

2019 年全国大学生电子设计竞赛

巡线机器人（B 题）



JD012

2019 年 8 月 10 日

摘要

本文对巡线机器人进行了初步的研究和设计，该机器人由四旋翼飞行器与检测识别模块构成。根据四旋翼飞行器飞行原理，选择合适的无刷电机作为系统动力装置，选取了功能强大且容易开发的微处理器、传感器和相关电子元器件，并做了大量的系统软硬件调试工作，最终完成了整体设计。

本机器人采用 TM4C123GH6PMI 单片机为主控制器，通过六轴加速度陀螺仪 ICM20602 采集的四旋翼欧拉角数据进行 ADRC 处理以保持飞机的姿态稳定；采用 SPL06 气压传感器以及激光测距模块 GY-53L1 控制巡线机器人的飞行高度；使用 OpenMV 实现巡检电力线路及杆塔状态。

关键字：TM4C123GH6PMI 单片机、OpenMV、ICM20602 加速度陀螺仪、GY-53L1 激光测距、ADRC 控制。

目录

一、 系统方案.....	4
1.1 电机的论证与选择.....	4
1.2 空间定位传感器的论证与选择.....	4
1.3 视觉处理传感器.....	4
二、 系统控制理论分析.....	5
2.1 运动模型分析.....	5
2.2 控制方式.....	6
2.3 ADRC 控制.....	7
三、 控制系统硬件与软件设计.....	8
3.1 系统硬件电路设计.....	8
3.1.1 系统总体框图.....	8
3.1.2 飞行控制.....	8
3.1.3 外扩控制电路.....	9
3.1.4 电源管理.....	9
3.2 系统软件设计.....	9
3.2.1 程序功能描述与设计思路.....	9
3.2.2 程序流程图.....	9
四、 测试.....	10
4.1 测试方案与测试条件.....	10
4.2 测试结果与分析.....	10
五、 参考文献.....	10

一、系统方案

本系统主要由电源模块、飞行控制模块、传感器模块组成，下面分别论证这几个模块的选择。

1.1 电机的论证与选择

方案一：普通直流电机。普通直流电机有价格低廉、使用简单等优点，但其扭矩较小，可控性差，此系统要求控制精度高、速度快、且质量要小，所以直流电机一般不能满足要求。

方案二：无刷电机是一种由电调间接控制转速的装置，具有体积小、重量轻、扭矩大的优点。而且，电调控制无刷电机具有快速响应，控制精度高等优点。

综合以上两种方案，选择方案二。

1.2 空间定位传感器的论证与选择

方案一：HC-SR04 超声波模块性能稳定，测量距离精确，盲区小，用于测量四旋翼飞行器距地面的距离。但在实际测试中发现四旋翼飞行器在飞行过程中螺旋桨的转动会对超声波干扰较大，测量距离误差较大。

方案二：GY-53L1 数字红外测距传感器模块。工作电压 3-5v，低成本，功耗小，体积小，安装方便。模块另外可以设置单独传感器芯片工作模式，GY-53-L1X VL53L1 作为简单传感器模块，MCU 不参与数据处理工作。

方案三：GY-53L1 数字红外测距传感器加光流传感器，光流传感器能快速的检测水平移动。控制简单。光流传感器对外部光照亮度有很大的依赖性，且只能在一定高度内进行检测。

考虑到测试环境在屋内，飞行高度较低等因素，我们选择方案三，采用激光加光流传感器进行空间定位。在光线充足的情况下，可以满足题目相关要求。

1.3 视觉处理传感器与巡检方案

OpenMV 它是一个可编程的摄像头，通过 MicroPython 语言，可以实现逻辑应用。而且摄像头本身内置了一些图像处理算法，我们可以很容易地在 OpenMV 所看到的内容上运行机器视觉算法，以便在短时间内进行巡检和拍照存储。赛道分区示意图如图 1 所示：

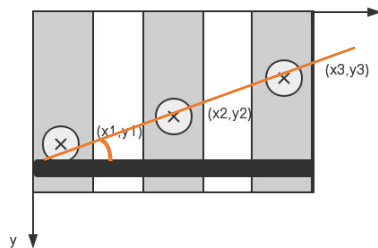


图 1 赛道分区示意图

分区巡线方式：通过巡线机器人实际飞行轨迹在各个区域的点与线缆的偏航角 α 反馈给飞控以达到巡线目的。 $\alpha = \arctan\left(\frac{y_3 - y_1}{x_3 - x_1}\right)$ 。

因在巡检时需要对塔杆上二维码与线缆异物上条形码进行拍照存储，因此我们又增加一个 OpenMV 进行专门识别二维码与条形码并拍照存储。

二、系统控制理论分析

2.1 运动模型分析

首先以机器人起始点建立绝对坐标系，以机器人机身重心处建立无人机的相对坐标系。我们定义我们将无人机的位置为 $L=[x,y,z]^T$ ，定义无人机的俯仰角，翻滚角，偏转角为 $an=[\phi,\theta,\psi]^T$ 。

我们可以将角速度转换成如下形式：
$$\omega = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \cdot an$$

三维空间的欧拉角旋转矩阵为：
$$R = \begin{bmatrix} c_\phi c_\psi - c_\theta s_\phi s_\psi & -c_\psi s_\phi - c_\phi c_\theta s_\psi & s_\theta s_\psi \\ c_\theta c_\psi s_\phi + c_\phi s_\psi & c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\psi s_\theta \\ s_\phi s_\theta & c_\phi s_\theta & c_\theta \end{bmatrix}$$

通过能量守恒，我们知道电机在给定时间内消耗的能量等于螺旋桨上产生的力乘以它所排出的空气移动的距离。

可得 $P \cdot dt = F \cdot dx$ ，即 $P = F \frac{dx}{dt}$ 。由此可得 $P = T v_h$

其中 T 为四旋翼的推力， v_h 为悬停时的速度， P 为悬停时消耗的功率， P 与角速度 ω 成正比。

由此可以推出，四旋翼的总推力为：
$$T_s = \sum_{i=1}^4 T_i = k \begin{bmatrix} 0 \\ 0 \\ \sum \omega_i^2 \end{bmatrix}$$

设四旋翼运动时受到的阻力为 f ，假设 f 四旋翼移动速度成正比：
$$f = \begin{bmatrix} -k_d \dot{x} \\ -k_d \dot{y} \\ -k_d \dot{z} \end{bmatrix}$$

由此可以讲运动模型概括为：
$$L = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R T_s + f$$

2.2 控制方式论证与选择

巡线机器人结构属于 X 型分布。即螺旋桨 1 和 4 与 2 和 3 关于 X 轴对称，螺旋桨 1 和 2 与 3 和 4 关于 Y 轴对称如图 2 所示：

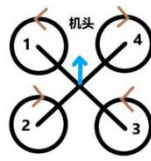


图 2 X 型分布

方案一：PID 算法是将偏差的比例（Proportion）、积分（Integral）和微分（Differential）通过线性组合构成控制量，用这一控制量对被控对象进行控制。但 PID 对环境变化敏感，机器人的螺旋桨在高速旋转时，受到很强的压缩空气阻力，PID 的作用力要很大才能维持稳定的速度，但是当低速时，空气阻力非常小，在 PID 作用力下，螺旋桨会震动、不稳定。

方案二：ADRC 控制方式，继承了经典 PID 控制器的精华，对被控对象的数学模型几乎没有任何要求，又在其基础上引入了基于现代控制理论的状态观测器技术，将抗干扰技术融入到了传统 PID 控制当中去，最终设计出了适合在工程实践中广泛应用的全新控制器。传统 PID 控制都要等到误差发生后，才能去补偿控制，而 ADRC 将观察到的扰动第一时间补偿到输出端。ADRC 在系统抗干扰，模型参数变化时的控制性能比传统的 PID 控制器有很强的优势。

综合以上两种方案，选择方案二。

2.3 ADRC 控制

ADRC 的核心有三大模块：微分跟踪器、扩张状态观测器和状态误差反馈控制律。基于 ADRC 的四旋翼飞行控制器，该控制器能够无稳态误差地实现悬停控制、轨迹跟踪控制。跟踪微分器安排过渡过程，加快响应速度，同时还可以用于对数据进行滤波。扩展状态观测器设计了一个扩展的状态量来跟踪模型未知部分和外部未知扰动的影响，然后给出控制量补偿这些扰动。非线性误差反馈控制律给出被控对象的控制策略。ADRC 控制结构如图 3 所示：

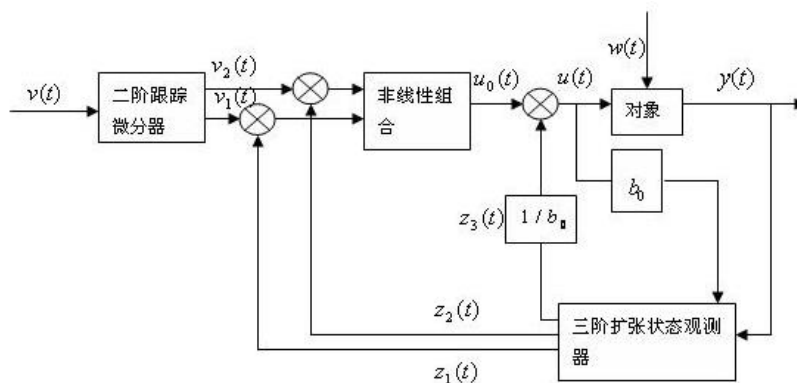


图 3 ADRC 控制结构

跟踪微分器的阶跃响应：

$$v1(k+1) = v1(k) + h * v2(k)$$

$$v2(k+1) = v2(k) + h * fhan(v1(k) - v(k), v2(k), r, h)$$

扩张状态观测器：

$$x1 = x2; x2 = a(t) + u(t); a(t) = f(x1, x2, \omega, t)$$

$$y = x1$$

$$z1(k+1) = z1(k) + h * (z2(k) - \beta1 * fal(\varepsilon1, \alpha1, \delta))$$

$$z2(k+1) = z2(k) + h * (z3(k) - \beta2 * fal(\varepsilon2, \alpha2, \delta) + b0 * u(k))$$

$$z3(k+1) = z3 - h * \beta3 * fal(\varepsilon1, \alpha2, \delta)$$

非线性控制策略方程：

$$e1(k) = v1(k) - z1(k)$$

$$e2(k) = v2(k) - z2(k)$$

$$u0(k) = kp * fal(e1(k), \alpha p, \delta0) + kd * fal(e2(k), \alpha D, \delta0) + ki * fal(e1, \alpha i, \delta0)$$

$$u(k) = u0(k) - z3(k) / b0$$

三、控制系统硬件与软件设计

3.1 系统硬件电路设计

3.1.1 系统总体框图

系统总体框图如图 4 所示：

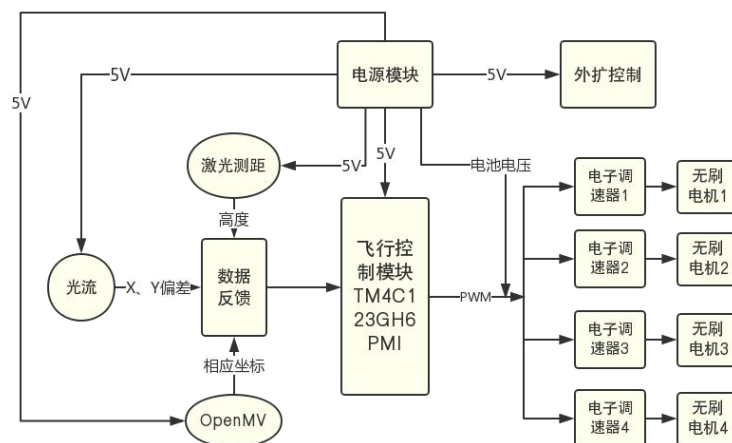


图 4 系统总体框图

3.1.2 飞行控制原理图

飞行控制模块是控制系统的核心部分。它在每个控制周期内实时处理传感器采集的数据和飞行器的姿态信息，完成 ADRC 控制的算法，得到四旋翼飞行器的姿态和位置信息，计算出

控制量，转化为相应的控制信号驱动四个电机工作，保持四旋翼飞行器稳定飞行。原理图如图 5 所示：

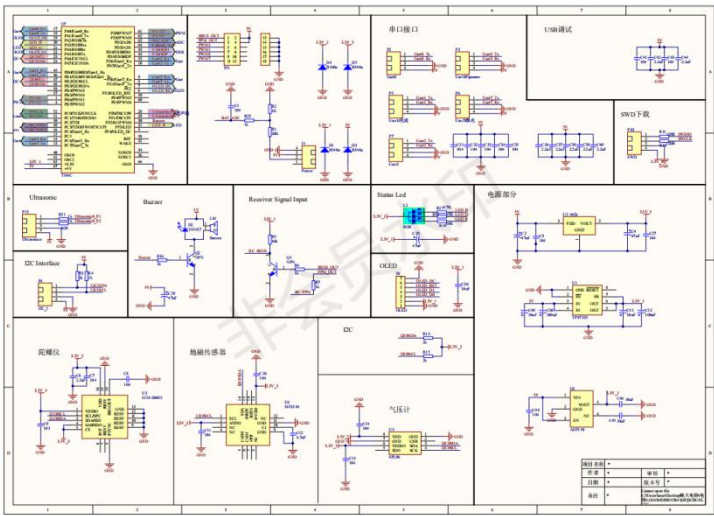


图 5 飞行控制

3.1.3 外扩控制电路

外扩控制电路是为了实现四旋翼飞行器执行相应的控制动作，激光笔实时指示所巡路线，以及实现对飞控信息的转发。

3.1.4 电源管理

电源由一块 11.1~12.6V 的锂电池提供，给四个电调供电，电源通过稳压模块稳出 5V 给飞控及外扩板供电，这样可满足各个小系统的电源要求。

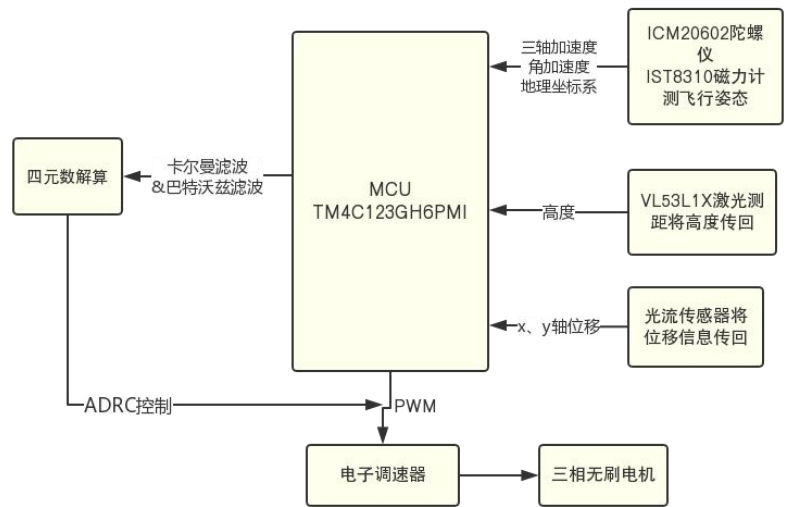
3.2 系统软件设计

3.2.1 程序功能描述与设计思路

软件部分主要实现传感检测和 PWM 输出的更改。

- (1) 传感检测功能：设置高度和角度的输出信号类型。
- (2) PWM 输出：根据检测的数据，通过 ADRC 算法更改 PWM 的输出。

3.2.2 程序流程图



四、测试

4.1 测试方案与测试条件

- 1.测试方案：模拟考场进行测试
- 2.测试条件：检查多次，仿真电路和硬件电路与系统原理图完全相同，并且检查无误，硬件电路无虚焊。

4.2 测试结果与分析

通过模拟考场环境对巡线机器人进行的测试的结果如下：

	具体功能要求	实际测试结果
基本部分	巡线机器人从距 A 杆塔 1 米范围内的起飞点起飞，以 1 米定高绕杆巡检，巡检流程为：起飞→杆塔 A→电力线缆→绕杆塔 B→电力线缆→杆塔 A，然后稳定降落；巡检期间，巡线机器人激光笔轨迹应落在地面虚线框内。	任意点起飞会存在不稳定情况，定点起飞情况稳定，其他基本满足要求
	从起飞到降落，巡线完成时间不得大于 150 秒，巡线时间越短越好。	多次测试飞行时间为 50 秒左右。
	发现线缆上异物(黄色凸起物),巡线机器人须在与异物距离不超过 30cm 的范围内用声或光提示。	发现线缆上异物灯光闪烁提示
发挥部分	拍摄所发现线缆异物上的条形码图片存储到 SD 卡，巡检结束后在显示装置上清晰显示，并能用手机识别此条形码内容。	OpenMV 实现拍摄条形码并存储到 SD 卡
	发现并拍摄杆塔 B 上的二维码图片存储到 SD 卡，巡检结束后在显示装置上清晰显示，并能用手机识别此二维码内容。	OpenMV 拍摄塔杆 B 上的二维码并存储到 SD 卡
	拍摄每张条形码、二维码图片存储的照片数不得超过 3 张。	可满足要求
	停机状况下，在巡线机器人某一旋翼轴下方悬挂一质量为 100g 的配重,然后巡线机器人在图 3 所示环形圆板上自主起飞,并在 1 米高度平稳悬停 10 秒以上，且摆动范围不得大于±25cm。	可满足要求

结果分析：经测试，系统基本可以满足基本要求和发挥部分，根据上述测试结果可以得出以下结论：

1、经过多次测试发现实际与理论差距太大，巡线机器人飞行不稳定时会影响 OpenMV 与机器人通信出现错误，导致机器人飞行姿态会处于不稳定状态。

2、通过微调 ADRC 参数使机器人起飞平衡。

3、通过调节加速度计可使机器人按要求飞行，减小误差。

综上所述，基本可以达到预期目标，说明我们的硬件和软件的设计基本趋于合理，达到要求。

五、参考文献

[1]韩京清. 从 PID 技术到"自抗扰控制"技术[J]. 控制工程, 2002, 9(3):13-18.

[2]夏长亮, 李正军, 杨荣,等. 基于自抗扰控制器的无刷直流电机控制系统[J]. 中国电机工程学报, 2005, 25(2):82-86.

[3]陈小龙, 唐强, 车军,等. 基于人工视觉的四旋翼飞行器室内定位与控制[J]. 兵工自动化, 2012(5):61-64.

[4]曹凯, 马贝, 王翔武. 四旋翼飞行器控制系统设计[J]. 计算机系统应用, 2018, 27(1):61-65.

[5]佚名. 四旋翼飞行器基于视觉的运动目标检测与跟踪[J]. 2014.