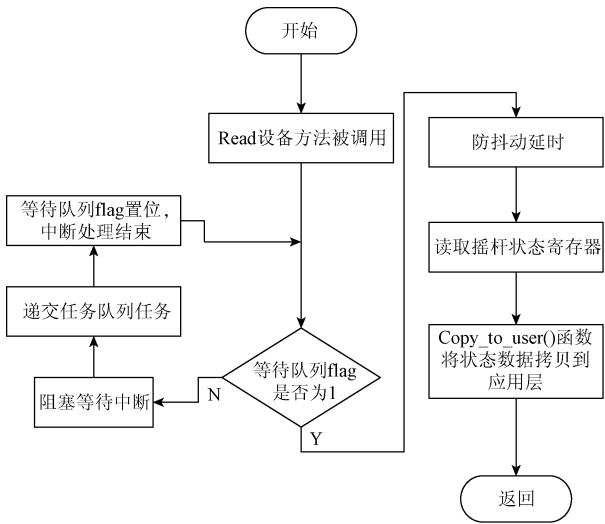


图 8 摇杆信息读取流程

Fig. 8 Flow chart of joystick information read

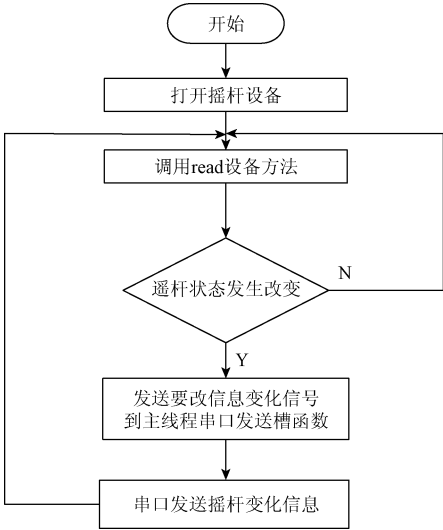


图 9 QT 程序摇杆读取线程流程

Fig. 9 Flow chart of QT program joystick read thread

```
//struct workqueue_struct * rocker_workqueue;
struct work_struct * rocker_work;

//工作队列
int rocker_change_flag = 0;

//等待队列标志
wait_queue_head_t rocker_changed;

//等待队列
struct timer_list rocker_timer;

//延时用于消抖
dev_t rocker_devno;

//摇杆设备号
struct cdev rocker_cdev;
Static struct class * rocker_class;
```

```
//创建设备文件所用到的结构体
static struct device * rocker_dev;

//创建设备文件所用到的结构体

//摇杆的状态数据更新函数
void rocker_data_update()
{
    char rocker_tem;
    port_tmp = * rocker_gpfdat;
    rocker_data[0] = ((port_tmp >> 4) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[1] = ((port_tmp >> 3) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[2] = ((port_tmp >> 7) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[3] = ((port_tmp >> 10) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[4] = ((port_tmp >> 1) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[5] = ((port_tmp >> 6) & 0x01);
    port_tmp = * rocker_gpgdat;
    rocker_data[6] = ((port_tmp >> 0) & 0x01);
}

//摇杆设备的打开函数
int rocker_open(struct inode * node, struct file * filp)
{
    int irq_bac = 0;
    //filp->private_data = &robot_stat;
    irq_bac = request_irq(IRQ_EINT19, on_eint19,
    IRQF_TRIGGER_RISING | IRQF_TRIGGER_FALLING, "
    EINT_19", &rocker_cdev);
    //printfk()
    return 0;
}
```

5 爬壁机器人的系统测试

通过对控制系统的深入研究,设计了爬壁机器人的样机,如图 10 所示。爬壁机器人采用气动马达进行移动驱动。当气源稳定时,气动马达的实际转速与理论转速有一定的差异,当气动马达的输入气源为 0.7 MPa 时,理论值应该为 260 r/min,但其实际速度在 257.2 r/min 左右。在爬壁机器人上的气动马达在使用时添加了减速比为 10:1 的减速机,减速后理论值为 26 r/min,实际值为25.72 r/min,差值为0.38 r/min。如图 11 所示为 QT 显示界面与显示效果图(其中高压水、真空泵未打开,无移动状态)。

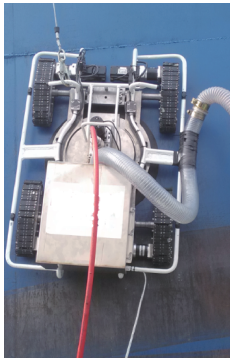


图 10 爬壁机器人样机

Fig. 10 The prototype of wall-climbing robot

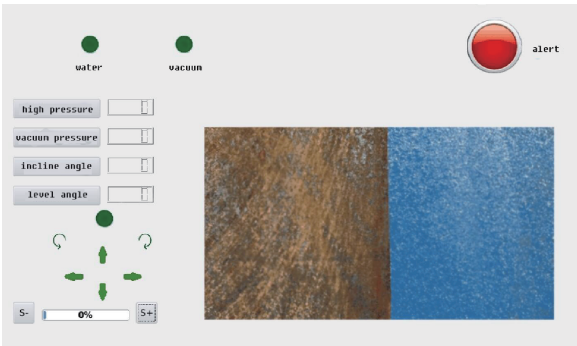


图 11 QT 显示界面与显示效果

Fig. 11 QT display interface and display result

同时为了减小理论值与实际值的偏差,采用左右气动马达差速控制的方式来进行方向控制^[15]。即当需要进行直向移动时,左右气动马达采用相同的转速,当需要变向时,采用左右马达产生差速的控制。表 2 所示为左马达气压设定为 0.5 Mpa 时,左、右马达不同压力差下的转速。

表 2 各电机的实测转速

Table 2 The actual speed of the motor

压力差/ Mpa	右侧马达转速($r \cdot \min^{-1}$)		左侧马达转速($r \cdot \min^{-1}$)	
	右 1 马达	右 2 电机	左 1 电机	左 2 电机
-0.2	162.5	162	217.7	217.6
-0.1	192	191.8	217.7	221.4
0	218.6	217.3	219.3	222.3
0.1	240.9	242.4	217.2	217.5
0.2	257.8	255.4	217.9	218.9

整体上看,采用差速控制方法,左右两侧的马达采用不同的压力差设定值时,能够产生较为稳定的差值。从而保证爬壁机器人直向移动和左右移动的稳定性。同时可以看出,虽然实际转速存在着一定的同步偏差。但伺服电机的转速要经过减速比为 10:1 的减速机减速后才

驱动爬壁机器人的链轮转动,所以,转速偏差除以减速比后,对爬壁机器人的运动速度造成的影响很小,甚至可以忽略。通过实验测试出机器人能够完美的完成控制器的命令,所以说爬壁机器人控制系统应用性能良好,并且能够满足机器人控制的要求。

6 结 论

针对人工除锈的效率低,危险性高,成本昂贵等问题,设计了船舶爬壁机器人进行自主除锈,引入 Q-Learning 算法更好的完成了控制器对爬壁机器人的控制要求。对爬壁机器人的控制器做了深入研究并研制了船舶除锈爬壁机器人的控制系统,对控制系统进行了测试和船舶除锈爬壁机器人样机试验。实践证明:此船舶除锈爬壁机器人控制系统性能良好。

参考文献

[1] 王兴如,弓永军,衣正尧,等. 超高压水射流船舶爬壁除锈机器人力学特性分析[J]. 机床与液压,2008, 36 (10) :67-70.
WANG X R,GONG Y J,YI ZH Y, et al,Analysis of wall rust removal robot mechanical properties of ultra high pressure water jet ship climb [J]. Machine and Hydraulic,2008,36(10) :67-70.

[2] HUANG H C, LI D H, XUE Z, et al. Design and performance analysis of a tracked wall-climbing robot for ship inspection in shipbuilding[J]. Ocean Engineering, 2017,131:59-63.

[3] 丁问司,汪兴潮,唐昌滔. 永磁吸附履带式船舶爬壁机器人结构设计[J]. 机床与液压,2016, 44 (6) :6-11.
DING W S,WANG X CH,TANG CH T. The structure design of the permanent magnet adsorption tracked wall climbing robot [J]. Machine and Hydraulic, 2016, 44(6) :6-11.

[4] 孔庆,马凤伟,周文婷. 手动机器人控制系统设计[J]. 中国科技信息, 2012(14) : 15-20.
KONG Q,MA F W,ZHOU W T. Design of manual robot control system [J]. China Science and Technology Information, 2012(14) : 15-20.

[5] 林文建,钟黎,黎福海,等. 两轮自平衡机器人控制系统设计与实现[J]. 电子测量与仪器学报,2013, 27(8) :750-759.
LIN W J,ZHONG H, LI F H, et al. Design and implementation of two wheeled self balancing robot control system [J]. Journal of Electronic Measurement and Instrumentation,2013,27(8) :750-759.

[6] 李易难,牛燕雄,杨露. 基于 DSP + FPGA 视频图像采集处理系统的设计[J]. 电子测量技术,2014,37(1) :

- 58-61.
- LI Y N, NIU Y X, YANG L. Design of video image acquisition and processing system based on DSP + FPGA[J]. Electronic Measurement Technology, 2014, 37(1):58-61.
- [7] 何宏,赵磊,张志宏. 基于 ARM-Linux 的爬壁机器人远程自动控制系统[J]. 国外电子测量技术, 2016, 35(12):56-60.
- HE H, ZHAO L, ZHANG ZH H. Remote automatic control system of wall climbing robot based on ARM-Linux[J]. Foreign Electronic Measurement Technology, 2016, 35(12):56-60.
- [8] 于乃功, 默凡凡. 基于深度自动编码器与 Q 学习的移动机器人路径规划方法[J]. 北京工业大学学报, 2016, 42 (5):668-673.
- YU N G, MO F F. Path planning method for mobile robot based on depth encoder and Q-learning. [J]. Journal of Beijing University of Technology, 2016, 42 (5):668-673.
- [9] POURPANAH F, TAN C J, LIM C P, et al. A Q-learning-based multi-agent system for data classification [J]. Applied Soft Computing, 2016(52):77-79.
- [10] WEI Q L, LIU D R. A novel policy iteration based deterministic Q-learning for discrete-time nonlinear systems[J]. Science China: Information Sciences, 2015, 58(12):147-161.
- [11] 杨旭东, 刘全, 李瑾. 一种基于资格迹的并行强化学习算法[J]. 苏州大学学报: 自然科学版, 2012, 28 (1):26-33.
- YANG X D, LIU Q, LI J. A parallel reinforcement learning algorithm based on trace [J]. Journal of Soochow University: Natural Science Edition, 2012, 28 (1):26-33.
- [12] 朱洁瑛. 基于迭代学习控制算法的网络机器人控制系统[J]. 计算机安全, 2013(7):58-62.
- ZHU J Y. Networked robot control system based on iterative learning control algorithm [J]. Computer security, 2013(7):58-62
- [13] 宋志章. 基于 Linux 移动机器人控制系统设计[J]. 微计算机信息, 2008, 24(29):240-241.
- SONG ZH ZH. Linux based mobile robot control system design [J]. Microcomputer Information, 2008, 24(29):240-241
- [14] 黄刚. 实时修正偏移量的循迹机器人控制系统研究与实现[J]. 仪器仪表学报, 2015, 36(11):2538-2547.
- HUANG G. Research and implementation of control system of robot tracking real-time offset correction [J]. Chinese Journal of Scientific Instrument, 2015, 36(11):2538-2547.
- [15] 谢其玲. 一款教育机器人控制系统设计[J]. 科技风, 2014(24):74-74.
- XIE Q L. An educational robot control system design [J]. Science and Technology Wind, 2014(24):74-74.

作者简介



贾云辉, 1993 年出生, 天津理工大学在读研究生, 主要研究方向为测试技术与自动化。

E-mail: 734349984@qq.com

Jia Yunhui was born in 1993. He is a M. Sc. candidate in Tianjin University of Technology now. His main research direction is test technology and automation.



何宏(通讯作者), 1960 年出生, 教授, 硕士生导师。主要研究方向为检测技术与自动化装置, 电子信息处理。

E-mail: heho604300@126.com

He Hong (Corresponding author) was born in 1960, professor, M. Sc. Tutor. Her main research direction is detection technology and automation devices, and electronic information processing.