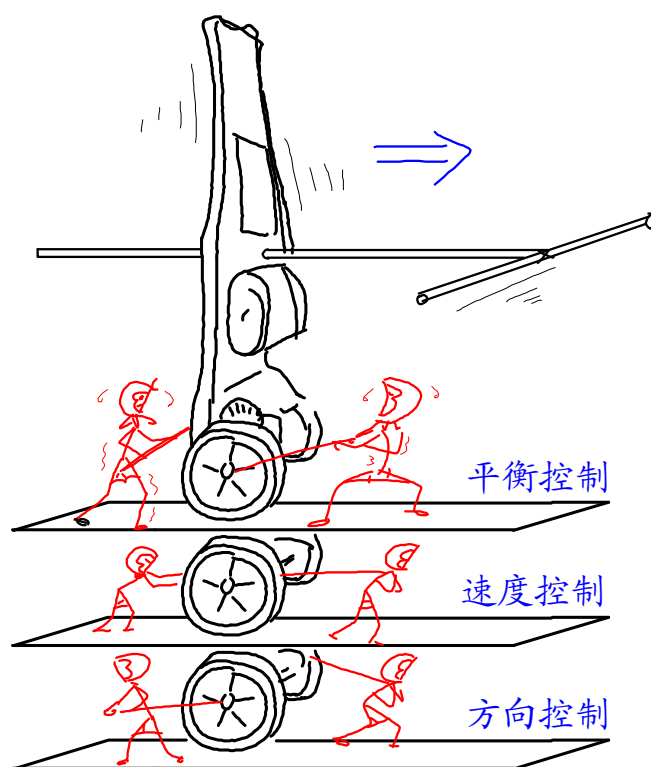


# 第七届全国大学生“飞思卡尔”杯智能汽车竞赛

## 电磁组直立行车参考设计方案

(版本 2.0)



竞赛秘书处

2012-3-1

# 目 录

目 录.....	2
图表索引.....	3
一、前言.....	7
二、原理篇.....	9
2.1 直立行走任务分解.....	9
2.2 车模直立控制.....	11
2.3 车模角度和角速度测量.....	19
2.3 车模速度控制.....	26
2.4 车模方向控制.....	33
2.6 车模直立行走控制算法总图.....	36
三、电路设计篇.....	37
3.1 整体电路框图.....	37
3.2 DSC介绍与单片机最小系统.....	39
3.3 倾角传感器电路.....	42
3.4 电机驱动电路.....	44
3.5 速度传感器电路.....	45
3.6 电磁线检测电路.....	46
3.7 角度计算电路.....	50
3.8 车模控制电路全图.....	55
四、机械设计篇.....	57
4.1 车模简化改装.....	57
4.2 传感器安装.....	59
4.3 注意事项.....	63
五、软件开发篇.....	64
5.1 软件功能与框架.....	65
5.2 DSC的硬件资源配置.....	68
5.3 主要算法及其实现.....	69
六、车模调试篇.....	<b>82</b>
6.1 调试参数.....	82
6.2 调试条件.....	85
6.3 桌面静态参数调试.....	91
6.4 现场动态参数调试.....	101
6.5 方案改进与车模整体水平提高.....	101
七、结束语.....	102
附录: .....	103

# 图表索引

## 第一章

图 1-1 电磁组车模直立运行模式.....	7
图 1-2 参考设计方案内容.....	8
图 1-3 车模制作路线图.....	9

## 第二章

图 2-1 车模控制任务分解.....	10
图 2-2 车模倾角会引起车速速度变化.....	10
图 2-3 三层控制之间相互配合.....	11
图 2-4 保持木棒直立的反馈控制.....	12
图 2-5 通过车轮运动保持车模直立.....	12
图 2-6 车模简化成倒立的单摆.....	13
图 2-7 普通单摆受力分析.....	13
图 2-8 不同阻尼力下的单摆运动.....	14
图 2-9 在车轮上的参照系中车模受力分析.....	14
图 2-10 车模控制两个系数作用.....	16
图 2-11 车模运动方程.....	16
图 2-12 加入比例微分反馈后的系统框图.....	17
图 2-13 电机在不同电压下的速度变化曲线.....	18
图 2-14 加速度传感器原理.....	19
图 2-15 MMA7260 三轴加速度传感器.....	20
图 2-16 车模运动引起加速度信号波动.....	21
图 2-17 实际测量MMA7260Z轴信号.....	21
图 2-18 车模运动引起加速度Z轴信号变化.....	22
图 2-19 角速度传感器及参考放大电路.....	22
图 2-20 角速度积分得到角度.....	23
图 2-21 角速度积分漂移现象.....	23
图 2-22 通过重力加速度来矫正陀螺仪的角度漂移.....	24
图 2-23 双加速度传感器获得车模角加速度.....	25
图 2-24 角度控制框图.....	25
图 2-25 电机速度检测.....	26
图 2-26 车模倾角给定.....	27
图 2-27 车模倾角控制速度中的正反馈.....	28
图 2-28 车模倾角控制分析.....	28
图 2-29 车模运动速度控制简化模型.....	29
图 2-30 增加微分控制后的系统.....	30
图 2-31 改进的微分控制.....	30
图 2-32 车模角度和速度控制框图.....	31
图 2-33 速度角度控制方案的改进.....	32

图 2-34 改进后的速度和角度控制方案.....	32
图 2-35 检测道路中心电磁线方式.....	33
图 2-36 通过电机驱动电压的差动控制控制车模方向.....	34
图 2-37 检测车模转动速度的陀螺仪.....	34
图 2-38 电感线圈的偏角影响感应电动势.....	35
图 2-39 车模方向控制算法.....	35
图 2-40 车模运动控制总框图.....	36

### 第三章

图 3-1 直立车模控制电路整体框图.....	39
图 3-2 56F8013 内部资源示意图.....	40
图 3-3 F8013 最小系统电路.....	41
图 3-4 F8013 最小系统电路实物.....	42
图 3-5 陀螺仪、加速度传感器电路.....	43
图 3-6 车模倾角传感器电路实物图.....	43
图 3-7 双电机驱动电路.....	44
图 3-8 单极性PWM、双极性PWM.....	45
图 3-9 两片 33886 组成的电机驱动电路.....	45
图 3-10 速度传感器电路.....	46
图 3-11 基于三极管的电磁信号放大检波电路.....	47
图 3-12 基于三极管的电磁放大检波电路实物图.....	48
图 3-13 使用R-R运放进行电磁信号放大检波.....	49
图 3-14 LMV358 放大检波输出波形.....	49
图 3-15 基于LMV358 放大检波电路实物图.....	50
图 3-16 双加速度测量角速度电路.....	51
图 3-17 双加速度计测量角度波形图.....	51
图 3-18 简化角速度电路.....	52
图 3-19 实测车模角速度信号波形.....	52
图 3-20 角度信号处理电路.....	53
图 3-21 不同角速度比例情况下输出波形.....	53
图 3-22 角度计算环节的传递函数.....	54
图 3-23 一个运算放大器实现角度计算.....	54
图 3-24 不同P1 阻值对应的输出波形.....	55
图 3-25 简化的角度和角速度处理电路.....	55
图 3-26 车模控制电路全图.....	56

### 第四章

图 4-1 完整的C型车模底盘.....	57
图 4-2 简化后的C型车模底盘.....	57
图 4-3 使用热熔胶固定电机支架与车模底盘.....	58
图 4-4 去掉后轮之后的车模底盘.....	58
图 4-5 电机引线转接板.....	59
图 4-6 使用复合胶水固定光电编码盘.....	59
图 4-7 固定好的光电码盘和光电检测管.....	60
图 4-8 电磁传感器支架.....	61

图 4-9 车模组装全图.....	64
<b>第五章</b>	
图 5-1 主程序框架.....	65
图 5-2 中断服务程序.....	66
图 5-3 任务中断时间波形.....	67
图 5-4 算法框图中与控制相关的软件函数.....	70
图 5-5 控制函数调用与参数传递关系.....	71
图 5-6 程序中变量命名规范.....	72
图 5-7 电机死区补偿.....	78
<b>第六章</b>	
图 6-1 调试车模参数复杂而关键.....	82
图 6-2 需要调整的参数和相关的单位.....	84
图 6-3 车模运动坐标定义.....	85
图 6-4 电源检查.....	86
图 6-5 单片机串口通信.....	86
图 6-6 PWM信号输出.....	87
图 6-7 采集电机光电码盘信号.....	87
图 6-8 陀螺仪、加速度传感器AD采集信号.....	88
图 6-9 车模静态参数调整桌面.....	88
图 6-10 车模动态参数调试场地.....	89
图 6-11 监控软件界面.....	89
图 6-12 无线遥控开关.....	90
图 6-13 无线通信模块进行参数监控.....	90
图 6-14 F8013 内部FLASH应用划分.....	91
图 6-15 需要整定的传感器参数.....	91
图 6-16 车模保持垂直静止.....	92
图 6-17 测量加速度传感器的极值.....	92
图 6-18 测量陀螺传感器比例因子，角度补偿回路断开.....	93
图 6-19 几种不同陀螺仪比例因子角度输出.....	94
图 6-20 车模控制参数.....	95
图 6-21 角度参数调整过程.....	96
图 6-22 速度参数调整过程.....	97
图 6-23 方向参数调整过程.....	98
图 6-24 角度补偿时间常数调整.....	99
图 6-25 Z轴附加信号分析.....	100
图 6-26 死区常数调整.....	100
<b>附录</b>	
图 7-1 参考设计方案视频截图.....	103
图 7-2 参数整定与调试指南.....	104
图 7-3 软件控制算法全图.....	106
图 7-4 参考方案电路全图.....	107



# 第一章、前言

为了提高全国大学生智能汽车竞赛创新性和趣味性，激发高校学生参与比赛的兴趣，提高学生的动手能力、创新能力和接受挑战能力，智能汽车竞赛组委会将电磁组比赛规定为车模直立行走，如图 1- 1所示。其它两个组别的车模行走方式保持不变。

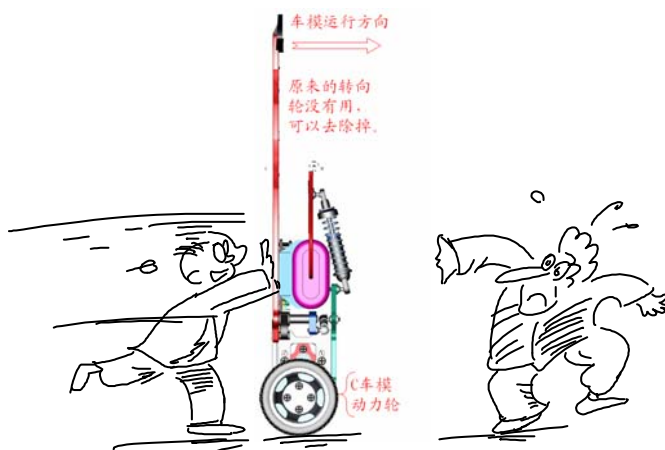


图 1-1 电磁组车模直立运行模式

车模直立行走比赛是要求仿照两轮自平衡电动车的行走模式，让车模以两个后轮驱动进行直立行走。近年来，两轮自平衡电动车以其行走灵活、便利、节能等特点得到了很大的发展。国内外有很多这方面的研究，也有相应的产品。在电磁组比赛中，利用了原来 C 型车模双后轮驱动的特点，实现两轮自平衡行走。相对于传统的四轮行走的车模竞赛模式，车模直立行走在硬件设计、控制软件开发以及现场调试等方面提出了更高的要求。为了能够帮助参赛学生尽快制作车模参加比赛，竞赛秘书处编写了 C 型车模直立行走的参考设计方案。参赛队员可以在此基础上，进一步改进硬件和软件方案，提高竞赛水平。

为了适应初学者，方案介绍过程中，尽可能减少公式推导，使用通俗的科学语言介绍控制原理和方法，给出 C 型车模制作过程中的核心环节。本文的主要内容安排如图 1- 2 所示。

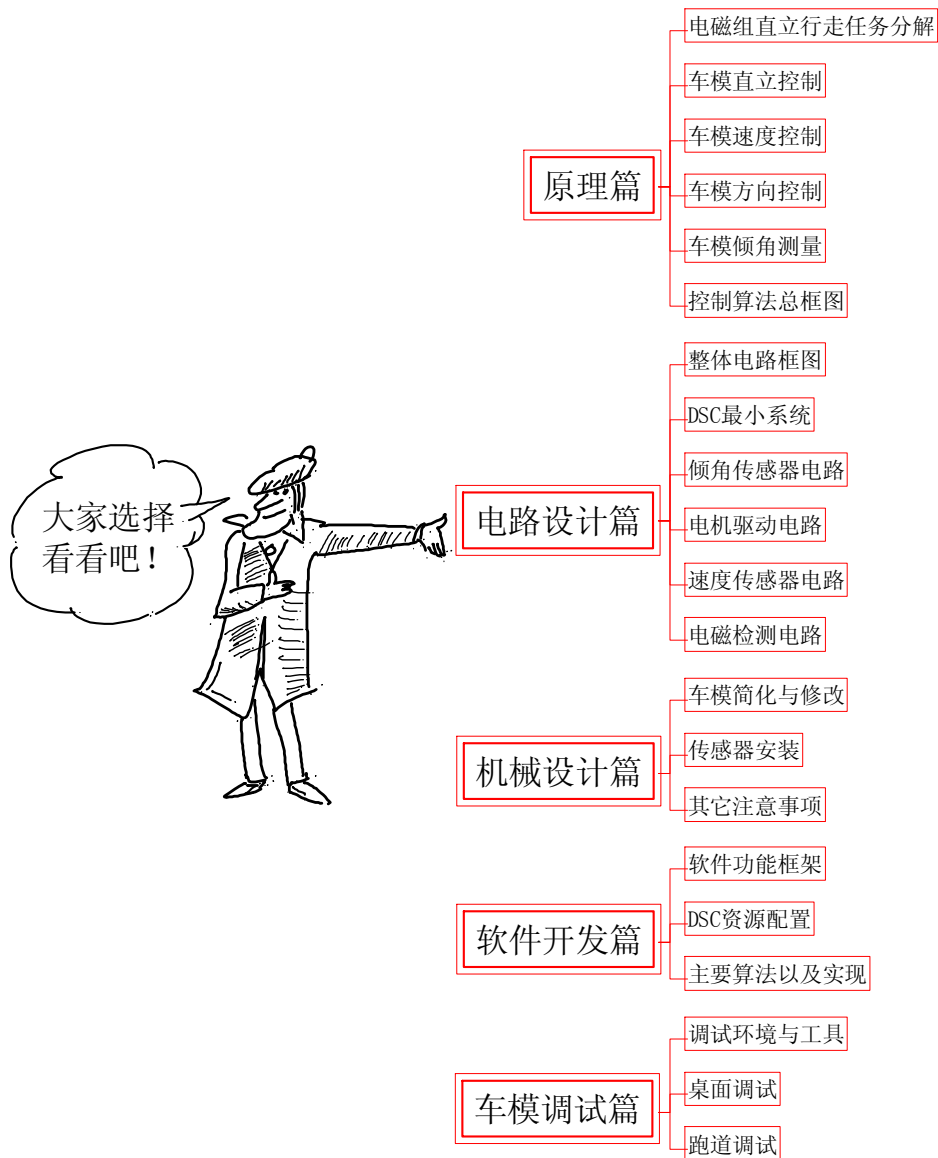


图 1-2 参考设计方案内容

参考设计方案经过了实际验证测试，测试运行视频请在竞赛网站上下载。

附录中给出了用于下载资料的竞赛网站地址以及可以下载的相关资料。

车模制作大致分为方案确定、车模制作、车模调试三个阶段，如图 1-3所示。由于电磁直立车模所涉及到的算法参数众多，所以调试阶段所占比重很大。车模调试不仅对于提高车模性能非常重要，同时也是理解车模控制原理、提高知识运用能力、培养现场实际操作技巧的重要阶段。



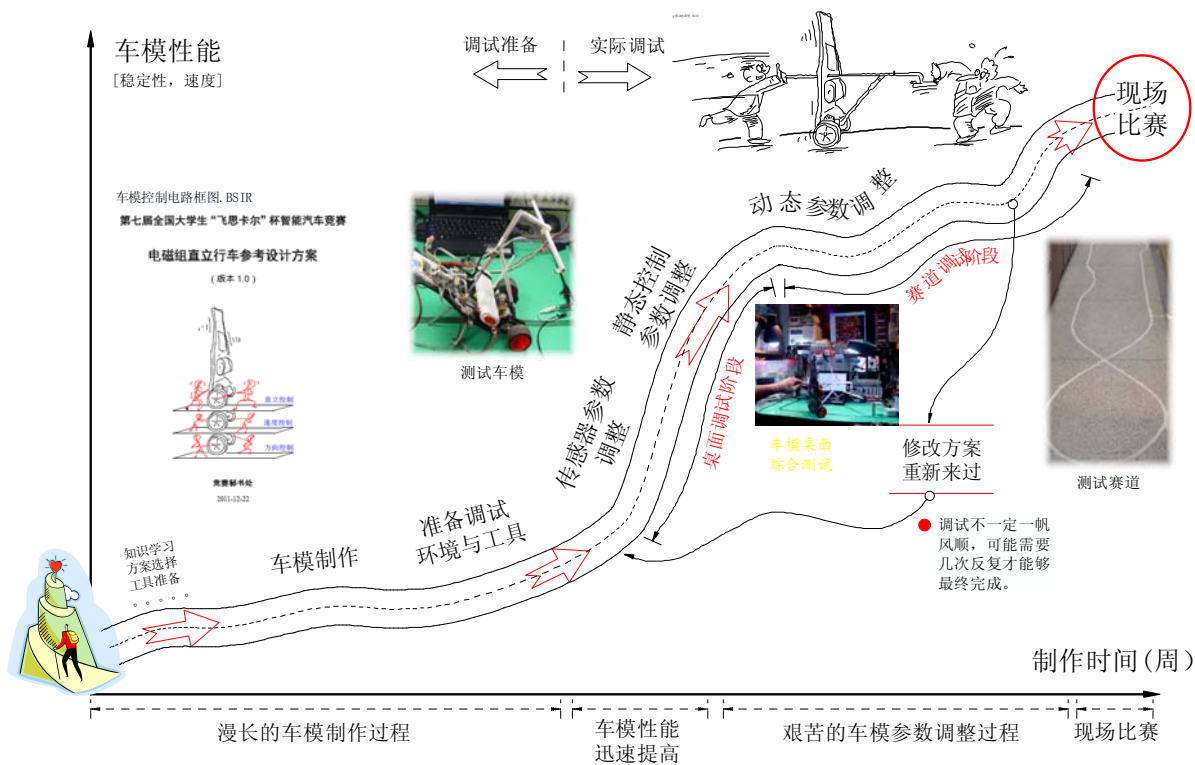


图 1-3 车模制作路线图

为了帮助同学进行车模调试，附录 4 给出了车模调试指南和相关的视频文件网站。可以通过这些资料尽快获得车模参数调整的直观感受。

## 第二章、原理篇

### 2.1 直立行走任务分解

电磁组比赛要求车模在直立的状态下以两个轮子着地沿着赛道进行比赛，相比四轮着地状态，车模控制任务更为复杂。为了能够方便找到解决问题的办法，首先将复杂的问题分解成简单的问题进行讨论。

根据比赛规则要求，维持车模直立也许可以设计出很多的方案，本参考方案假设维持车模直立、运行的动力都来自于车模的两个后车轮。后轮转动由两个直流电机驱动。因此从控制角度来看，车模作为一个控制对象，它的控制输入量是两个电机的转动速度。车模运动控制任务可以分解成以下三个基本控制任务，如图 2-1 所示：

- (1) 控制车模平衡：通过控制两个电机正反向运动保持车模直立平衡状态；

- (2) 控制车模速度：通过调节车模的倾角来实现车模速度控制，实际上最后还是演变成通过控制电机的转速来实现车轮速度的控制。
- (3) 控制车模方向：通过控制两个电机之间的转动差速实现车模转向控制。

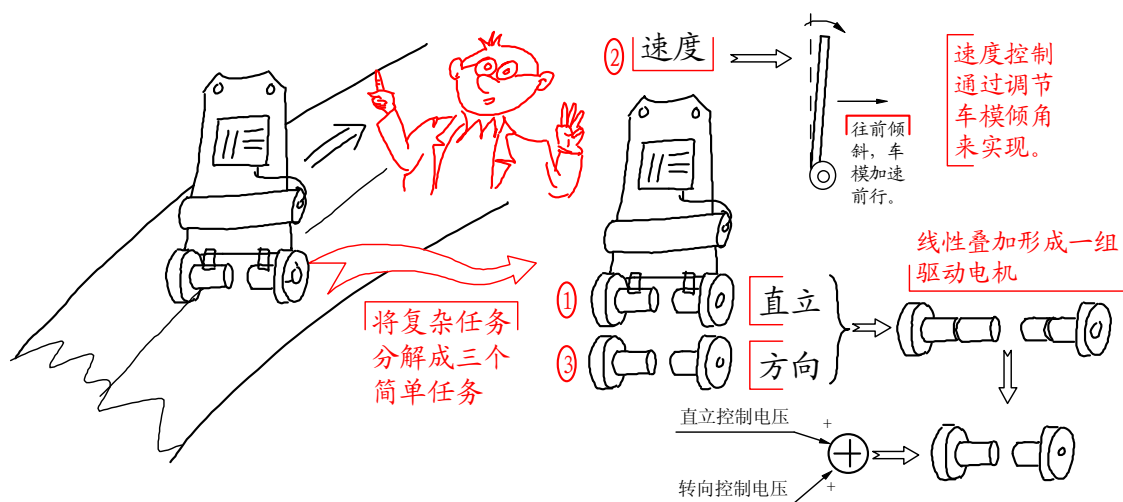


图 2-1 车模控制任务分解

车模直立和方向控制任务都是直接通过控制车模两个后轮驱动电机完成的。假设车模电机可以虚拟地拆解成两个不同功能的驱动电机，它们同轴相连，分别控制车模的直立平衡、左右方向。在实际控制中，是将控制车模直立和方向的控制信号叠加在一起加载电机上，只要电机处于线性状态就可以同时完成上面两个任务。

车模的速度是通过调节车模倾角来完成的。车模不同的倾角会引起车模的加减速，从而达到对于速度的控制。

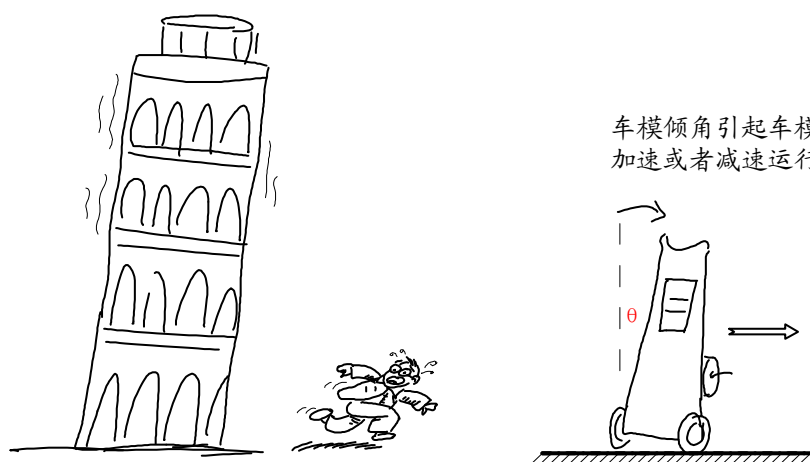


图 2-2 车模倾角会引起车速速度变化

三个分解后的任务各自独立进行控制。由于最终都是对同一个控制对象（车模的电机）进行控制，所以它们之间存在着耦合。为了方便分析，在分析其中之一时假设其它控制对象都已经达到稳定。比如在速度控制时，需要车模已经能够保持直立控制；在方向控制的时候，需要车模能够保持平衡和速度恒定；同样，在车模平衡控制时，也需要速度和方向控制也已经达到平稳。这三个任务中保持车模平衡是关键。由于车模同时受到三种控制的影响，从车模平衡控制的角度来看，其它两个控制就成为它的干扰。因此对车模速度、方向的控制应该尽量保持平滑，以减少对于平衡控制的干扰。以速度调节为例，需要通过改变车模平衡控制中车模倾角设定值，从而改变车模实际倾斜角度。为了避免影响车模平衡控制，这个车模倾角的改变需要非常缓慢的进行。这一点将会在后面速度控制中进行详细讨论。

三者之间的配合如图 2- 3所示。

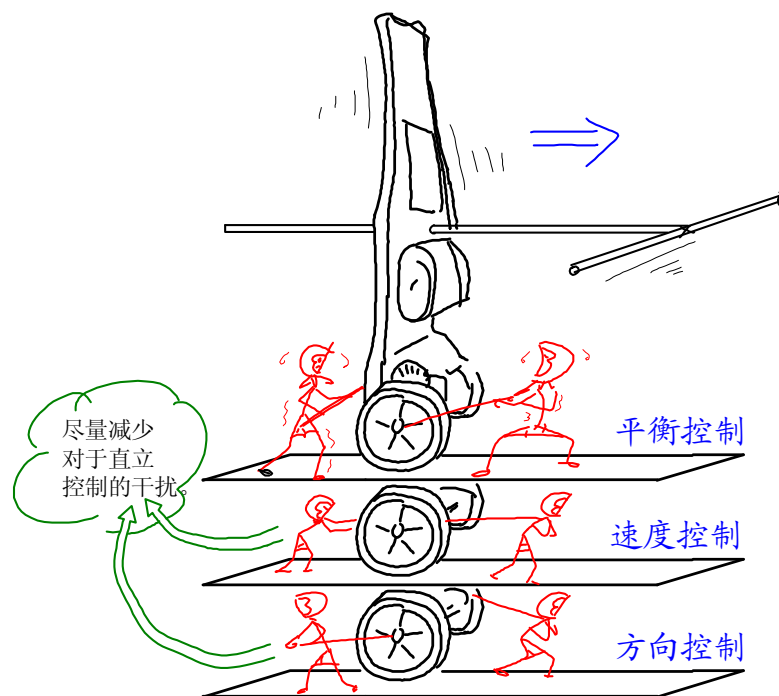


图 2- 3 三层控制之间相互配合

下面分别讨论车模任务分解的三个控制的实现方式。

## 2.2 车模平衡控制

控制车模平衡的直观经验来自于人们日常生活经验。一般的人通过简单练习就可以

让一个直木棒在手指尖上保持直立。这需要两个条件：一个是托着木棒的手掌可以移动；另一个是眼睛可以观察到木棒的倾斜角度和倾斜趋势（角速度）。通过手掌移动抵消木棒的倾斜角度和趋势，从而保持木棒的直立。这两个条件缺一不可，实际上就是控制中的负反馈机制，参见图 2-4。

世界上还没有任何一个天才杂技演员可以蒙着眼睛使得木棒在自己指尖上直立，因为没有了眼睛观察进行负反馈。

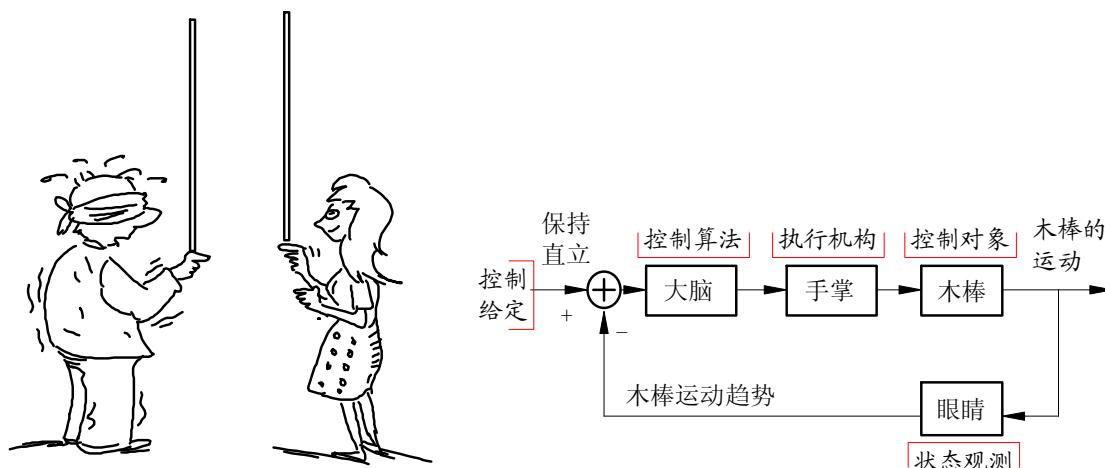


图 2-4 保持木棒直立的反馈控制

车模平衡控制也是通过负反馈来实现的，与上面保持木棒直立比较则相对简单。因为车模有两个轮子着地，车体只会在轮子滚动的方向上发生倾斜。控制轮子转动，抵消在一个维度上倾斜的趋势便可以保持车体平衡了。如图 2-5 所示。

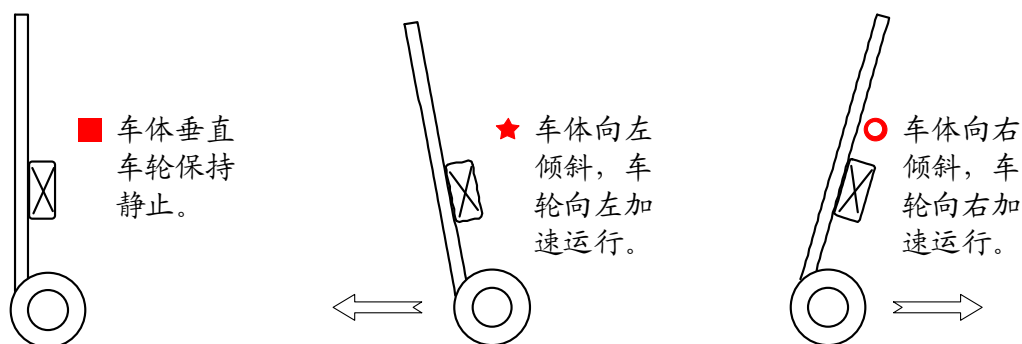


图 2-5 通过车轮运动保持车模平衡

那么车轮如何运行，才能够最终保持车体平衡稳定？为了回答这个问题，可以通过建立车模的运动学和动力学数学模型，设计反馈控制来保证车模的平衡。为了使得同学

们能够比较清楚理解其中的物理过程。下面通过对比单摆模型来说明保持车模平衡的控制规律。

重力场中使用细线悬挂着重物经过简化便形成理想化的单摆模型。直立着的车模可以看成放置在可以左右移动平台上的倒立着的单摆。如图 2-6 所示。

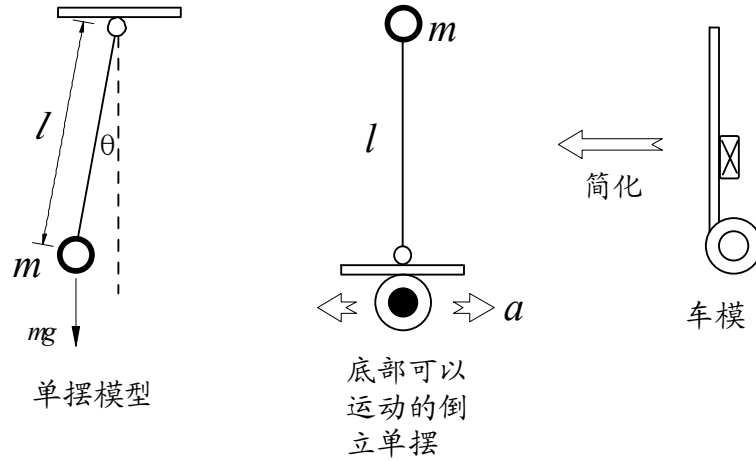


图 2-6 车模简化成倒立的单摆

对普通的单摆受力分析如图 2-7 所示。

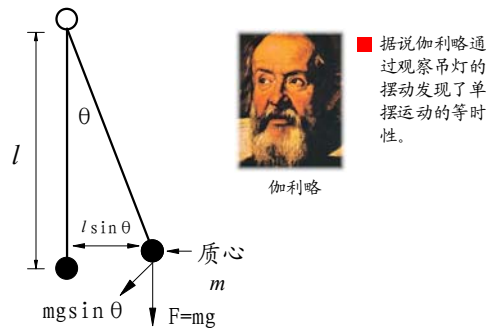


图 2-7 普通单摆受力分析

当物体离开垂直的平衡位置之后，便会受到重力与悬线的作用合力，驱动重物回复平衡位置。这个力称之为回复力，其大小为

$$F = -mg \sin \theta \approx -mg\theta$$

在偏移角度很小的情况下，回复力与偏移的角度之间大小成正比，方向相反。在此回复力作用下，单摆便进行周期运动。在空气中运动的单摆，由于受到空气的阻尼力，单摆最终会停止在垂直平衡位置。空气的阻尼力与单摆运动速度成正比，方向相反。阻尼力越大，单摆越会尽快在垂直位置稳定下来。图 2-8 显示出不同阻尼系数下，单摆的

运动曲线。

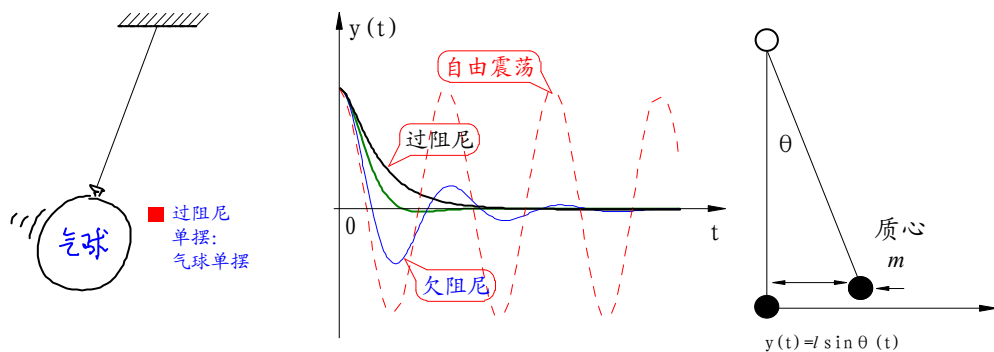


图 2-8 不同阻尼力下的单摆运动

总结单摆能够稳定在垂直位置的条件有两个：

- (1) 受到与位移（角度）相反的恢复力；
- (2) 受到与运动速度（角速度）相反的阻尼力。

如果没有阻尼力，单摆会在垂直位置左右摆动。阻尼力会使得单摆最终停止在垂直位置。阻尼力过小（欠阻尼）会使得单摆在平衡位置附件来回震荡。阻尼力过大（过阻尼）会使得单摆到达平衡位置时间加长。因而存在一个临界阻尼系数，使得单摆稳定在平衡位置的时间最短。

为什么倒立摆在垂直位置时，在受到外部扰动的情况下，无法保持稳定呢？分析倒立摆的受力，如图 2-9所示。

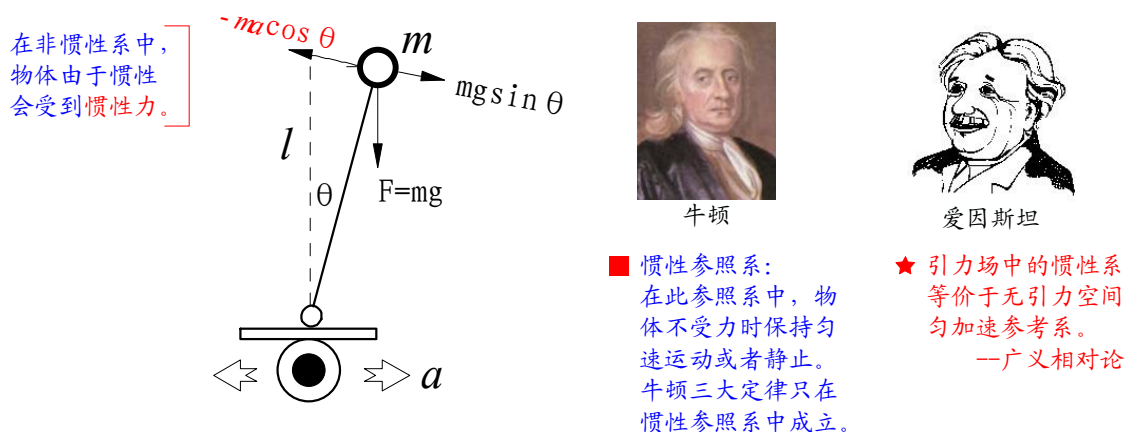


图 2-9 在车轮上的参照系中车模受力分析

倒立摆之所以不能象单摆一样可以稳定在垂直位置，就是因为它偏离平衡位置的时候，所受到的回复力与位移方向相同，而不是相反！因此，倒立摆便会加速偏离垂直

位置，直到倒下。

如何通过控制使得倒立摆能够像单摆一样，稳定在垂直位置呢？要达到这一目的，只有两个办法：一个是改变重力的方向；另一个是增加额外的受力，使得恢复力与位移方向相反才行。显然能够做到的只有第二种方法。

控制倒立摆底部车轮，使得它作加速运动。这样站在小车上（非惯性系，以车轮作为坐标原点）分析倒立摆受力，它就会受到额外的惯性力，该力与车轮的加速度方向相反，大小成正比。这样倒立摆所受到的回复力为：

$$F = mg \sin \theta - ma \cos \theta \approx mg\theta - mk_1\theta \quad (2-1)$$

式中，由于 $\theta$ 很小，所以进行了线性化。假设负反馈控制是车轮加速度 $a$ 与偏角 $\theta$ 成正比，比例为 $k_1$ 。如果比例 $k_1 > g$ ，（ $g$ 是重力加速度）那么回复力的方向便于位移方向相反了。

此外，为了使得倒立摆能够尽快地在垂直位置稳定下来，还需要增加阻尼力。虽然存在着空气和摩擦力等阻尼力，相对阻尼力比较小。因此需要另外增加控制阻尼力。增加的阻尼力与偏角的速度成正比，方向相反。因此式（2-1）可变为

$$F = mg\theta - mk_1\theta - mk_2\theta' \quad (2-2)$$

按照上面的控制方法，可把倒立摆模型变为单摆模型，能够稳定在垂直位置。因此，可得控制车轮加速度的控制算法

$$a = k_1\theta + k_2\theta' \quad (2-3)$$

式中， $\theta$ 为车模倾角； $\theta'$ 为角速度； $k_1$ 、 $k_2$ 均为比例系数；两项相加后作为车轮加速度的控制量。只要保证在 $k_1 > g$ 、 $k_2 > 0$ 条件下，可以使得车模像单摆一样维持在直立状态。其中有两个控制参数 $k_1, k_2$ ， $k_1$ 决定了车模是否能够稳定到垂直平衡位置，它必须大于重力加速度； $k_2$ 决定了车模回到垂直位置的阻尼系数，选取合适的阻尼系数可以保证车模尽快稳定在垂直位置。这两个系数的作用如图 2-10所示。

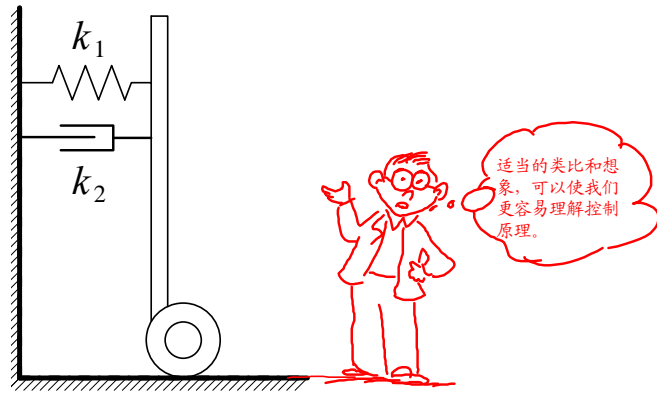
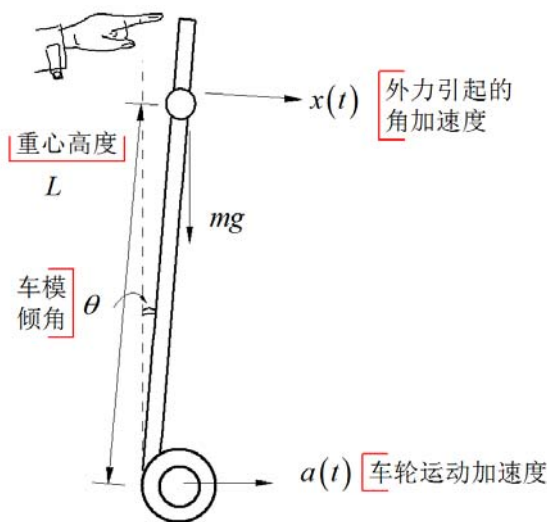


图 2-10 车模控制两个系数作用

在上面简单分析中，通过类比倒立摆得到了车模直立的控制方案。下面对倒立车模进行简单数学建模，然后建立速度的比例微分负反馈控制，根据基本控制理论讨论车模通过闭环控制保持稳定的条件。

假设倒立车模简化成高度为  $L$ ，质量为  $m$  的简单倒立摆，它放置在可以左右移动的车轮上。假设外力干扰引起车模产生角加速度  $x(t)$ 。沿着垂直于车模地盘方向进行受力分析，可以得到车模倾角与车轮运动加速度  $a(t)$  以及外力干扰加速度  $x(t)$  之间的运动方程。如图 2-11 所示。



车模运动方程

$$L \frac{d^2\theta(t)}{dt^2} = g \sin[\theta(t)] - a(t) \cos[\theta(t)] + Lx(t)$$

在角度  $\theta$  很小时，运动方程简化为：

$$L \frac{d^2\theta(t)}{dt^2} = g\theta(t) - a(t) + Lx(t)$$

车模静止时： $a(t) = 0$

$$L \frac{d^2\theta(t)}{dt^2} = g\theta(t) + Lx(t)$$

图 2-11 车模运动方程

对应车模静止时，系统输入输出的传递函数为：

$$H(s) = \frac{\Theta(s)}{X(s)} = \frac{1}{s^2 - \frac{g}{L}}, \text{ 此时系统具有两个极点 } s_p = \pm \sqrt{\frac{g}{L}}。 \text{ 一个极点位于 } s \text{ 平面的}$$



右半平面，因此车模不稳定。车模引入比例、微分反馈之后的系统如下图所示：

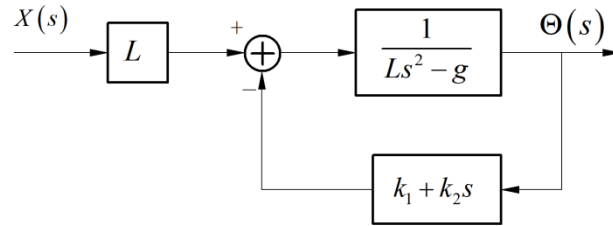


图 2-12 加入比例微分反馈后的系统框图

系统传递函数为： $H(s) = \frac{\Theta(s)}{X(s)} = \frac{1}{s^2 + \frac{k_2}{L}s + \frac{k_1 - g}{L}}$ 。此时两个系统极点位于：

$$s_p = \frac{-k_2 \pm \sqrt{k_2^2 - 4L(k_1 - g)}}{2L}$$

系统稳定需要两个极点都位于  $s$  平面的左半平面。要满足这一点，需要  $k_1 > g, k_2 > 0$ 。由此可以得出结论，当  $k_1 > g, k_2 > 0$  时，直立车模可以稳定。这与前面通过分析所得出的结论是一致的。

在角度反馈控制中，与角度成比例的控制量是称为比例控制；与角速度成比例的控制量称为微分控制（角速度是角度的微分）。因此上面系数  $k_1, k_2$  分别称为比例和微分控制参数。其中微分参数相当于阻尼力，可以有效抑制车模震荡。通过微分抑制控制震荡的思想在后面的速度和方向控制中也同样适用。

总结控制车模直立稳定的条件如下：

- (1) 能够精确测量车模倾角  $\theta$  的大小和角速度  $\theta'$  的大小；
- (2) 可以控制车轮的加速度。

第一个条件，即如何测量车模倾角和倾角速度  $\theta, \theta'$ ，参见下一小节“车模角度测量”。

如何确定控制参数  $k_1, k_2$  参见“调试篇”中的参数调节。下面先讨论第二个条件的实现，即车轮的加速度控制。

车模运行速度和加速度是通过控制车轮速度实现的，车轮通过车模两个后轮电机经由减速齿轮箱驱动，因此通过控制电机转速可以实现对车轮的运动控制。

电机的运动控制有三个作用：

- (1) 通过电机加速度控制实现车模平衡稳定。其中控制规律由上一节给出；
- (2) 通过电机速度控制，实现车模恒速运行和静止。虽然本届比赛规则中没有要

求车模速度恒定，也没有要求车模在比赛之前和冲过终点之后保持静止状态。通过速度控制，可以提高车模稳定性。在将来的比赛中，如果规则增加了静止要求，或者需要通过桥梁等障碍物时，速度控制将会发挥作用。

(3) 通过电机差速控制，可以实现车模方向控制。差速的控制方法参见后面“车模方向控制”。

电机运动控制是通过改变施加在其上的驱动电压大小实现的。对于电机的电磁模型、动力学模型以及车模的动力学模型进行分析和简化，可以将电机转速与施加在其上的电压之间的关系简化成如下的一阶惯性环节模型。施加在电机上一个阶跃电压  $Eu(t)$ ，电机的速度变化曲线为

$$\omega(t) = Ek_m \left( 1 - e^{-\frac{t}{T_1}} \right) u(t) \quad (2-4)$$

式中， $E$  为电压； $u(t)$  为单位阶跃函数； $T_1$  为惯性环节时间常数； $k_m$  为电机转速常数。对应不同的电压，电机的速度变化曲线如图 2-13 所示。

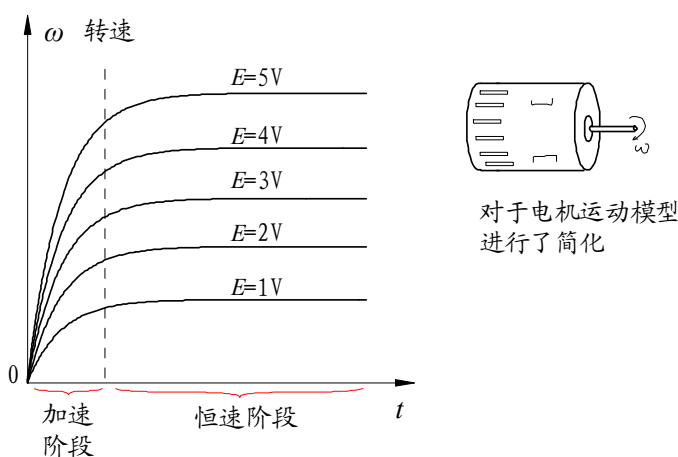


图 2-13 电机在不同电压下的速度变化线

由图 2-13 可以看出，电机运动明显分为两个阶段：第一个阶段是加速阶段；第二个阶段为恒速阶段。其中，在加速阶段，电机带动车模后轮进行加速运动，加速度近似和施加在电机上的电压成正比，加速阶段的时间长度取决于时间常数  $T_1$ 。该常数由电机转动惯量、减速齿轮箱减速比、车模的转动惯量决定，一般在十几到几百个毫秒。在恒速阶段，电机带动车模后轮进行恒速运行，运行速度与施加在电机上的电压成正比。

调整车模角度的控制周期很短，时间一般是几个毫秒，远小于时间常数  $T_1$ 。此时电

机基本上运行在加速阶段。由(2-3)计算所得到的加速度控制量  $a$  再乘以一个比例系数，即为施加在电机上的控制电压，这样便可以控制车模保持直立状态。

电机的加速度实际上是由通过电机的电流所产生的电磁力矩决定。考虑到电机电流的控制需要更高的速度，所以在此就简化电机的控制方案。

## 2.3 车模角度和角速度测量

在上一节中介绍了控制车模直立控制算法，通过测量车模的倾角和倾角速度控制车模车轮的加速度来消除车模的倾角。因此，车模倾角以及倾角速度的测量成为控制车模直立的关键。测量车模倾角和倾角速度可以通过安装在车模上的加速度传感器和陀螺仪实现。

### (1) 加速度传感器

加速度传感器可以测量由地球引力作用或者物体运动所产生的加速度。竞赛规则规定如果车模使用加速度传感器必须使用飞思卡尔公司产生的加速度传感器。该系列的传感器采用了半导体表面微机械加工和集成电路技术，传感器体积小，重量轻。它的基本原理如图 2-14 所示。

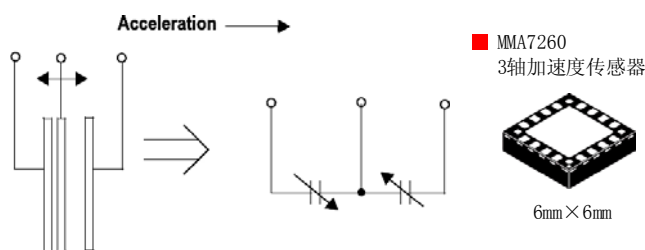


图 2-14 加速度传感器原理

通过微机械加工技术在硅片上加工形成了一个机械悬臂。它与相邻的电极形成了两个电容。由于加速度使得机械悬臂与两个电极之间的距离发生变化，从而改变了两个电容的参数。通过集成的开关电容放大电路量测电容参数的变化，形成了与加速度成正比的电压输出。MMA7260 是一款三轴低g半导体加速度计，可以同时输出三个方向上的加速度模拟信号，如图 2-15 所示。

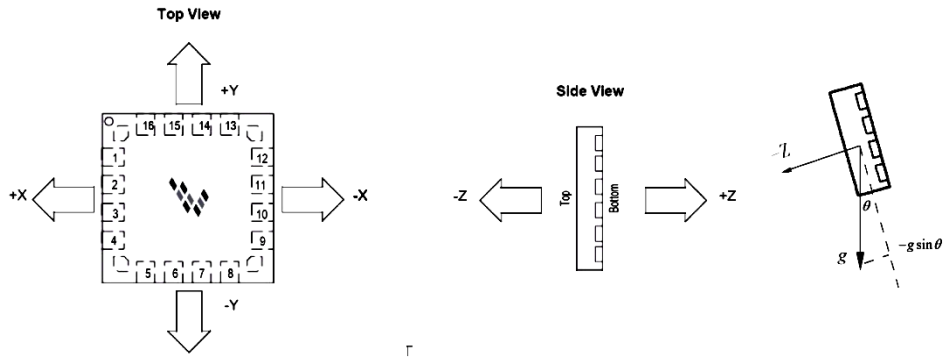


图 2- 15 MMA7260 三轴加速度传感器

通过设置可以使得 MMA7260 各轴信号最大输出灵敏度为 800mV/g，这个信号无需要在进行放大，直接可以送到单片机进行 AD 转换。实际上，飞思卡尔公司还有更多系列的低  $g$  值的加速度传感器，特别是具有数字接口的传感器可以方便单片机接口设计。

只需要测量其中一个方向上的加速度值，就可以计算出车模倾角，比如使用 Z 轴方向上的加速度信号。车模直立时，固定加速度器在 Z 轴水平方向，此时输出信号为零偏电压信号。当车模发生倾斜时，重力加速度  $g$  便会在 Z 轴方向形成加速度分量，从而引起该轴输出电压变化。变化的规律为

$$\Delta u = kg \sin \theta \approx kg\theta$$

式中， $g$  为重力加速度； $\theta$  为车模倾角； $k$  为加速度传感器灵敏度系数。当倾角  $\theta$  比较小的时候，输出电压的变化可以近似与倾角成正比。

似乎只需要加速度就可以获得车模的倾角，再对此信号进行微分便可以获得倾角速度。但在实际车模运行过程中，由于车模本身的摆动所产生的加速度会产生很大的干扰信号，它叠加在上述测量信号上使得输出信号无法准确反映车模的倾角，如图 2- 16 所示。

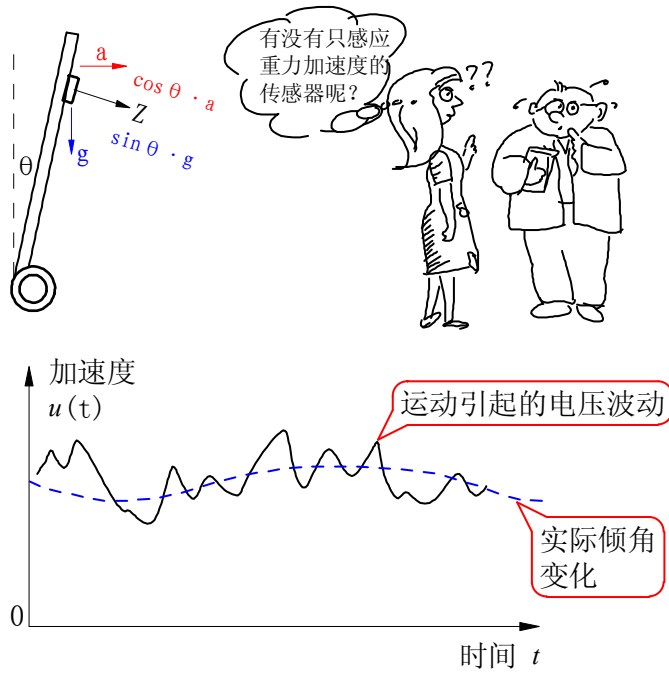


图 2-16 车模运动引起加速度信号波动

下图是实际测量安装在车模上 MMA7260 的 Z 轴信号。车模倾角在两个角度位置过渡，看到除了角度变化信号之外，还存在由于运动引起的电压波动，这个电压波动随着车模运动速度增加会变得很大。

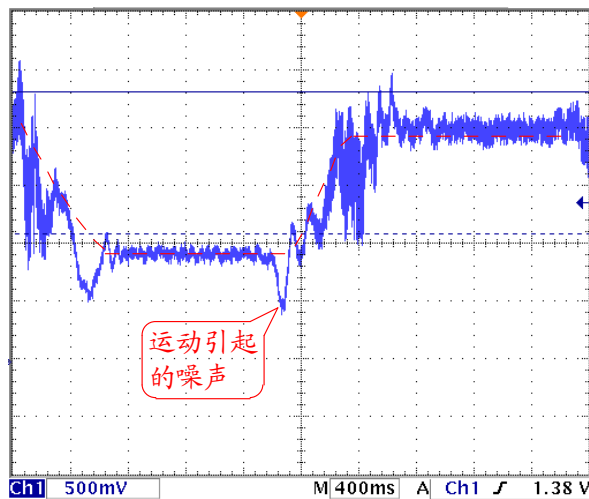


图 2-17 实际测量 MMA7260Z 轴信号

下面简单分析运动所产生的干扰信号。加速度传感器安装在车模上，距离车轴高度为  $h$ 。车模转动具有角加速度  $\theta'$ ，运动加速度  $\alpha$ 。那么在加速度传感器 Z 轴上出现由于车模运动引起的加速度为  $h\theta' + \alpha$ ，如图 2-18 所示。为了减少运动引起的干扰，加速度

传感器安装的高度越低越好，但是无法彻底消除车模运动的影响。

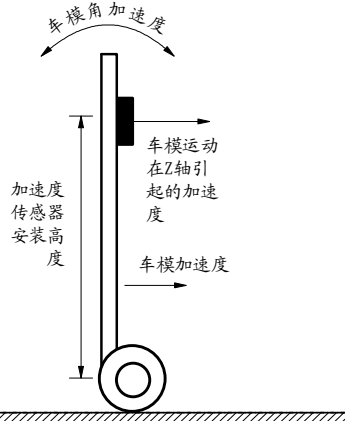


图 2- 18 车模运动引起加速度 Z 轴信号变化

车模运动产生的加速度使得输出电压在实际倾角电压附近波动。这些波动噪声可以通过数据平滑滤波将其滤除。但是平滑滤波一方面会使得信号无法实时反映车模倾角变化，从而减缓对于车模车轮控制。另一方面也会将车模角速度变化信息滤掉。上述两方面的滤波效果使得车模无法保持平衡。因此对于车模直立控制所需要的倾角信息需要通过另外一种器件获得，那就是角速度传感器-陀螺仪。

## (2) 角速度传感器-陀螺仪

陀螺仪可以用来测量物体的旋转角速度。竞赛允许选用村田公司出品的ENC-03 系列的加速度传感器。它利用了旋转坐标系中的物体会受到科里奥利力的原理，在器件中利用压电陶瓷做成振动单元。当旋转器件时会改变振动频率从而反映出物体旋转的角速度。ENC-03 角速度传感器以及相关参考放大电路如图 2- 19所示。

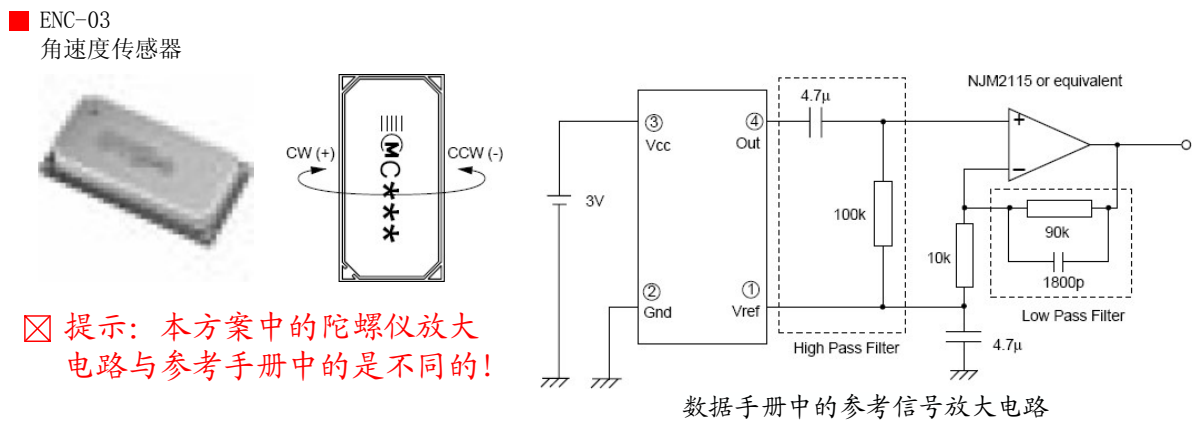


图 2- 19 角速度传感器及参考放大电路

在车模上安装陀螺仪，可以测量车模倾斜角速度，将角速度信号进行积分便可以得到车模的倾角。如图 2- 20所示。

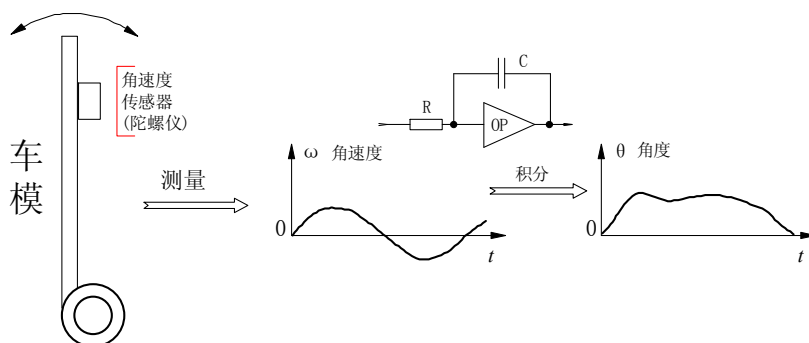


图 2- 20 角速度积分得到角度

由于陀螺仪输出的是车模的角速度，不会受到车体运动的影响，因此该信号中噪声很小。车模的角度又是通过对角速度积分而得，这可进一步平滑信号，从而使得角度信号更加稳定。因此车模控制所需要的角度和角速度可以使用陀螺仪所得到的信号。

由于从陀螺仪角速度获得角度信息，需要经过积分运算。如果角速度信号存在微小的偏差和漂移，经过积分运算之后，变化形成积累误差。这个误差会随着时间延长逐步增加，最终导致电路饱和，无法形成正确的角度信号，如图 2- 20所示。

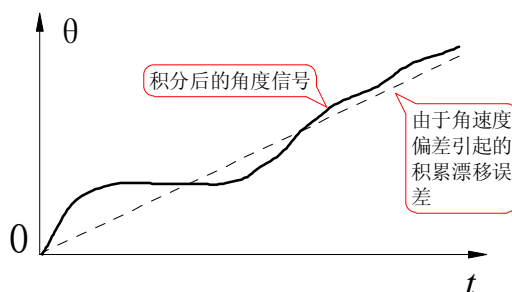


图 2- 21 角速度积分漂移现象

如何消除这个累积误差呢？

一种简单的方法就是通过上面的加速度传感器获得的角度信息对此进行校正。通过对比积分所得到的角度与重力加速度所得到的角度，使用它们之间的偏差改变陀螺仪的输出，从而积分的角度逐步跟踪到加速度传感器所得到的角度。如图 2- 22所示。

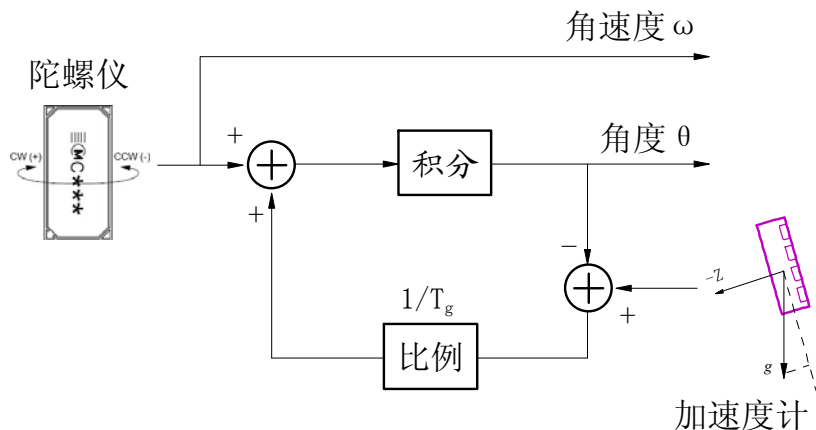


图 2-22 通过重力加速度来矫正陀螺仪的角度漂移

在上述方案中，利用加速度计所获得的角度信息  $\theta_g$  与陀螺仪积分后的角度  $\theta$  进行比较，将比较的误差信号经过比例  $1/T_g$  放大之后与陀螺仪输出的角速度信号叠加之后再行积分。对于加速度计给定的角度  $\theta_g$ ，经过比例、积分环节之后产生的角度  $\theta$  必然最终等于  $\theta_g$ 。

写出图 2-22 系统的传递函数，可以看出输出角度包含了两个一阶惯性环节。其中第一项是重力加速度 Z 轴对应的角度，第二项是陀螺仪加速度经过惯性环节的数值。系统不再存在纯积分环节，所以陀螺仪的角速度微小的偏差不会形成积累误差。其中参数  $T_g$  决定了这两个惯性环节的时间常数。 $T_g$  越大，角度输出跟踪 Z 轴输出越慢，但可以有效抑制重力加速度计上的噪声。 $T_g$  过大，就会放大陀螺仪输出误差。这个参数需要根据实际车模运行效果进行调整，详见后面“车模调试篇”中的方法。

为了避免输出角度  $\theta$  跟踪时间过长，可以采取以下两个方面的措施：

- (1) 仔细调整陀螺仪的放大电路，使得它的零点偏置尽量接近于设定值，并且稳定。
- (2) 在控制电路和程序运行的开始，尽量保持车模处于直立状态，这样一开始就使得输出角度  $\theta$  与  $\theta_g$  相等。此后，加速度计的输出只是消除积分的偏移，输出角度不会出现很大的偏差。

使用加速度计来矫正陀螺仪的积分漂移只是其中一种方法。还可以通过测量车模的运行速度和加速度来矫正陀螺仪的积分漂移，这样就可以省略加速度器件。这种控制方



法在后面速度控制中给出。

### (3) 双加速度传感器获得角度和角速度

在图 2-18 所显示的加速度传感器 Z 轴信号除了由于重力加速度引起的输出之外，还包括有车模的角加速度和移动加速度产生的信息。在车模现有的参数基础上，这些信号在幅值、频率等方面没有太大差异，它们叠加在一起无法将它们分开。如果在车模上另外再增加一个加速度传感器，两个加速度传感器安装的高度不同，那么就可以通过这两个信号的差值求出车模的角加速度。如图 2-23 所示。

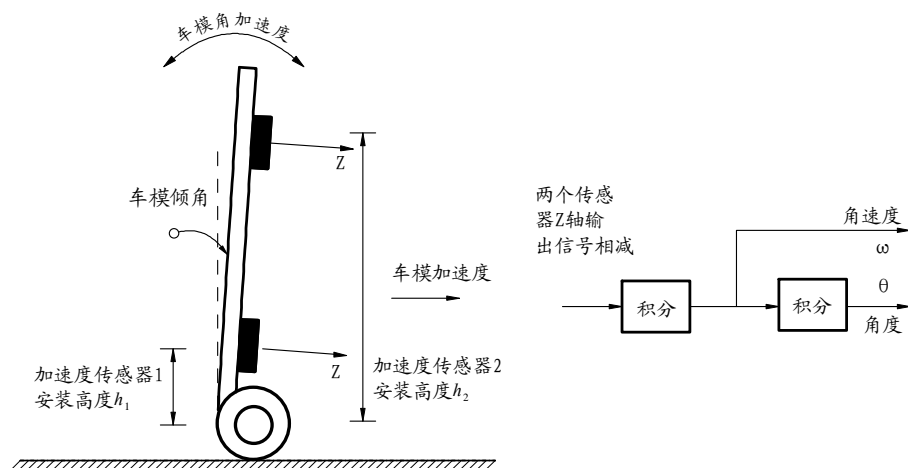


图 2-23 双加速度传感器获得车模角加速度

通过上下两个加速度传感器输出信号相减，便可以得到车模倾角加速度。对于这个信号进行两次积分，便可以获得的获得车模倾角的角速度和角度。对于积分所可能带来的积分漂移问题仍然可以采用上面的重力加速度计补偿的方法进行消除。

根据前两节介绍车模角度控制和角度测量方法，可以得到如下车模角度控制方案框图。

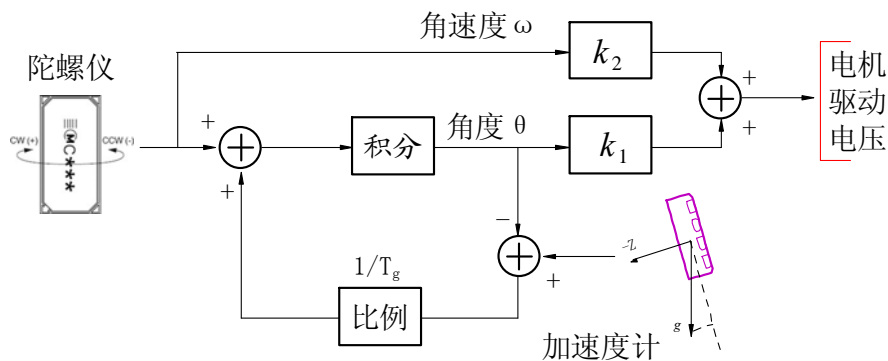


图 2-24 角度控制框图

该方案中采用重力加速度计和陀螺仪通过角度互补融合方式获取车模倾角和角速度，通过两个比例常数加权后，控制电机驱动电压，使得车模产生相应的加速度，维持车模的直立。

在此方案中，需要保证重力加速度传感器安装 Z 轴与车模直立中轴线严格垂直。如果出现角度偏差，上述控制实际结果是车模与地面不是严格垂直，而是存在一个对应的倾角。在重力的作用下，车模会朝着一个方面加速前进。为了保持车模的静止或者匀速运动需要消除这个安装误差。在实际车模制作过程中需要进行机械调整和软件参数设置。另外需要通过软件中的速度控制来实现速度的稳定性。

在车模角度控制中出现的车模倾角偏差，使得车模在倾斜的方向上产生加速。这个结果可以用来进行车模的速度控制。下一节将利用这个原理来调节车模的速度。

## 2.3 车模速度控制

对于直立车模速度的控制相对于普通车模的速度控制则比较复杂。由于在速度控制过程中需要始终保持车模的平衡，因此车模速度控制不能够直接通过改变电机转速来实现。下面先分析一下引起车模速度变化的原因。

假设车模在上面直立控制调节下已经能够保持平衡了，但是由于安装误差，传感器实际测量的角度与车模角度有偏差，因此车模实际不是保持与地面垂直，而是存在一个倾角。在重力的作用下，车模就会朝倾斜的方向加速前进。图 2-2 显示了这个关系。控制速度只要通过控制车模的倾角就可以实现了。具体实现需要解决三个问题：

- (1) 如何测量车模速度？
- (2) 如何通过车模直立控制实现车模倾角的改变？
- (3) 如何根据速度误差控制车模倾角？

第一个问题可以通过安装在电机输出轴上的光码盘来测量得到车模的车轮速度。如图 2-25 所示。

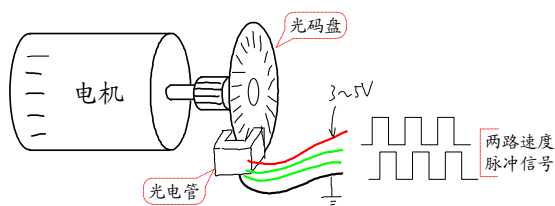


图 2-25 电机速度检测

利用控制单片机的计数器测量在固定时间间隔内速度脉冲信号的个数可以反映电机的转速。

第二个问题可以通过角度控制给定值来解决。给定车模直立控制的设定值，在角度控制调节下，车模将会自动维持在一个角度。通过前面车模直立控制算法可以知道，车模倾角最终是跟踪重力加速度Z轴的角度。因此车模的倾角给定值与重力加速度Z轴角度相减，便可以最终决定车模的倾角。如图 2- 26所示。

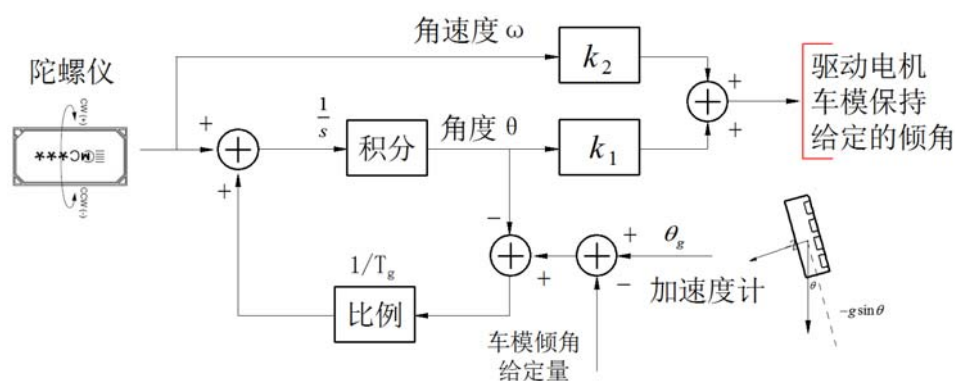


图 2- 26 车模倾角给定

分析图 2- 22中车模角度控制规律可以看出车模倾角设定量与车模倾角变化之间大体呈现一个一阶惯性环节  $\frac{1}{T_z s + 1}$ ，这个结论在下面进行速度控制时需要应用。

第三个问题分析起来相对比较困难，远比直观进行速度负反馈分析复杂。首先对一个简单例子进行分析。假设车模开始保持静止，然后增加给定速度，为此需要车模往前倾斜以便获得加速度。在车模直立控制下，为了能够有一个往前的倾斜角度，车轮需要往后运动，这样会引起车轮速度下降（因为车轮往负方向运动了）。由于负反馈，使得车模往前倾角需要更大。如此循环，车模很快就会倾倒。原本利用负反馈进行速度控制反而成了“正”反馈。如图 2- 27所示。

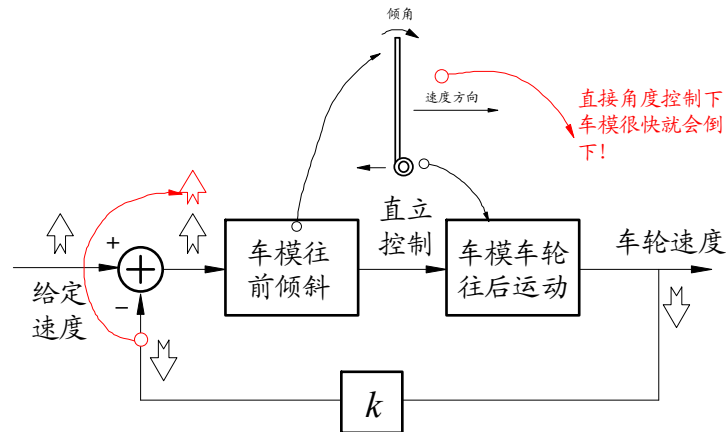


图 2-27 车模倾角控制速度中的正反馈

为什么负反馈控制在这儿失灵了呢？原来在直立控制下的车模速度与车模倾角之间传递函数具有非最小相位特性（在此省略了分析），在反馈控制下容易造成系统的不稳定性。但根据实际经验，是可以通过速度控制进行车模倾角控制的。为了得到合理的经验控制结论，需要将前面速度负反馈控制问题进行简化建模分析一下。所做简化假设如下：

(1) 车模角度在直立控制下简化成一个一阶过渡过程  $\frac{1}{T_z s + 1}$ 。这一点参见前面第

二个问题的最后结论。

(2) 车模倾角是由车轮运动产生，因此车轮速度是倾角变量求导再乘以车模长度。

(3) 忽略车模倾角引起车轮加速度。由于这个讨论仅对控制一开始短暂的过渡阶段而言，此时由于车模倾角比较小，引起速度变化很小。

(4) 系统只进行速度比例负反馈。

简化模型如图 2-28 所示。

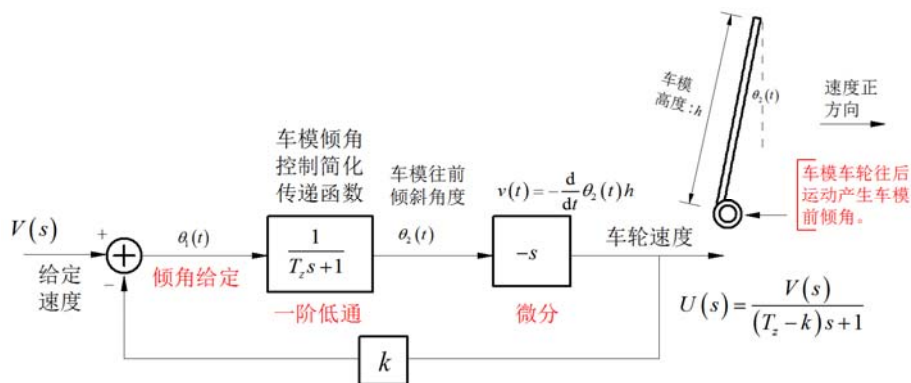


图 2-28 车模倾角控制分析

分析简化系统的传递函数，它具有一个一阶极点  $\frac{-1}{T_z - k}$ 。如果要保证系统稳定，需要满足  $T_z > k$ 。这就要求车模倾角的调整时间常数  $T_z$  要大，调整速度缓慢和平滑，同时速度的反馈比例  $k$  要小。这一结论与实际经验是相符合的。

以上两点在程序实现方法可以通过改变控制周期，减小控制参数，信号进行平滑滤波等方式来实现。具体可以参见第五部分软件代码。

在实现上述两点之后，再分析车模运动速度的稳定性。为了简化分析，首先忽略了由于倾角控制引起的车模速度的变化。车模的速度最终由倾角所产生的加速度决定，因此将倾角的进行积分便可以得到车模的运动速度。此时车模速度控制框图如图 2- 29所示。

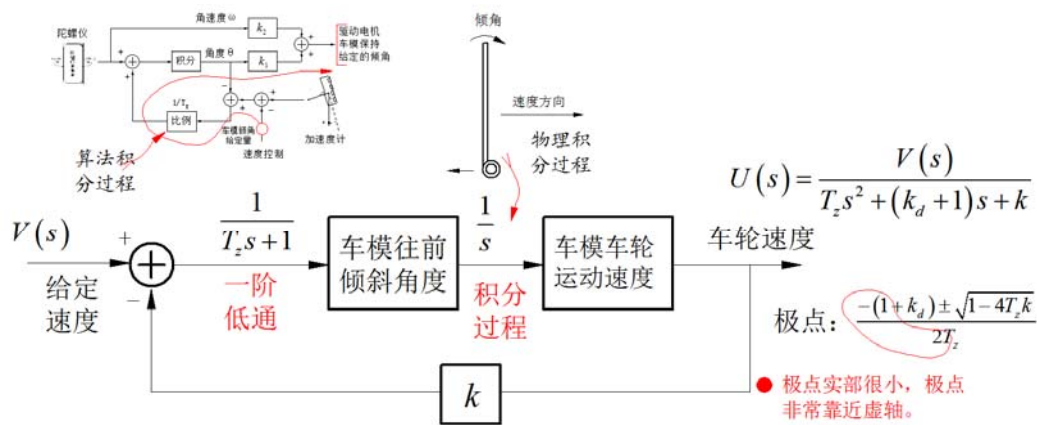


图 2- 29 车模运动速度控制简化模型

前面分析可知为了保证系统稳定，往往取的车模倾角控制时间常数  $T_z$  很大。这样便会引起系统产生两个共轭极点，而且极点的实部变得很小，使得系统的速度控制会产生震荡现象。这个现象在实际参数整定的时候可以观察到。那么如何消除速度控制过程中的震荡呢？

要解决控制震荡问题，在前面的车模角度控制中已经有了经验，那就是在控制反馈中增加速度微分控制。如图 2- 30所示

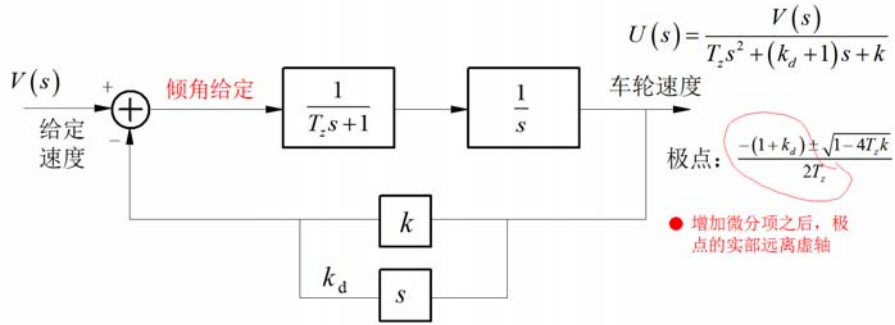


图 2-30 增加微分控制后的系统

此时系统传递函数的极点实部变大，加快了系统的收敛速度。

由于车轮的速度反馈信号中往往存在着噪声，对速度进行微分运算会进一步加大噪声的影响。为此需要对上面控制方法进行改进。原系统中倾角调整过程时间常数往往很大，因此可以将该系统近似为一个积分环节。将原来的微分环节和这个积分环节合并，形成一个比例控制环节。这样可以保持系统控制传递函数不变，同时避免了微分计算。具体改进如图 2-31 所示。

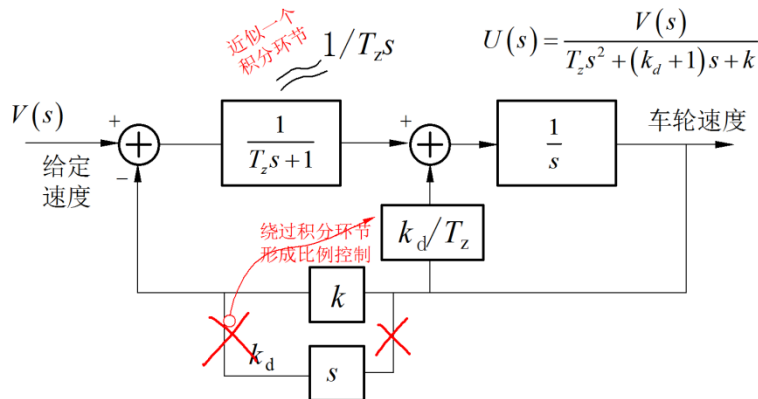


图 2-31 改进的微分控制

上述改进可以避免对于速度反馈信号的分，但在控制反馈中，只是使用反馈信号的比例和微分，没有利误差积分，所以最终这个速度控制是有残差的控制。但是直接引入误差积分控制环节，会增加系统的复杂度，为此就不再增加积分控制，而是通过与角度控制相结合后在进行改进。

结合图 2- 26给出的车模角度控制与角度给定框图，将图 2- 31引入后形成车模角度和速度控制框图。如图 2- 32所示：

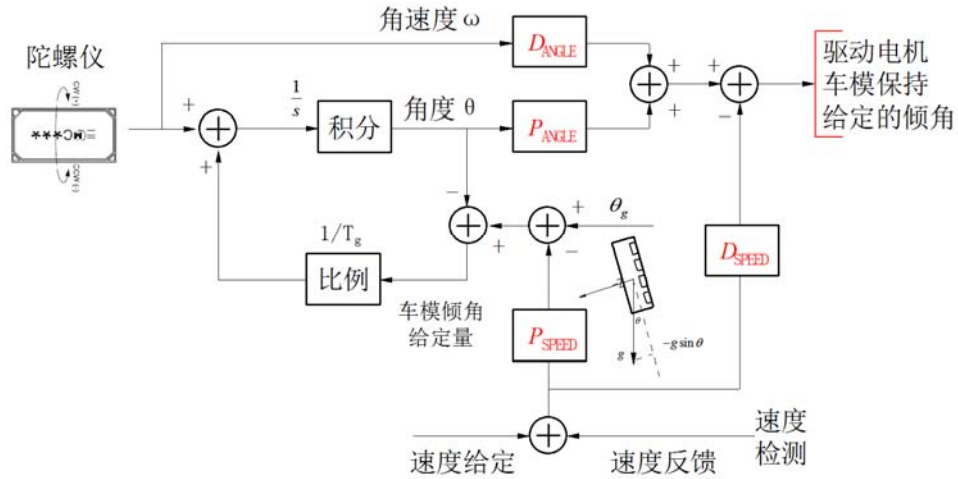


图 2-32 车模角度和速度控制框图

在上图中，角度控制需要两个控制参数，分别是比例控制参数  $P_{ANGLE}$  和微分控制参数  $D_{ANGLE}$ 。速度控制同样也有两个参数，分别是比例控制参数  $P_{SPEED}$  和微分控制参数  $D_{SPEED}$ 。在这两个控制中都使用了微分控制，目的是增加车模的角度和速度的稳定性，防止控制超调。这个经验在后面车模方向控制中仍然适用。

下面对于图 2-30 的控制方案再进两点的改进。第一点，图 2-32 给出的控制方案对于速度误差没有进行积分控制，这样最终速度不会严格等于设定值。比如要求车模在原地停止，速度为 0。但是由于采用的是比例控制，如果此时陀螺仪有漂移，或者加速度传感器安装有误差，最终车模倾角不会最终调整到 0，车模会朝着倾斜的方向恒速运行下去。注意此时车模不会像没有速度控制那样加速运行了，但是速度不会最终为 0。为了消除这个误差，可以将车模倾角设定量直接积分补偿在角度控制输出中，这样就会彻底消除速度控制误差。第二点，由于加入了速度控制，它可以补偿陀螺仪和重力加速度的漂移和误差。所以此时重力加速度传感器实际上没有必要了。因此可以将该传感器去掉，但是比例负反馈回路仍然保留，它用于克服陀螺仪的积分漂移。综合上两点的改进，图 2-32 可以进行如下图所示的改进。

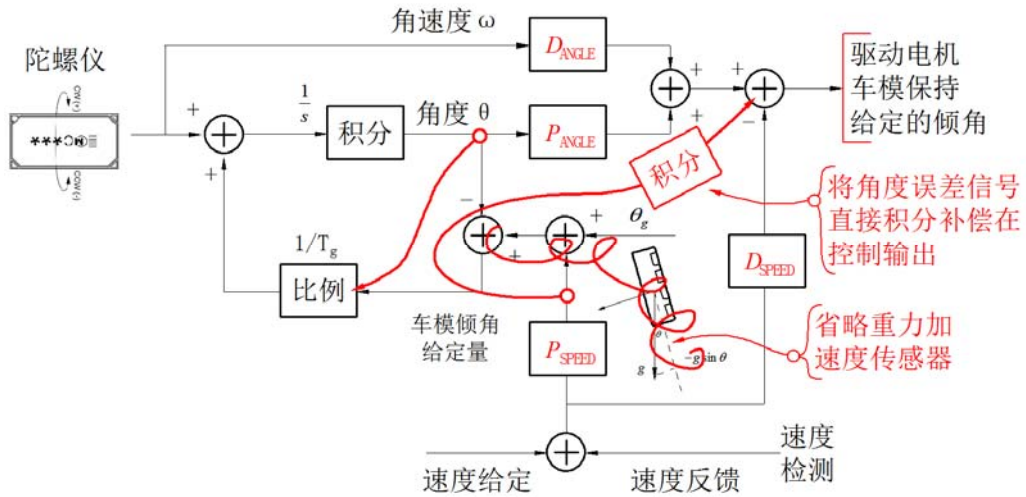


图 2-33 速度角度控制方案的改进

改进后的角度和速度控制方案如下图所示：

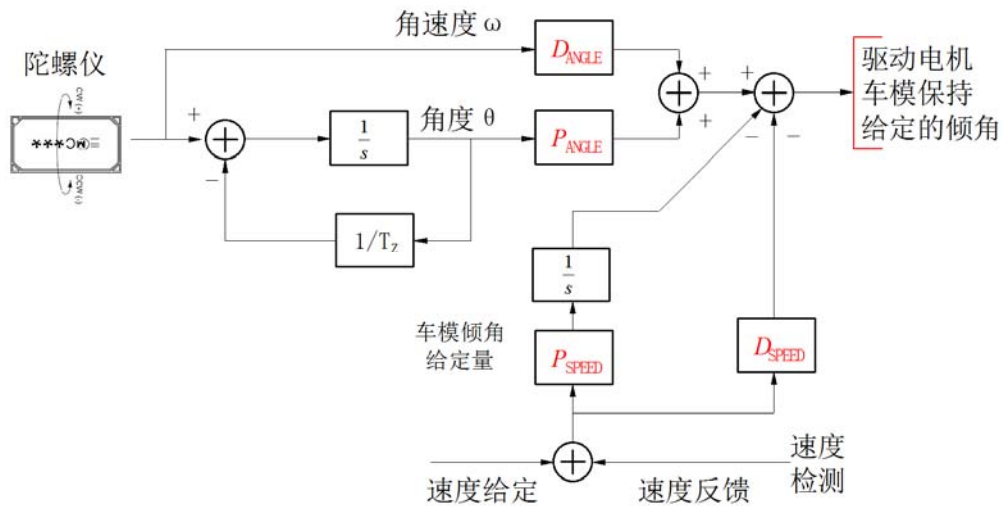


图 2-34 改进后的速度和角度控制方案

图 2-34 中没有了重力加速度传感器，速度回路控制可以补偿传感器的误差。此时车模在控制启动的时候，需要保持车模的垂直状态。此时陀螺仪的积分角度也初始化为 0。当然如果电路中已经包括了重力加速度传感器，也可以保留这部分，从而提高车模的稳定性。在后面的最终给定的控制方案中，保留了这部分的控制回路。

在图 2-34 中速度控制误差经过积分和比例直接叠加在电机控制量上。速度控制在开始讨论的时候是通过调整车模倾角来实现，通过上面进行的简化和改进，最后在图 2-34 中，已经演变成对于电机的直接控制了。请注意两点：第一点就是速度控制量的极性是负，和通常速度反馈控制的极性“恰好相反”，这是由于该控制方案是从调节车模



角度反馈进行设计和改进而得到的。第二点就是两个控制参数分别对应着误差信号的积分和比例值，所以在有的场合也可以称这两个参数为积分和比例控制参数。

## 2.4 车模方向控制

实现车模方向控制是保证车模沿着竞赛道路比赛的关键。直立车模所在的电磁组的道路中心线铺设有一根漆包线，里面通有 100mA 的 20kHz 交变电流。因此在道路中心线周围产生一个交变磁场。通过道路电磁中心线偏差检测与电机差动控制实现方向控制，从而进一步保证车模在赛道上。将在下面分别进行介绍。

### (1) 道路电磁中心线的偏差检测

道路电磁中心线检测简单的方法可以通过安装在车模前方的两个电磁感应线圈实现。线圈一般采用 10mH 的工字型电感。如图 2- 35 所示。

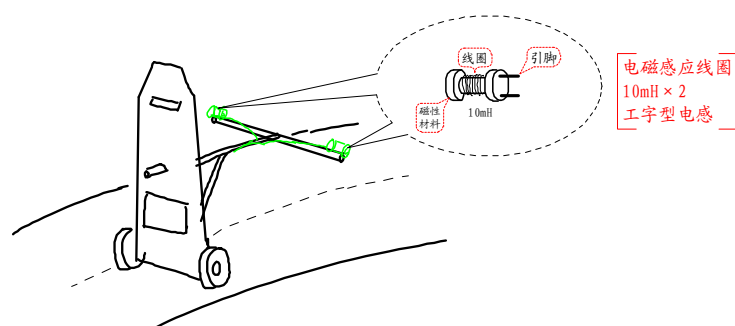


图 2- 35 检测道路中心电磁线方式

详细的参考设计方案请参见文档《电磁组竞赛车模设计参考方案，2010》。在后面电路设计中给出了具体的电路设计。

### (2) 电机差动控制

利用电磁线偏差检测信号分别与车模速度控制信号进行加和减，形成左右轮差动控制电压，使得车模左右轮运行角速度不一致进而控制车模方向。如图 2- 36 所示。

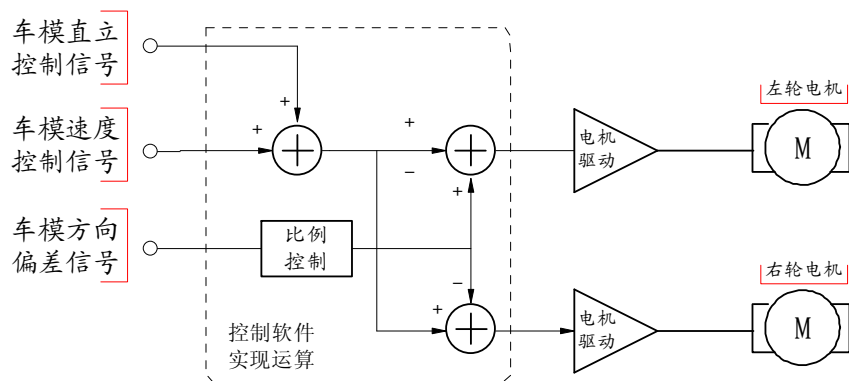


图 2-36 通过电机驱动差动电压控制车模方向

### (3) 方向控制算法

方向控制算法根据车模检测到电磁感应电压来生成电机差动控制量。通过左右电机速度差驱动车模转向消除车模距离道路中心的偏差。通过调整车模的方向，再加上车前行运动，可以逐步消除车模距离中心线的距离差别。这个过程是一个积分过程，因此车模差动控制一般只需要进行简单的比例控制就可以完成车模方向控制。但是由于车模本身安装有电池等比较重的物体，具有很大的转动惯量，在调整过程中会出现车模转向过冲现象，如果不加以抑制，会使得车模冲出赛道。根据前面角度和速度控制的经验，为了消除车模方向控制中的过冲，需要增加微分控制。

微分控制就是根据车模方向的变化率对电机差动控制量进行修正的控制方式，因此需要增加车模的转动速度检测传感器。可以使用陀螺仪传感器进行检测，如图 2-37所示图 2-37 检测车模转动速度的陀螺仪。陀螺仪在车模倾角检测中也使用。

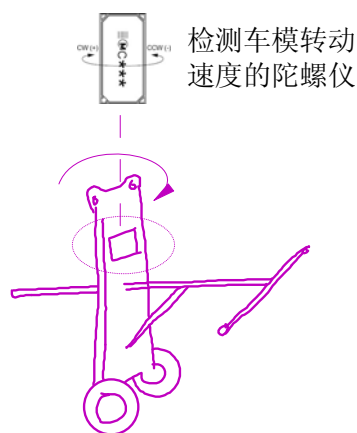


图 2-37 检测车模转动速度的陀螺仪

由于电磁导引线的磁场强度与检测线圈的距离和方向都有关系。当车模的方向偏差很大的时，比如在车模前方出现急转弯的时候，检测线圈中轴线与电磁导引线不再垂直，出现一个很大的角度偏差。此时两个检测线圈的感应电动势都下降。图 2- 38说明了这种情况。

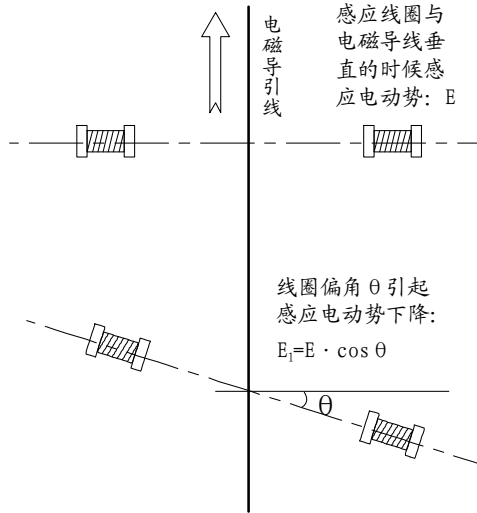


图 2- 38 电感线圈的偏角影响感应电动势

为了更能准确反映车模重心距离电磁线缆的距离差别，避免角度的影像，在进行方向控制时，使用左右两个线圈感应电动势之差除以左右两个线圈感应电动势之和，使用该比值进行方向控制。这样可以消除检测线圈角度的影响。

综上所述，得到车模方向控制算法框图，如图 2-39 所示。其中方向微分控制部分可以根据情况省略，这样可以进一步降低控制的复杂度。

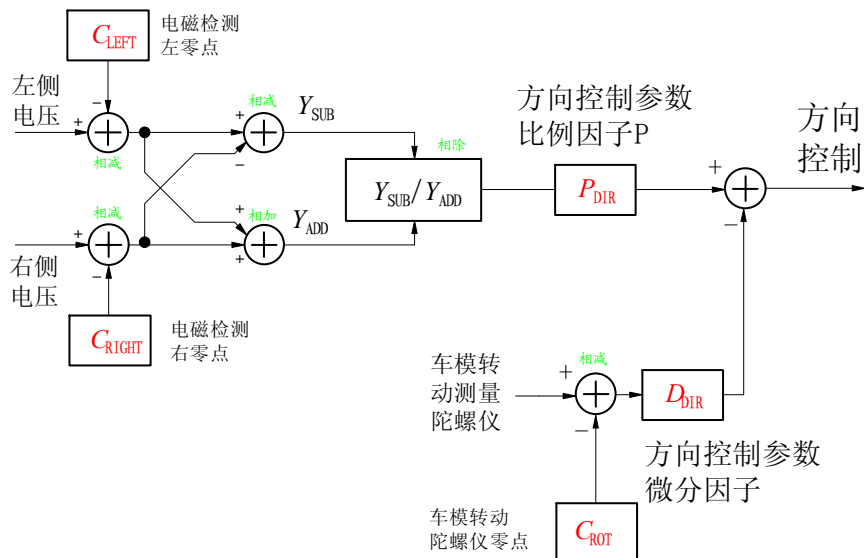


图 2- 39 车模方向控制算法

## 2.6 车模直立行走控制算法总图

通过上面介绍，将车模直立行走主要的控制算法集中起来，结合图 2-34 和图 2-39 形成控制算法总框图，如图 2-40 所示。

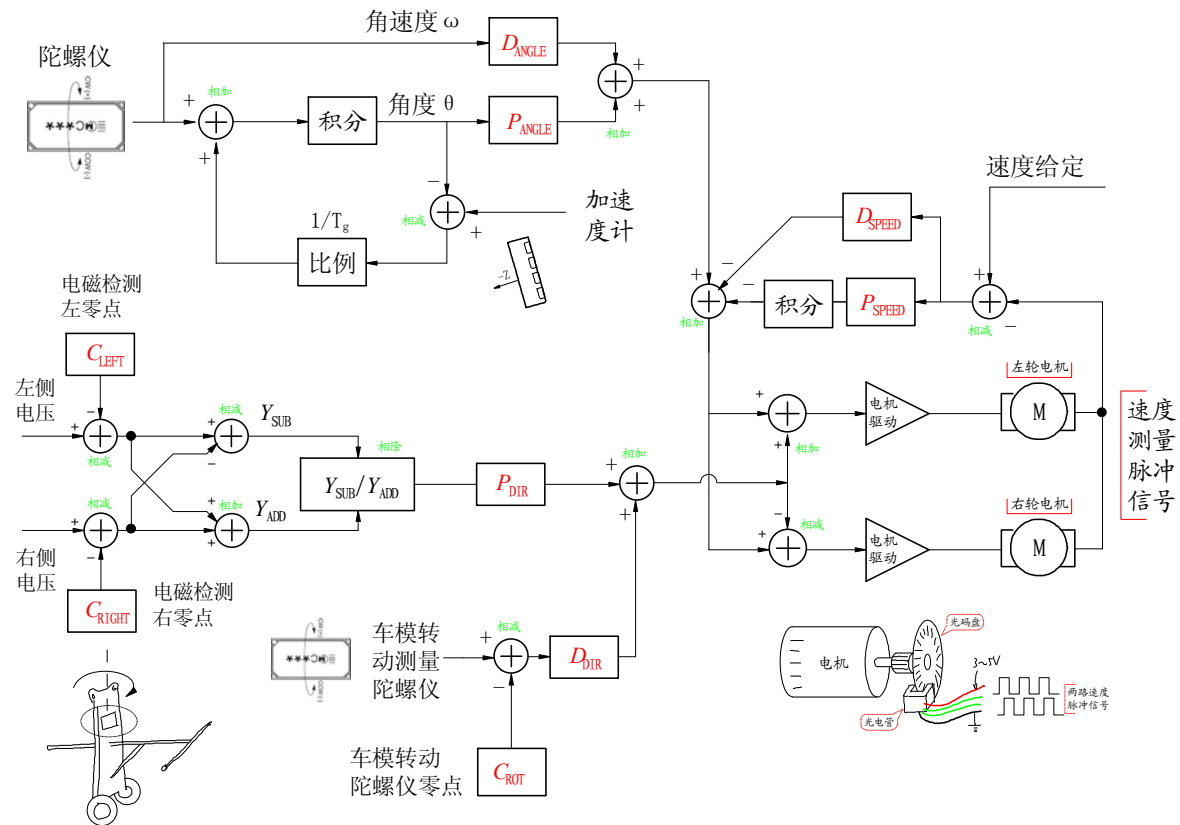


图 2-40 车模运动控制总框图

图 2-40 中还是保留了重力加速度传感器，在完整的车模控制中，它可以由速度控制部分替代。

为了实现车模直立行走，需要采集如下信号：

- (1) 车模倾角速度陀螺仪信号，获得车模的倾角和角速度。
- (2) 重力加速度信号（z 轴信号），补偿陀螺仪的漂移。该信号可以省略，有速度控制替代。
- (3) 车模电机转速脉冲信号，获得车模运动速度，进行速度控制。
- (4) 车模电磁偏差信号（两路），获得车模距离中心线的位置偏差，进行方向控制。

(5) 车模转动速度陀螺仪信号，获得车模转向角速度，进行方向控制。

上面采集信号中，可以简化掉重力加速度信号和车模转动速度陀螺仪信号。

在车模控制中的直立、速度和方向控制三个环节中，都使用了比例微分（PD）控制，这三种控制算法的输出量最终通过叠加通过电机运动来完成。

(1) 车模直立控制：使用车模倾角的 PD（比例、微分）控制；

(2) 车模速度控制：使用 PD（比例、微分）控制；

(3) 车模方向控制：使用 PD（比例、微分）控制。

可通过单片机软件实现上述控制算法。

在上面控制过程中，车模的角度控制和方向控制都是直接将输出电压叠加后控制电机的转速实现的。而车模的速度控制本质上是通过调节车模的倾角实现的，由于车模是一个非最小相位系统，因此该反馈控制如果比例和速度过大，很容易形成正反馈，使得车模失控，造成系统的不稳定性。因此速度的调节过程需要非常缓慢和平滑。

上述控制框图中只是原理性说明了控制关系。在软件实现过程中，还需要考虑到各个采集量的比率和零点值以及信号的极性。在图 2-40 中，方向控制中已经加入了零点矫正过程。同样的过程对于角度控制中的陀螺仪和加速计采集通道有需要进行。信号的极性与传感器的安装有关系，比如在车模转向控制中使用的车模转向陀螺的 Z 轴朝上和朝下，对应的车模转向角速度的极性相反，从而影响车模方向控制微分控制量与比例控制量之间的加减关系。其它部分的加减关系也与此相同。

详细的框图可以参见附录 5 中的算法框图。

## 第三章、电路设计篇

### 3.1 整体电路框图

设计车模控制系统的电路，首先需要分析系统的输入、输出信号，然后选择合适的核心控制嵌入式计算机（单片机），逐步设计各个电路子模块，最后形成完整的控制电路。

系统的输入输出包括：

**(1) AD 转换接口 (至少 5 路)**

- a) 电磁检测: 左右两路, 用于测量左右两个感应线圈电压。
- b) 陀螺仪: 两路。一路用于检测车模倾斜角速度, 一路用于检测车模转动角速度。
- c) 加速度计: 一路, 测量加速度 Z 轴输出电压。
- d) 辅助调试: (备用) 1 到 3 路, 用于车模调试、设置作用。

**(2) PWM 接口 (4 路)**

- a) 控制左右两个电极双方向运行。由于采用单极性 PWM 驱动, 需要四路 PWM 接口。如果采用双极性 PWM 驱动, 可以使用两路。

**(3) 定时器接口 (2 路)**

- a) 测量两个电机转速, 需要两个定时器脉冲输入端口。

**(4) 通讯接口 (备用)**

- a) SCI (UART): 一路, 用于程序下载和调试接口;
- b) I2C: (备用) 如果选择飞思卡尔公司的数字加速度计, 可以通过 I2C 接口直接读取加速度值。

**(5) IO 接口 (备用)**

4 到 8 路输入输出, 应用车模运行状态显示, 功能设置等。

上述接口中, 测量重力加速度传感器和车模转动速度的陀螺仪可以简化省略掉。

竞赛允许使用飞思卡尔公司处理器系列, 绝大部分都能够满足上面的控制要求。

本参考方案中选择飞思卡尔的 DSC 16 位处理器 MC56F8013 作为核心的控制处理器进行设计。车模控制电路整体框图如图 3-1 所示。

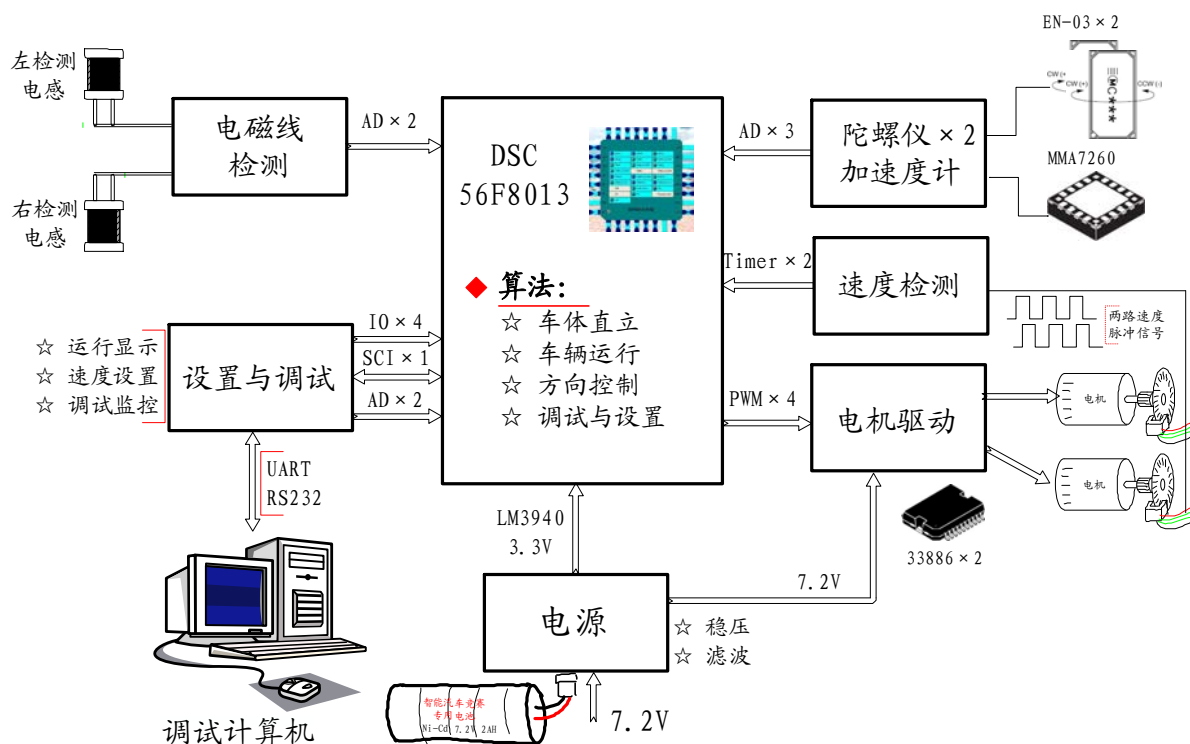


图 3-1 直立车模控制电路整体框图

根据图 3-1，控制电路划分为如下子模块：

- (1) 单片机最小系统：包括 DSC 处理器，程序下载调试接口等；
- (2) 电磁线检测：包括两路相同的电磁感应信号放大与检波电路；
- (3) 陀螺仪与加速度计：包括三个姿态传感器信号放大滤波电路；
- (4) 速度检测：检测电机光电码盘脉冲频率，实际上只包括了两个光电码盘的传感器；
- (5) 电机驱动：驱动两个电机运行功率电路；
- (6) 电源：电源电压转换，稳压，滤波电路；
- (7) 设置与调试：显示系统运行状态、速度设定、程序下载与监控。

以下将分别对以上电路给出设计参考方案。

### 3.2 DSC 介绍与单片机最小系统

单片机选择飞思卡尔公司 DSC MC56F8013，它体积小（32PIN TQFP），功耗低（3.3V 工作电压），运算速度快（32MIPS，DSP 结构），具有丰富的外设模块，非常适合控制

车模运行。它的主要外设包括：

- (1) PWM : 6 通道；
- (2) AD 转换器: 6 通道, 12bit；
- (3) 定时器: 16bit, 4 通道；
- (4) 丰富的外部串行接口: SCI, I2C, SPI；
- (5) IO 口: 最多可以提供 26 路, 可以与 5 V 器件直接相连。

此外, 内部还集成了时钟电路、电源检测电路以及看门狗电路等。

内部存储器资源包括: 16k程序Flash, 4k数据RAM。图 3- 2显示该单片机的内部资源情况。

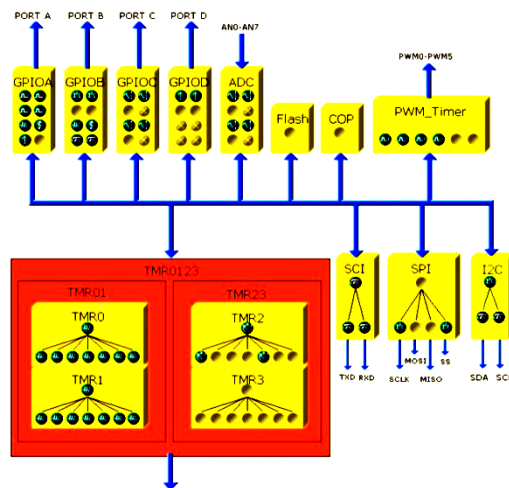


图 3- 2 56F8013 内部资源示意图

由于该单片机内已经包含了时钟和复位电路, 所以单片机的最小系统电路非常简洁。如图 3- 3所示。



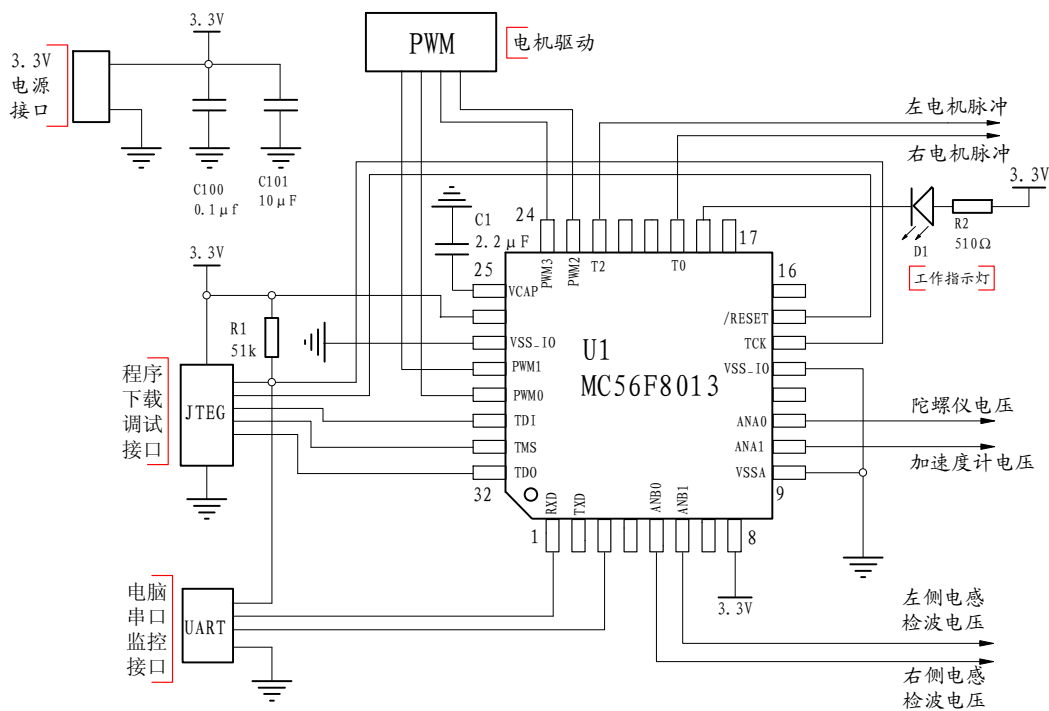


图 3-3 F8013 最小系统电路

所使用 F8013 单片机的资源包括：

- (1) 模拟量检测接口
  - a) ANA0 (PIN12)：检测车模倾角陀螺仪电压；
  - b) ANA1 (PIN11)：加速度计电压；
  - c) ANB0 (PIN5)：右侧电感检波电压；
  - d) ANB1 (PIN6)：左侧电感检波电压；
  - e) ANB2 (PIN7)：检测车模转动陀螺仪电压；
- (2) 电机转速脉冲接口
  - a) T0 (PIN19)：右侧电机光电码盘脉冲；
  - b) T2 (PIN22)：左侧电机光电码盘脉冲；
- (3) 电机 PWM 驱动接口
  - a) PWM0-3 (PIN23,24,28,29)：电机驱动。
- (4) 程序下载调试 JTEG 接口
  - a) TDI (PIN30)；
  - b) TDO (PIN32)；
  - c) TMS (PIN31)；

- d) TCK (PIN14);
  - e) /RESET(PIN15)。
- (5) 串口监控 UART 接口:
- a) RXD (PIN1);
  - b) TXD (PIN3)。

其它没有使用的 IO 和模拟量口可以用作状态显示、运行设置以及辅助调试作用。

使用快速制板方法制作的最小单片机系统实物图如图 3-4所示。

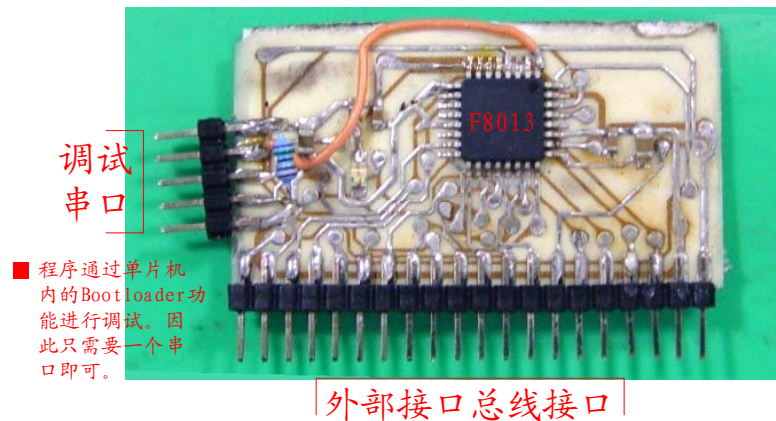


图 3-4 F8013 最小系统电路实物

图 3-4显示的实际电路中，将外部所需要的各个端口通过总线方式引出，通过统一的接口底板连接其他各子电路。

上面电路板也没有 JTEG 的程序接口。程序的开发主要是通过单片机内部的 Bootloader 程序完成程序的下载和调试的，因此单片机最小系统只需要一个 UART 接口便可以进行，无需额外的调试器。相关的资料可以参见飞思卡尔公司网站中的介绍。

### 3.3 倾角传感器电路

车模倾角传感器电路主要是将陀螺仪信号进行放大滤波。由于加速度传感器采用是低g值的传感器MMA7260，它的输出信号很大，不需要再进行放大。电路图如图 3-5所

示。

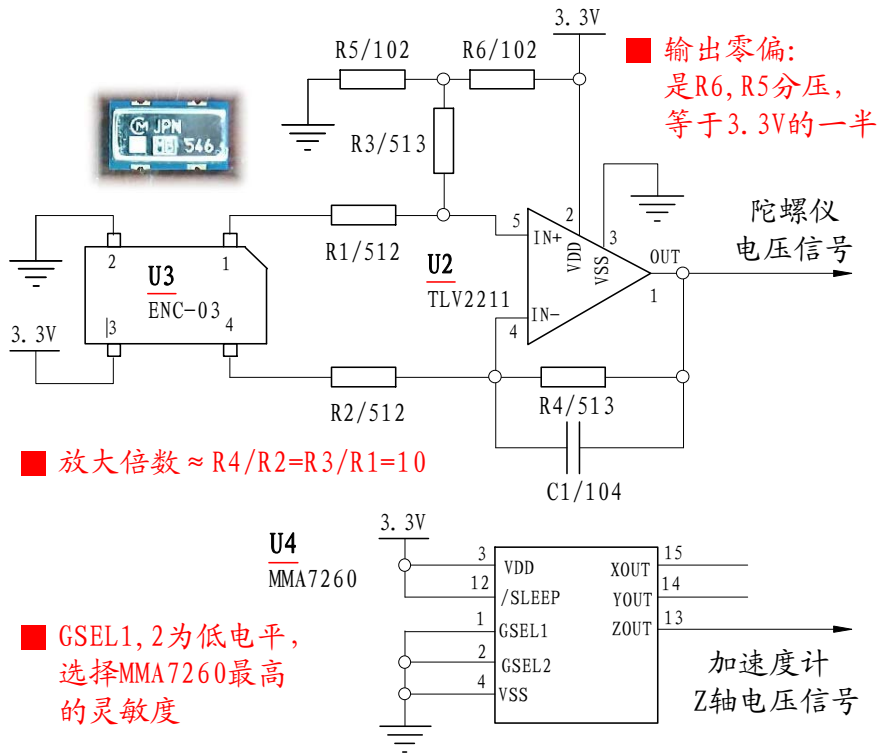


图 3-5 陀螺仪、加速度传感器电路

上图中，将陀螺仪的输出信号放大了 10 倍左右，并将零点偏置电压调整到工作电源的一半（1.65V）左右。放大倍数需要根据选取的传感器输出灵敏度设计，可以选择 5 至 10 倍范围都可以满足车模控制需要。

将上述电路单独制作成小的电路板，可以比较方便放置在车模的最稳定的位置。实际的电路图如图 3-6所示。

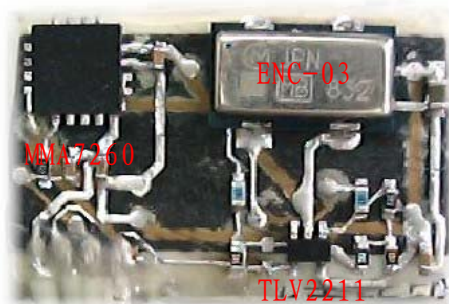


图 3-6 车模倾角传感器电路实物图

以上电路只是将传感器的信号进行了放大处理，角度和角加速度的计算都是依靠单

片机的软件来完成。对于角度信号的处理也可以通过硬件电路来完成，这部分内容参见 3.7 节中的内容。在 3.7 节中还给出了只利用两片 MMA7260 加速度传感器便可以获得车模的角度和角速度的电路。

参与控制车模转向的陀螺仪的电路和上面的倾角测量电路中陀螺仪的电路是一样的。只是在安装转向陀螺的时候，将该陀螺垂直安装，由此可以测量车模的转向角速度。在此就将其省略了。

### 3.4 电机驱动电路

由于车模具有两个后轮驱动电机，因此需要两组电机驱动桥电路。图 3-7 选用了两片飞思卡尔公司专用电机驱动芯片 33886 组成了电机驱动电路。

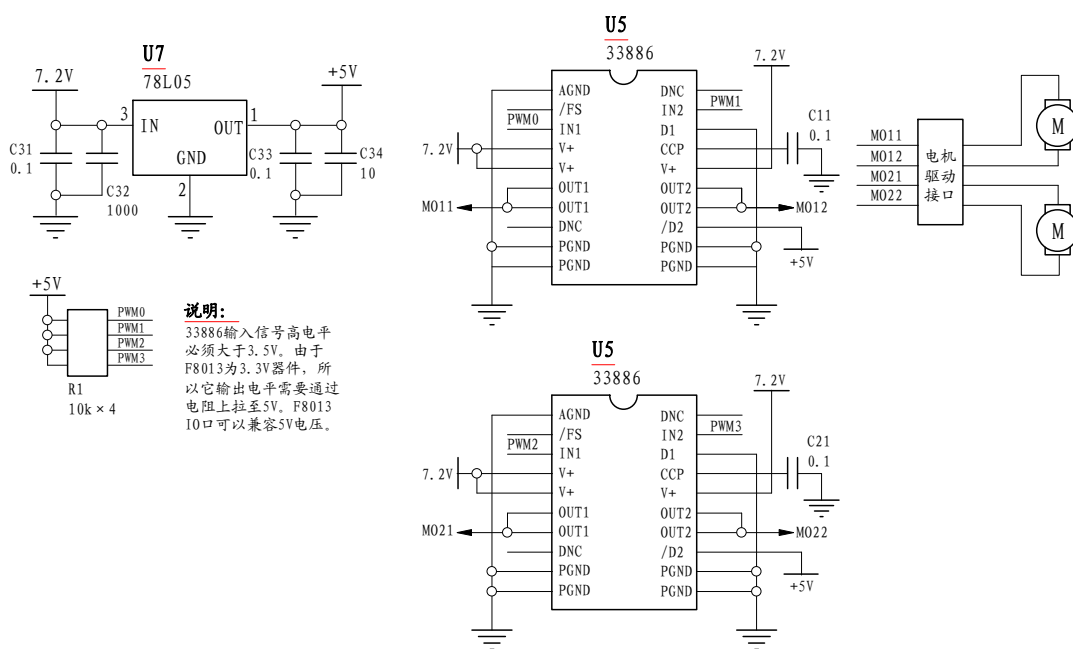


图 3-7 双电机驱动电路

图 3-7 中的 DSC F8013 是 3.3V 器件，它的 IO 输出电压最高位 3.3V，达不到 33886 对于高电平必须大于 3.5V 的要求，所以在电路中专门设计了 5V 电源，将 33886 的驱动信号上拉至 5V。由于 F8013 的 IO 端口可以容忍 5V 电压，所以上面的电路便可以使得 33886 的驱动信号电压达到 5V。

为了提高电源的应用效率，驱动电机的 PWM 波形采用了单极性的驱动方式。也就

是在一个PWM周期内，施加在电机上的电压为一种电压，如图 3- 8所示。

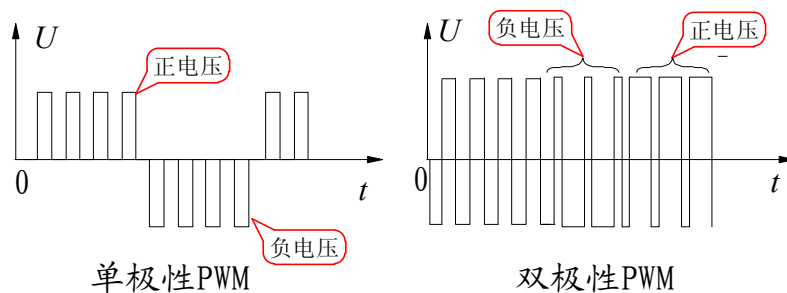


图 3- 8 单极性 PWM、双极性 PWM

因此每一路电机为了能够实现正反转，都需要两个PWM信号。两个电机总共需要 4 路PWM信号。具体实现的驱动电路如图 3- 9所示。

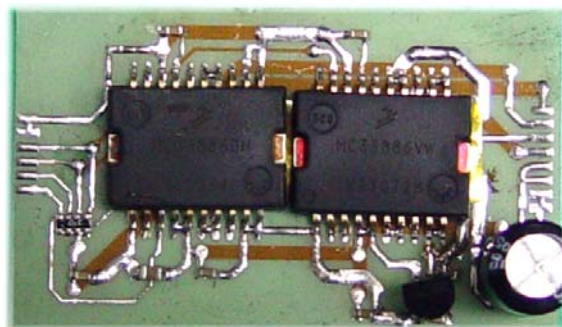


图 3- 9 两片 33886 组成的电机驱动电路

图 3.9 中，为了防止电机输出电流对于电源的冲击，在电路板的电源输入（7.2V）端口并联了一个 1000 微法的电容。

### 3.5 速度传感器电路

电机速度传感器使用了固定在电机输出轴上的光码盘以及相互配合的光电对管器件，如图 3- 10所示。

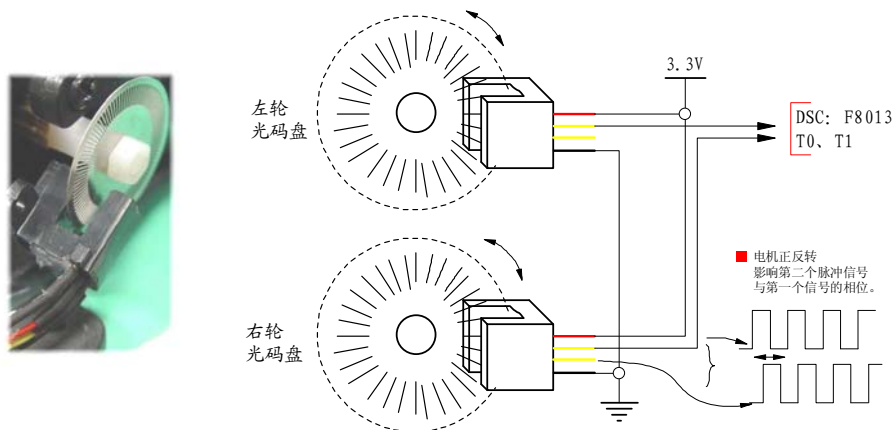


图 3-10 速度传感器电路

由于光电管器件直接输出数字脉冲信号，因此可以直接将这些脉冲信号连接到单片机的计数器端口。

每个光电管输出两个脉冲信号，它们波形相同，只是相位相差  $90^\circ$ 。如果电机正转，第二个脉冲落后  $90^\circ$ ；如果电机反转，第二个脉冲超前  $90^\circ$ 。可以通过这个关系判断电机是否正反转。在实际电路中，只检测了一路脉冲信号，通过它的频率测量得到电机的转速。电机的转向是通过施加在电机上的电压正负进行判断的。虽然由于车模惯性存在，可能在电机转速低的情况下，电机的转动方向与施加在其上的电压正负有可能不同，从而造成电机转速测量不准。由于比赛时竞速比赛，一般情况下电机都是运行在高速状态下，电机的转向与电压的方向是一致的。通过实验验证这个方法可以有效判断电机的转动方向并进行速度控制。

### 3.6 电磁线检测电路

#### (1) 分立元器件电磁放大检波电路

道路中心线的电磁线检测是保证车模能够运行在赛道上。由于电磁组在第五届竞赛中已经设立，在 2010 年竞赛秘书处公布了电磁线检测的参考设计方案。详细设计原理和电路请参见附录中的参考文献。

图 3-11 给出了其方案的电路图。该方案采用了单级三极管共射放大电路对检测的电磁信号进行放大，后级采用倍压整流电路进行检波，可以得到左右两个感应线圈所检测到磁场的强度。为了提高电路的灵敏度，三极管应该采用截止频率大于  $100\text{MHz}$ ，电流

放大倍数大于 200 的三极管。被压整流的二极管采用正向导通电压比较低的锗二极管或者肖特基二极管。

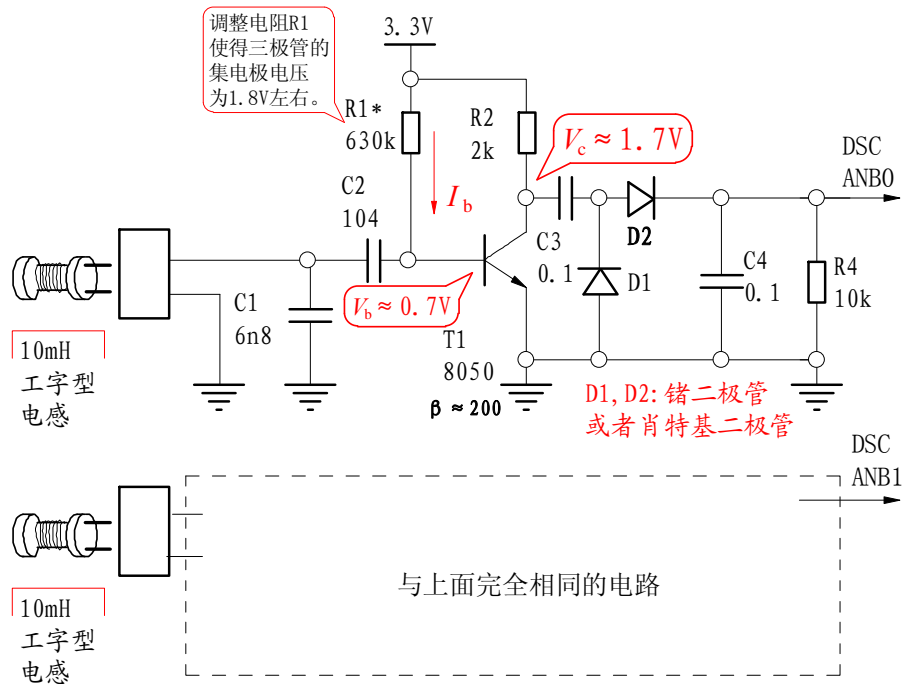


图 3-11 基于三极管的电磁信号放大检波电路

为了保证三极管放大电路的信号放大的动态范围足够大，需要将三极管的集电极电压调整在电路工作电压的一半左右： $V_c \approx 3.3 / 2 = 1.65V$ 。这需要通过调整三极管基极偏置电阻  $R_1$  的完成。

$R_1$  取值可由公式计算而得： $R_1 = \frac{V_{cc} - V_b}{I_b} = \frac{V_{cc} - V_b}{I_c / \beta}$ ，其中集电极电流

$I_c = \frac{V_{cc} - V_c}{R_2}$ 。将电路参数代入可以计算机得到  $R_1 = 630k\Omega$ 。

根据上面电路制作的电路图如图 3-12 所示。

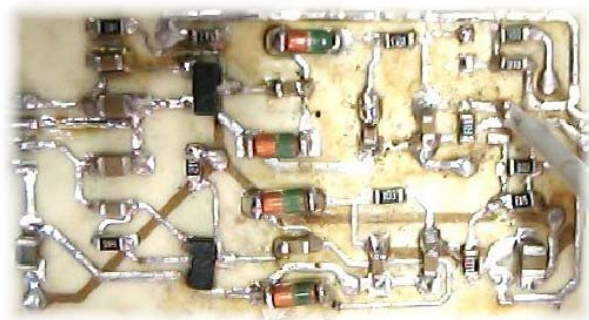


图 3-12 基于三极管的电磁放大检波电路实物图

## (2) 集成运放电磁放大检波电路

三极管电磁放大检波电路简单实用，已经在比赛中得到了广泛的使用。如果需要进一步提高检波的灵敏度，可以再增加一级三极管放大电路。使用三极管组成的分立元件的放大检波电路存在着一些缺点：

- (1) 工作点电压调整比较复杂；
- (2) 电路的放大倍数依赖于三极管的电流放大倍数和基极导通阻抗等，比较分散；
- (3) 由于检波二极管的正向导通电压的存在对于非常弱的电磁信号无法进行检波。

采用集成放大电路可以简化电路设计，提高电路的性能。下面给出了一种基于输出满电压量程（Rail-to-Rail, 轨对轨）运算放大器的电磁放大检波电路。

R-R运放一般工作在单电源供电状态下，输出电压范围基本接近于电源电压和地。输入的共模电压范围宽。常用到的R-R运放包括LMV321, LMV358, LMV324, AX4451等。请注意前面三种运放名称都是LMV，这些运放都有前缀为LM相对应型号的运放。LMV运放与LM运放封装管脚相同，只是LMV为输出满电压量程输出满电压量程。表格3-1给出了LMV358的主要电气性能，它的共模输入电压范围可以低于0V。

表格 3-1 LMV358 主要性能

参数	测试条件	取值范围	单位
Vcc: 工作电压 (单电源供电)		2.7 至 5.5	V
Vicr: 共模电压输入范围	CMRR>50db	-0.2 至 1.9	V
输出电压范围	RL=10k	60 至 Vcc-10	mV
B1: 单位增益带宽	CL=200pF	1	MHz
Vio: 输入偏置电压		1.7	mV



图 3- 13为基于R-R运放LMV358 的电磁放大检波电路。

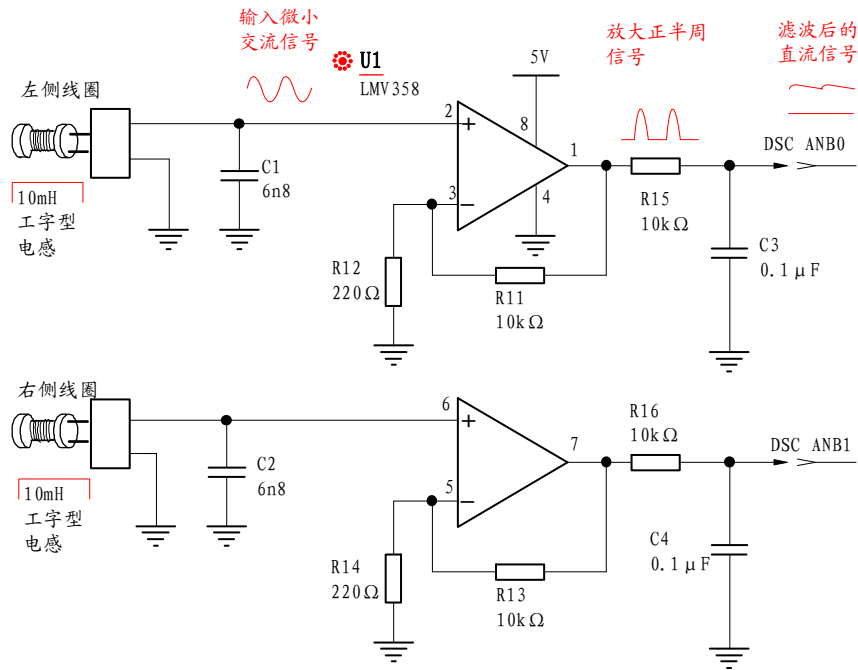


图 3- 13 使用 R-R 运放进行电磁信号放大检波

在该电路中有意将运放的中点设置在 0V，因此运放只对输入信号的正半周信号进行放大，负半周的信号则无法放大。运放对信号进行了放大的同时也完成了与检波（半周检波）。检波后的信号经过RC滤波后得到信号的直流分量，送到单片机进行检测。图 3- 14为电路放大检波后的波形。

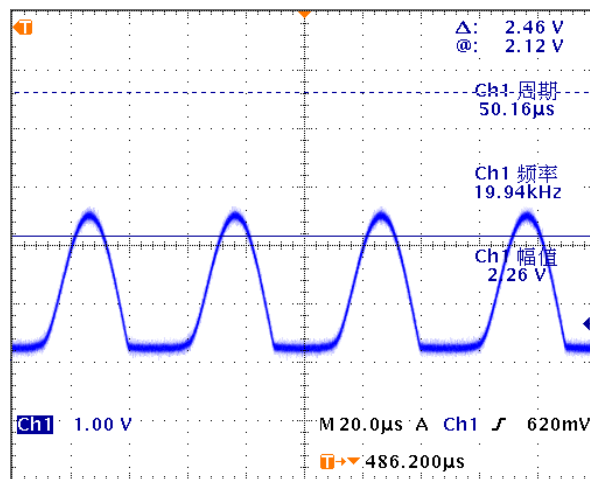


图 3- 14 LMV358 放大检波输出波形

下图是基于 LMV358 的设计的两个电磁感应通道的放大检波电路实物图。



图 3-15 基于 LMV358 两个通道的放大检波电路实物图

由于运放的共模输入电压范围可以小于 0V，所以该电路对于弱小信号具有非常好的放大与检波性能。电路的放大倍数由反馈电阻网络决定，工作稳定。由于检测电磁场的电感谐振回路直接接入运放的正向输入端，输入阻抗很高，从而提高了谐振回路的品质因数（Q 值），使得输入频率选择能力大大增强。

由于输出信号时半波检波信号，所以输出信号的最大值不会超过运放的电源一半。为了提高电路放大信号的动态范围，选择运放的工作电压为 5V。

由于信号的频率是 20kHz，所以选择的 R-R 运放的单位增益带宽一定要足够大，这样才能保证对于 20kHz 信号的放大能力，建议运放的单位增益带宽至少大于 1MHz。

### 3.7 角度计算电路

在车模控制原理篇中介绍车模角度和角速度测量的方法有两种，第一种是通过陀螺仪和加速度传感器来实现，第二种可以通过两个加速度传感器来实现。角速度经过反馈积分可以得到与加速度传感器一致的平滑角度信号。这部分的信号处理可以通过单片机软件来完成，也可以通过外部运放电路来实现。本节给出这部分信号处理的电路实现形式。

#### (1) 由双加速度传感器计算角速度

根据 2.3 节测量车模角度介绍，安装在车模顶部和底部的两个加速度传感器

MMA7260 的 Z 轴信号经过相减，便可以获得车模的角加速度。将此信号进行积分便可得到车模的角速度。图 3-16 给出了实现的电路图。

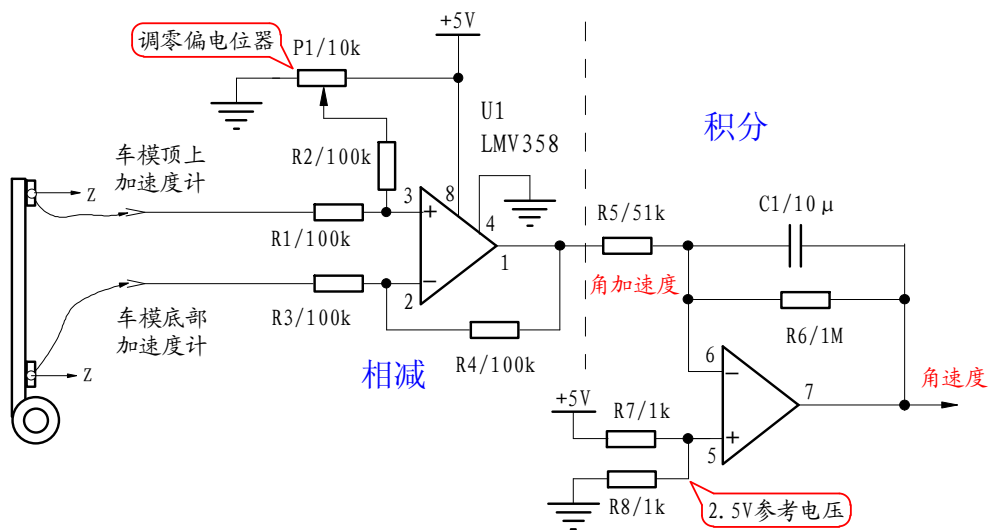


图 3-16 双加速度测量角速度电路

上面电路图由两部分组成，第一部分是实现减法运算获得车模的角加速度信号。第二部分是积分运算，得到车模的角速度。下图是实际信号的波形图。

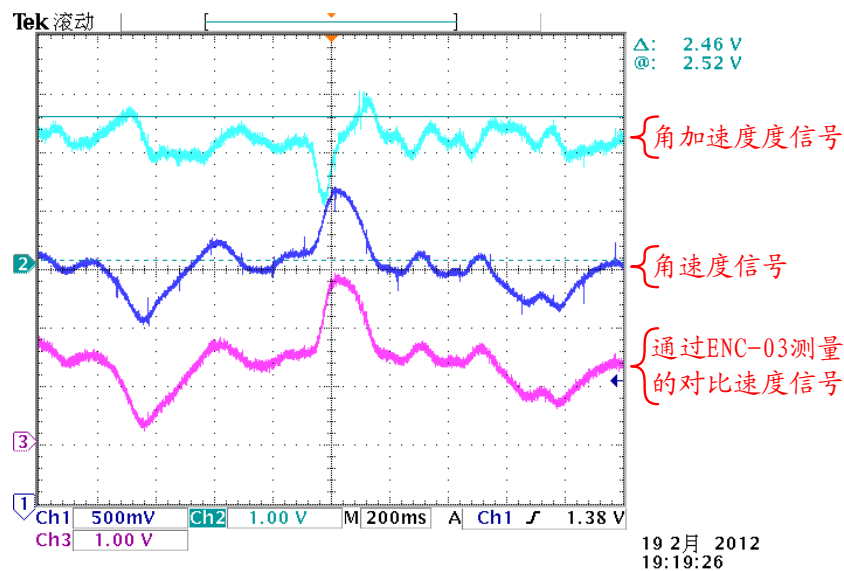


图 3-17 双加速度计测量角度波形图

为了验证该电路的性能，图 3-17 中还同时采集了安装在车模上的 ENC-03 测量信号。从波形上可以看出由加速度传感器积分得到车模的角速度与陀螺仪测量的角速度的波形是一样的，验证了该方法的正确性。

为了进一步简化电路，可以将车模顶部的加速度传感器反向安装，这样顶部的传感器 Z 轴信号的变量便和底部加速度传感器的信号符号相反，只要将两个信号通过电阻叠加在一起便可以实现信号的相减，从而可以进一步简化电路，如图 3-18 所示。

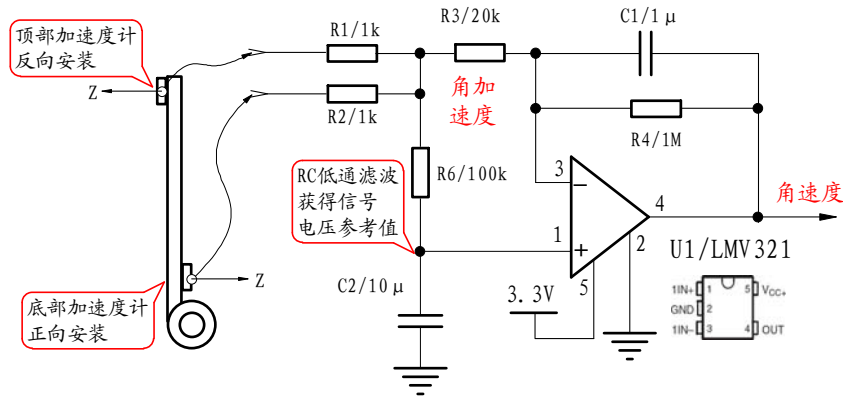


图 3-18 简化角速度电路

下图便是通过示波器观察到简化后电路各部分的信号，通过固定在同一车模上的 ENC-03 输出信号的对比，可以验证简化后的加速度电路的正确性。

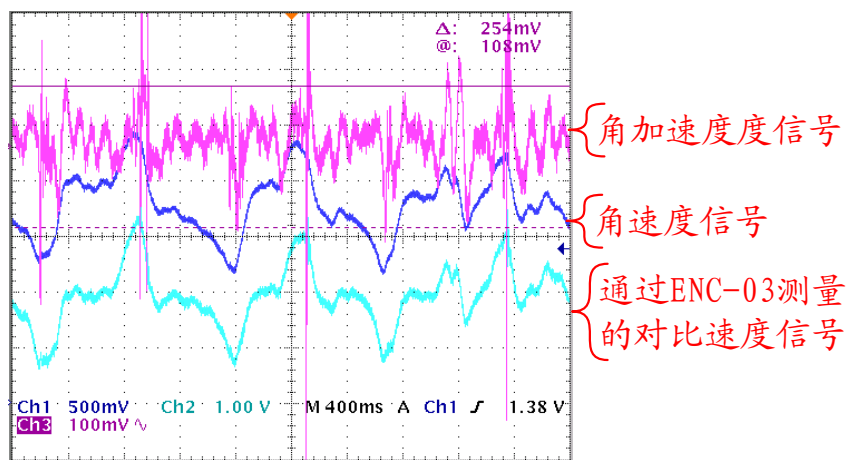


图 3-19 实测车模角速度信号波形

## (2) 角度计算电路

根据 2.3 节中介绍，通过对角速度信号进行积分便可以得到车模的角度。为了抑制角速度信号的漂移以及积分电路的漂移，需要根据加速度传感器给出的 Z 轴信号进行角度信号矫正。图 3-20 电路是根据前面介绍的信号处理框图获得角度信号处理电路。

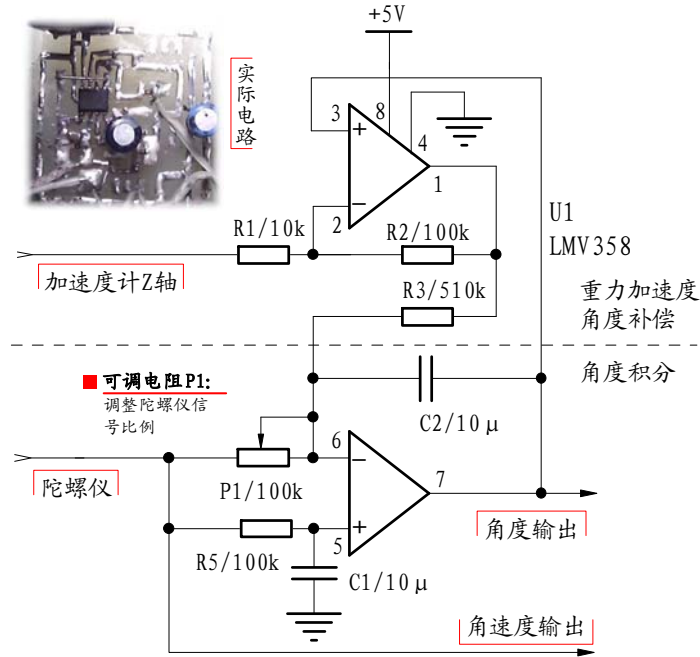


图 3- 20 角度信号处理电路

该电路包括角度积分以及重力加速度补偿电路两部分。调节电位器 P1 大小可以改变角速度的比例系数。P1 越大，角速度比例值越小；P1 越小，角速度比例值越大。通过调节 P1 使得角度输出可以很好的跟踪加速度传感器 Z 轴信号输出。下图显示了不同的 P1 值对应电路角度输出跟踪 Z 轴信号的情况。P1 过大，角度输出跟踪 Z 轴慢。P1 过大，角度输出跟踪 Z 轴有过冲。P1 调整到合适的值时，角度输出便可以 and Z 轴输出保持一致，同时信号非常平滑。

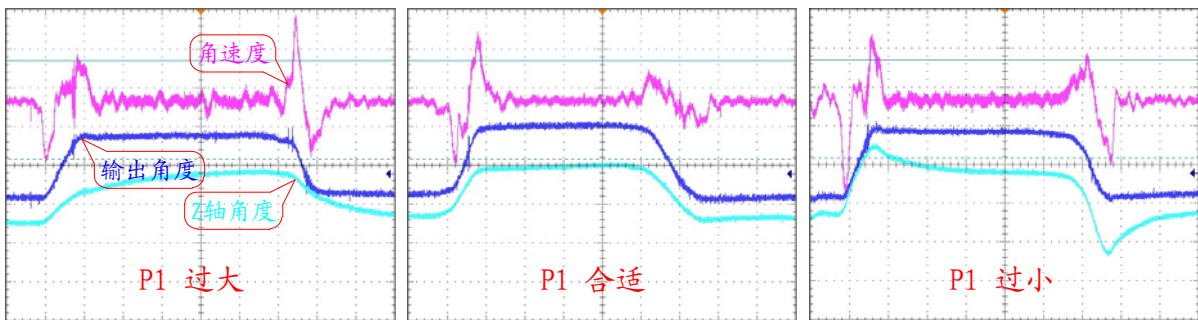


图 3- 21 不同角速度比例情况下输出波形

为了进一步简化电路实现，分析角度计算环节的传递函数（如图 3- 22所示）。角度输出是由陀螺仪和加速度计Z轴信号通过一个一阶惯性环节叠加而成。因此可以只是用一个运放就可以完成角度的计算，上面的电路可以进行简化，简化后的电路如图 3-23

所示。

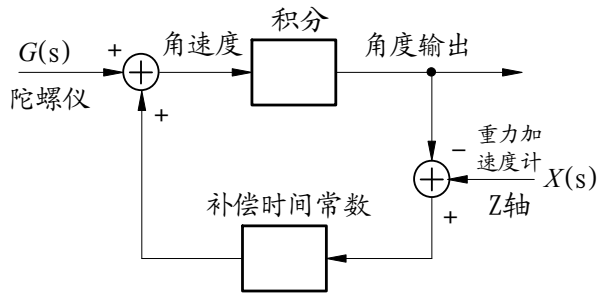


图 3-22 角度计算环节的传递函数

下图给出了使用一个运算放大器计算角度的电路。该电路的功能和图 3-22是一样的。其中电位器P1可以调节陀螺仪的比例因子。

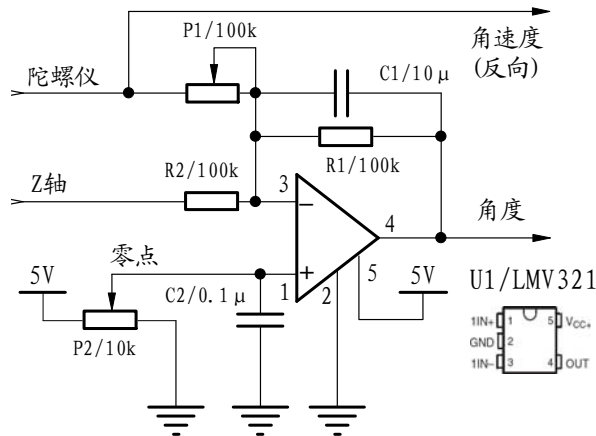


图 3-23 一个运算放大器实现角度计算

图 3-24给出了不同的P1阻值对应的电路输出。P1值过大，对应加速度计的比例小，角度输出跟踪Z轴输出比较慢。如果P1只过小，对应加速度计比例大，输出角度出现过冲。通过调整P1可以获得输出角度跟踪速度快而没有过冲。由于只是用了一个运放，因此电路的角度和角速度的输出极性相反，因此在单片机软件处理的时候需要通过软件改变角速度的极性。

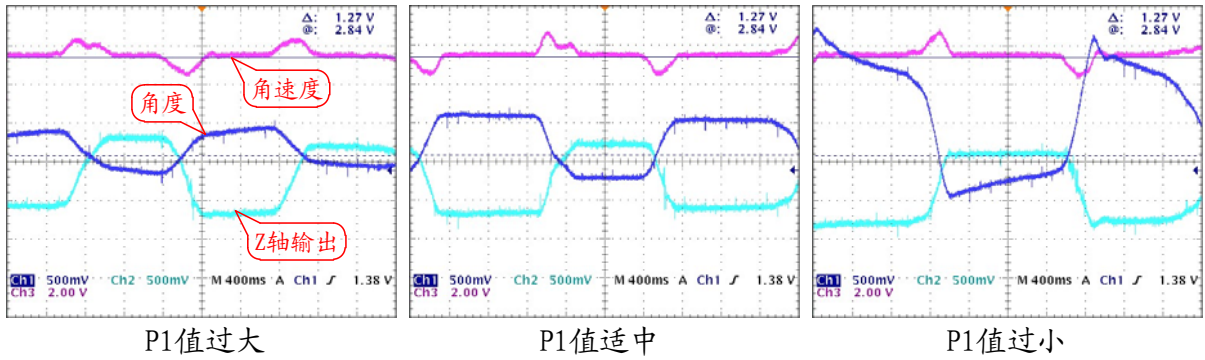


图 3-24 不同 P1 阻值对应的输出波形

根据前面设计，可以将角度测量和信号处理进行组合，形成统一的简化电路。如图 3-25 所示。该电路可以输出车模的角度和角速度取值。通过硬件实现角度信号处理，可以简化程序编写和调试，提高软件的可靠性，加快车模初学者制作过程。所以这部分电路适合车模初学者使用。

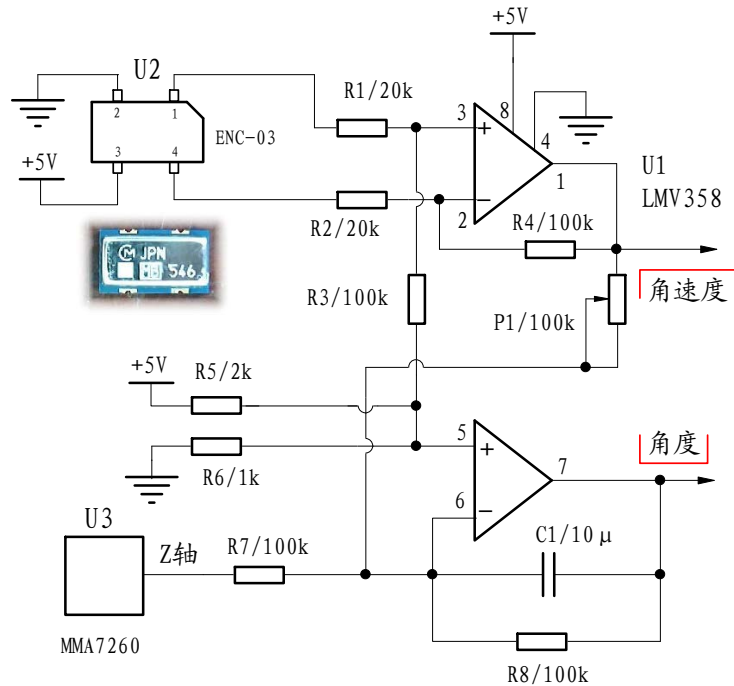


图 3-25 简化的角度和角速度处理电路

以上各部分的电路都通过实际验证，可以实现对车模的直立控制。

### 3.8 车模控制电路全图

汇总前面各个部分的电路设计，最终选定的车模控制电路全图如图 3-26 所示。

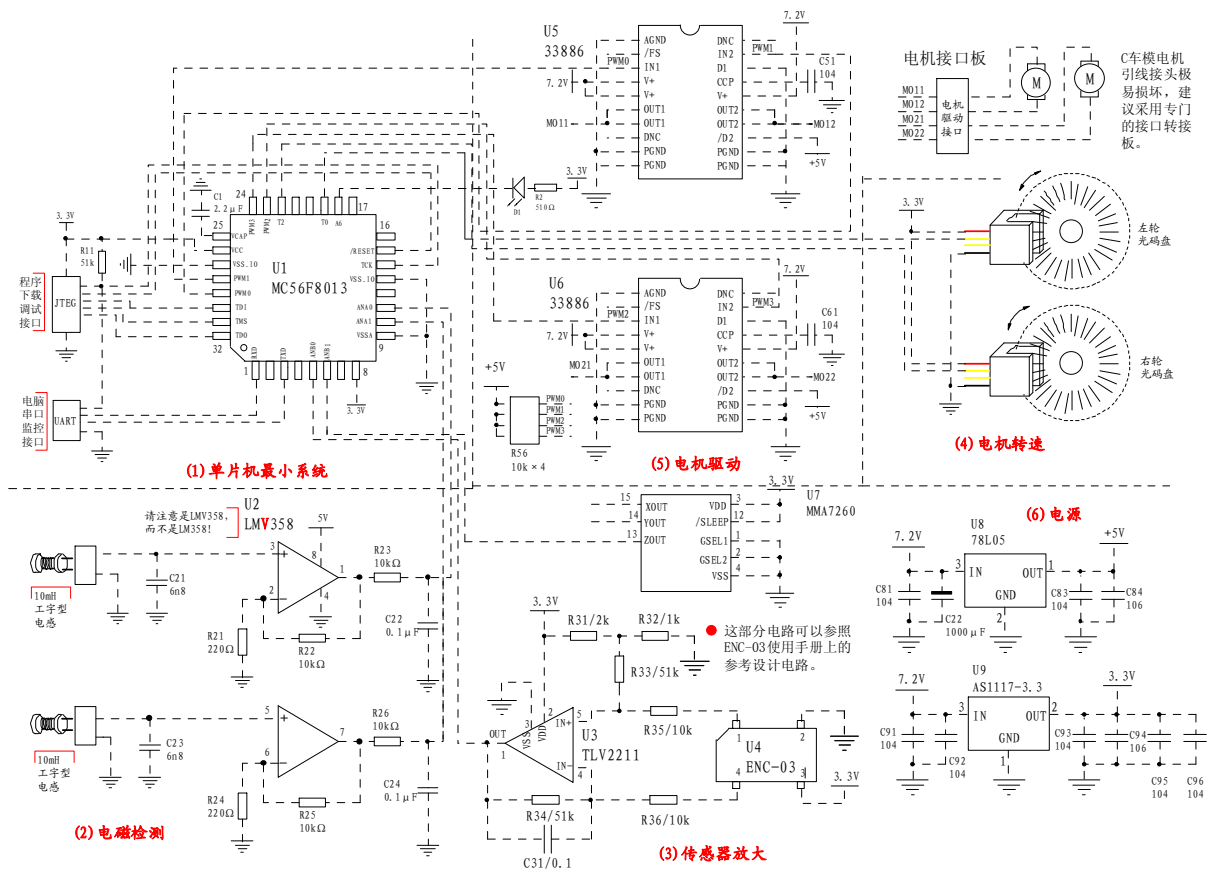


图 3-26 车模控制电路全图

控制电路包含有数字电路与模拟电路，小信号电路和大功率电路。设计电路板时，可以将全部电路集成在一个电路板中实现，也可以分解成若干块电路分别制作。从电路原理图到设计电路板，不仅需要注意电路板引线的逻辑功能，还需要考虑到电路板的布线，防止电路各部分之间相互干扰，从而保证电路的稳定可靠。

在上面的电路中没有涉及到相关的调试电路，在后面的调试过程中，实际上使用到了单片机的串口、另外两路 AD 输入端口（ANA2,ANB2）等资源。大家可以根据自己的调试方案设计完成这部分的内容

图 3-26 电路没有角度计算电路，对于角度测量部分是由单片机内部软件实现。

以上给出了车模控制电路的主要电路设计，此外还需要设计电源管理电路、参数设置与监控电路、程序调试与运行接口电路等。同学们可以根据往届竞赛技术报告进行相关的设计。



## 第四章、机械设计篇

良好的车模机械设计与制作，对于车模稳定运行、安全调试都非常重要。如下仅就车模简化改装与传感器安装两个方面进行讨论。

### 4.1 车模简化改装

由于今年电磁组车模采用了原来竞赛 C 型车模，它是双后轮驱动，前轮舵机转向的运动模式，而竞赛规定 C 型车模直立行走，因此车模前轮以及部分相关部件都可以进行简化。具体可以参照以下改装步骤：

- (1) 去掉前轮及其支撑部件，去掉后轮悬挂缓冲支架  
拆卸后的情况如下图所示。

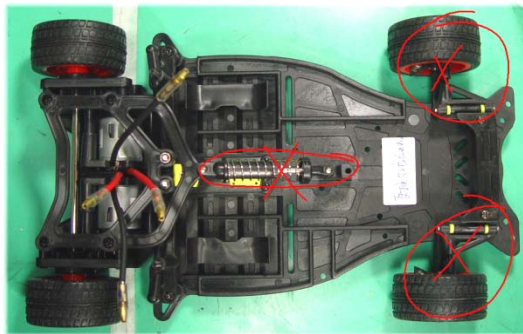


图 4-1 完整的 C 型车模底盘

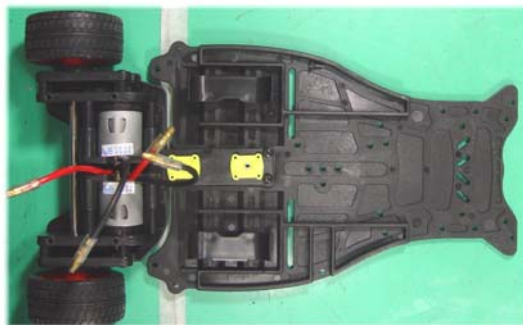


图 4-2 简化后的 C 型车模底盘

- (2) 固定车模底盘与后轮支架

原有车模为了减轻后轮振动对于车体的影响，后轮的支架与底盘之间采用了活动连接方式。但是，为了保证车模直立车体稳定性，需要将原有车模地盘与后轮支架固定在

一起。最简便的方式就是可以使用热熔胶在后轮支架与底盘之间的缝隙处进行粘接。这样后轮与车体之间形成一个刚体，便于进行直立控制。图 4-3 所示为热熔胶粘接的位置。



图 4-3 使用热熔胶固定电机支架与车模底盘

### (3) 去掉后轮，准备安装电机测速光码盘

拆装后轮可以使用原车模配备的十字套筒扳手。拆卸后的车模如图 4-4 所示。

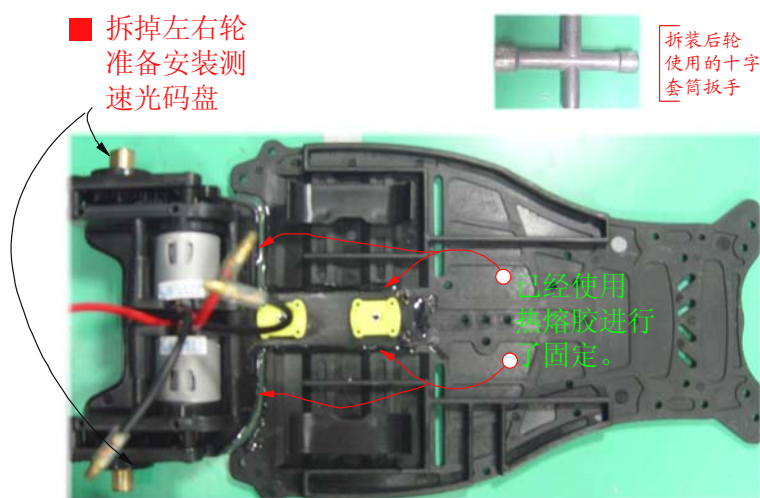


图 4-4 去掉后轮之后的车模底盘

在图 4-4 中，请注意使用热熔胶固定后轮支架与车模底盘的情况。

在安装测速传感器之后，再将两个后轮重新安装在后轮支架上。

请注意比赛规则中对于车模底盘改动的限制，不允许使用别的材料对于车模底盘进行较大的改动。

### (4) 加固电机引线

C 车模的后轮电机引线管脚非常单薄，多次晃动之后极易从根部折断，造成电机无

法使用。建议增加一个电机转接板，将电机的引线先焊接的转接板上，然后将转接板固定在电机支架或者车模地板上，这样拆卸相关的电路板时则不会造成电机引线的折断。



图 4-5 电机引线转接板

当然也可以直接使用热熔胶或者复合胶水对于电机引线管脚进行封装固定，也可以起到保护电机引线的作用。

此外，对于驱动电机同轴的金属轴实际上也没有太大的作用，如果为了其它安装目的，也可以拆除它们。

## 4.2 传感器安装

车模中的传感器包括有：速度传感器，车模姿态传感器（陀螺仪、加速度计），以及电磁检测感应线圈。下面分别介绍这些传感器的安装。

### (1) 速度传感器安装

速度传感器是安装在驱动电机输出轴上的光电码盘以及相配合的光电对管。固定光电码盘可以使用复合胶水，将光电码盘的塑料轴与电机的输出轴粘合。如图 4-6 所示。

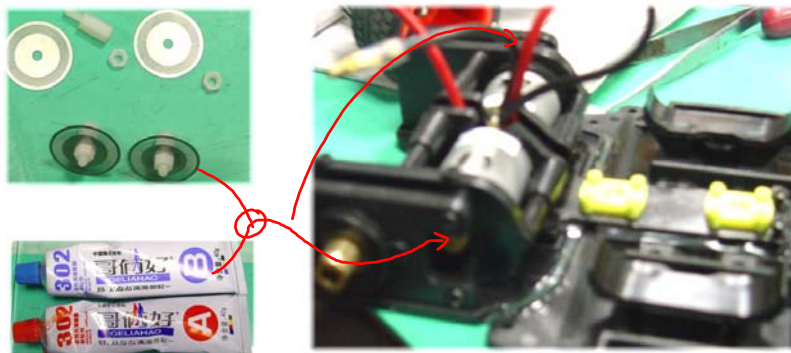


图 4-6 使用复合胶水固定光电编码盘

粘合码盘之后，再固定检测光电对管。可以使用热熔胶将光电对管固定在后轮支架上固定光电对管，使得光电码盘在对管的中间缝隙中通过。如图 4-7 所示。

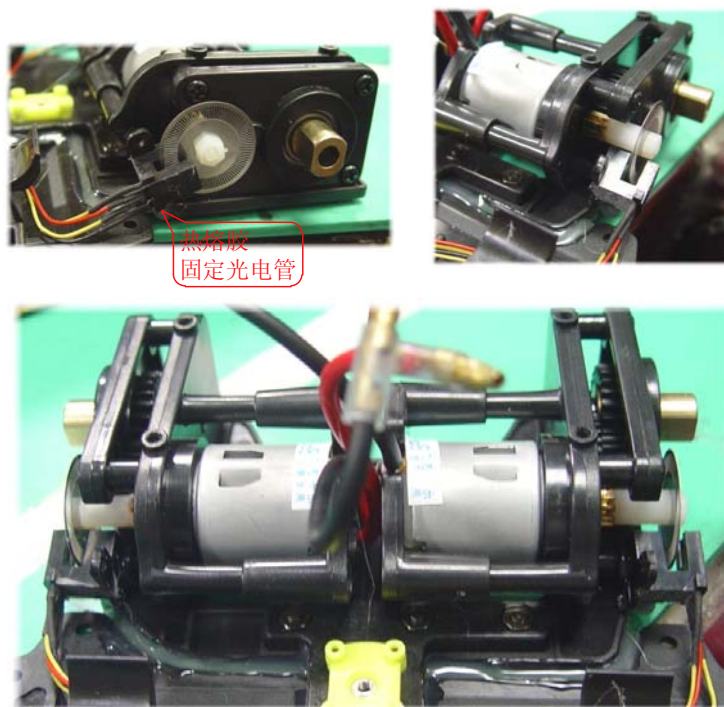


图 4-7 固定好的光电码盘和光电检测管

请注意，码盘的固定一定要使用热稳定性比较好复合胶水，比如“哥俩好”。由于电机在运行过程中发热，而且振动大，所以千万不要使用热熔胶固定码盘。

上述测速传感器固定完毕之后，便可以将后轮重新安装在后轮的支架上。

## (2) 电磁传感器安装

本设计方案中的电磁传感器为两个工字型的 10 毫亨电感。为了能够更好的检测前面的道路，一般将这两个工字型的电感尽可能的安装在车模运行前方较远的地方。由于车模是直立运行，可以考虑如下的参考方案，如图 4-8 所示。

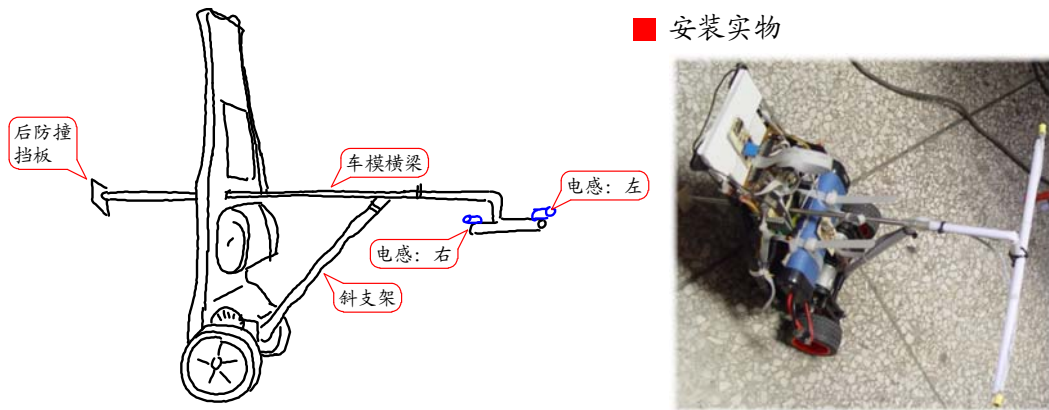


图 4-8 电磁传感器支架

请注意，竞赛规则对于车模外形尺寸的限制。由于本届比赛只是限制了车模的宽度，要小于 25 厘米。对于车模的长度和高度均未限制。

由于本届比赛没有要求车模在竞赛完毕后自动停止在其实现后三米的赛道上，因此无需检测赛道起始线上的永磁铁。

为了避免车模在通过十字交叉路口时受到的影响，在固定两个电感线圈的时候，尽量保持这两个线圈呈水平位置，并且在一条直线上，如下图所示。



图 4-9 电磁线圈水平安装

如果电磁线圈安装不在同一条水平线上，那么车模在过十字交叉路口的时候，水平方向的电线中的磁场就会在这两个线圈上产生额外的感应电动势。这些电动势会造成车模方向控制的不稳定性。

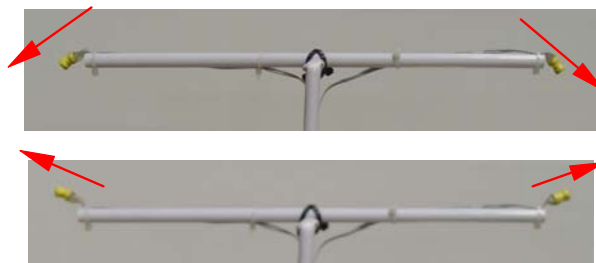


图 4-10 电磁线圈安装不在同一条水平线上

由于电磁线圈在车模的最前方，在车模失控之后跑出赛道撞击其它物体，最先遭到

破坏的就是电磁感应线圈。为了保护电磁感应线圈，同时使得它们能够在一条水平的直线上，可以采用硬质的塑料管将电磁线圈封装在一起。如下图所示：

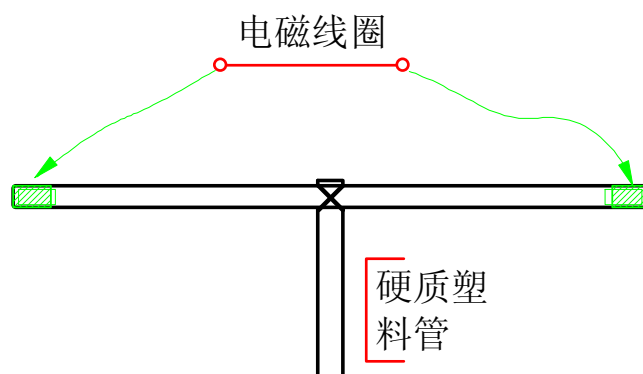


图 4-11 使用硬质塑料管封装的电磁感应线圈

如果电磁检测线圈多余两个，也可以考虑采用其它的支架固定方式，增加传感器支架的强度。

### (3) 车模倾角传感器

车模倾角传感器包括陀螺仪和加速度计。它们都是表贴元器件，一般固定在同一电路板上。建议将这块带有陀螺仪和加速度计的电路板固定在车模中间质心的位置，或者车模的底部，这样可以最大程度减少车模运行时前后振动对于测量倾角的干扰。

安装角度传感器电路板时应该尽量保证陀螺仪传感器水平安装。

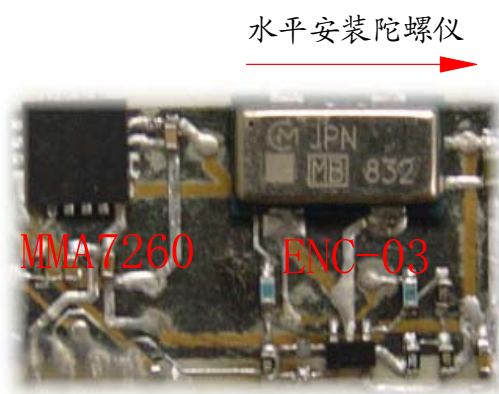


图 4-12 水平安装陀螺仪

如果陀螺仪安装不能够保证水平，则会影响车模过弯道时的速度。表现为车模在过弯道时速度变快或者变慢。分析车模在过弯道使得运动，车模在过弯道时同时具有两种

运动：平动和转动。其中转动会带动陀螺仪转动。如果陀螺仪安装不是绝对的水平，那么这个转动就会在陀螺仪的 Z 轴方向存在一个分量。根据陀螺仪倾斜的方向不同，这个分量有可能是正，有可能是负。从而会使得车模控制“仿佛感觉到在上坡或者是在下坡”，引起车模的速度变慢或者变快。

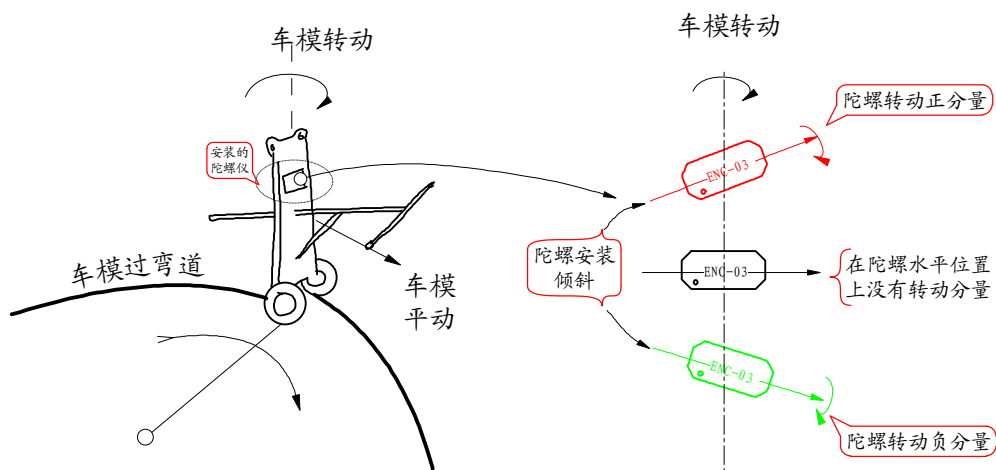


图 4-13 陀螺仪安装倾角影响车模过弯道

#### 4、车模转向速度陀螺仪

车模转向速度传感器是用来对方向进行微分控制，消除方向控制中的过冲现象。

该陀螺仪的旋转轴需要和陀螺仪的垂直纵轴保持一致。如果对车模转向不需要进行微分控制，那么这个传感器就可以省略。

### 4.3 注意事项

由于车模直立运行，在制作电路板的时候尽可能减少电路板的尺寸，一方面便于固定，另一方面可以减少车模的惯量。

固定电路板应尽可能贴近车模的底盘，使其能够稳固。

为了避免车模运行过程中倾倒，摔坏车模及其上的电路板，在车模机械设计的时候，需要考虑在车模前后安装有防撞支架或者缓冲物，一旦车模倾倒或者失控，防撞支架可以保护车模机械的安全性。

此外，也可以在防撞支架上安装车模跌倒检测开关，一旦车模倾倒，控制电路便立即停止运行。也可以使用无线遥控开关，可以使得车模紧急刹车，从而避免车模失控因撞击损坏。图 4-9 显示了参考设计方案测试车模在组装后的情况。

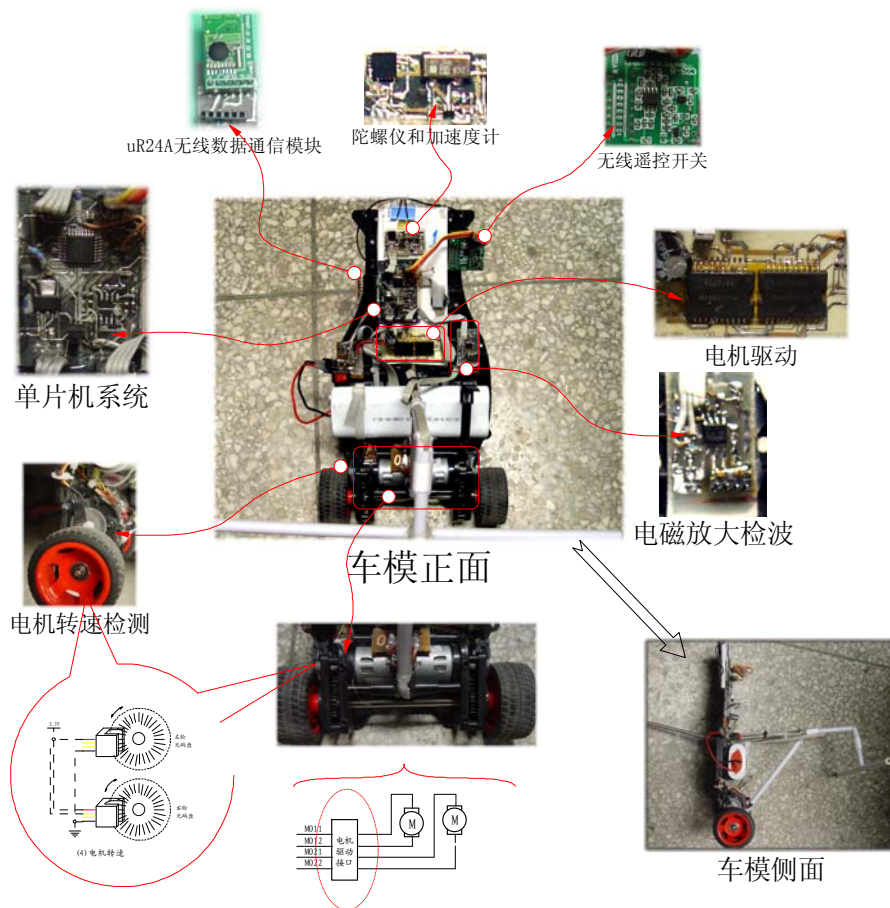


图 4-9 车模组装全图

## 第五章、软件开发篇

通过前面的介绍，车模控制电路制作与安装均已完毕。车模是否能够正常高速稳定运行，需要通过编写控制软件和车模调试来完成。开发软件主要任务包括：

- (1) 建立软件工程，配置 DSC 资源，初步编写程序的主框架；
- (2) 编写上位机监控软件，建立软件编译、下载、调试的环境；
- (3) 编写实现各个子模块，并测试各个子模块的功能正确性；
- (4) 通过程序初步调试，验证控制电路板的正确性。

开发飞思卡尔公司 56800/E 系列 DSC 单片机应用程序可以使用 CodeWarrior 集成开发环境（CodeWarrior for 56800/E Digital Signal Controllers，目前版本 v5.9）。与其它版本的 CodeWarrior 一样，DSC 版 CodeWarrior 也提供了 Processor Expert 功能模块，可以通过工程配置非常方便地生成单片机的各个外设的初始化代码和接口程序，帮助开发



者将精力集中在应用程序的开发上。

## 5.1 软件功能与框架

软件的主要功能包括有：

- (1) 各传感器信号的采集、处理；
- (2) 电机 PWM 输出；
- (3) 车模运行控制：直立控制、速度控制、方向控制；
- (4) 车模运行流程控制：程序初始化、车模启动与结束、车模状态监控；
- (5) 车模信息显示与参数设定：状态显示、上位机监控、参数设定等。

上述功能可以分成两大类：

第一类包括 1-3 功能，它们属于需要精确时间周期执行，因此可以在一个周期定时中断里完成。第二类包括 4-5 功能。它的执行不需要精确的时间周期。可以放在程序的主程序中完成。这两类任务之间可以通过全局变量实现相互的通讯。

主程序框架如**错误！未找到引用源。**所示。

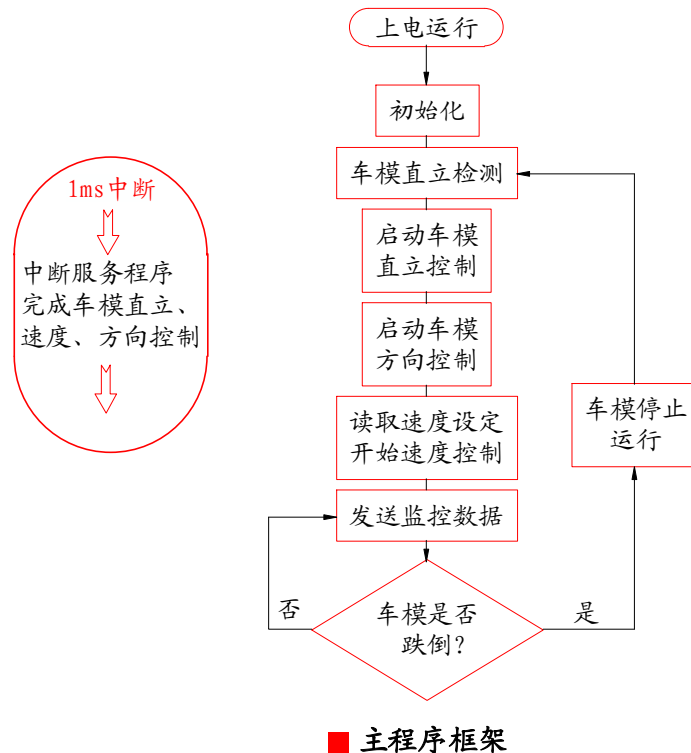


图 5-1 主程序框架

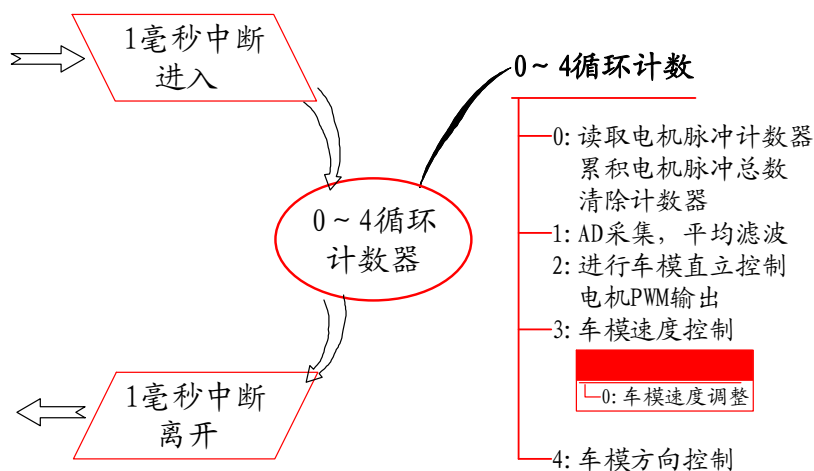
**错误！未找到引用源。**中，程序上电运行后，便进行单片机的初始化。初始化的工作包括有两部分，一部分是对于单片机各个应用到的模块进行初始化。这部分的代码由 CodeWarrior 集成环境的 ProcessorExpert 工具生成。第二部分是应用程序初始化，是对于车模控制程序中应用到的变量值进行初始化。

初始化完成后，首先进入车模直立检测子程序。该程序通过读取加速度计的数值判断车模是否处于直立状态。如果一旦处于直立状态则启动车模直立控制、方向控制以及速度控制。

程序在主循环中不停发送监控数据，在通过串口发送到上位机进行监控。同时检查车模是否跌倒。跌倒判断可以通过车模倾角是否超过一定范围进行确定，或者通过安装在车模前后防撞支架上的微动开关来判断。一定车模跌倒，则停止车模运行。包括车模直立控制、速度控制以及方向控制。然后重新进入车模直立判断过程。

车模的直立控制、速度控制以及方向控制都是在中断程序中完成。通过全局标志变量确定是否进行这些闭环控制。

中断程序框架如图 5.2 所示。



### ■ 1毫秒中断程序

图 5-2 中断服务程序

图 5-2 中，使用 DSC 的一个定时器，产生一毫秒的周期中断。中断服务程序的任务被均匀分配在 0-4 的中断片段中。因此每个中断片段中的任务执行的周期为 5 毫秒，频率为 200Hz。

将任务分配到不同的中断片段中，一方面防止这些任务累积执行时间超过 1 毫秒，

扰乱一毫秒中断的时序，同时也考虑到这些任务之间的时间先后顺序。当然也可以将所有的中断任务都合并在一起，使用一个 5 毫秒的中断来完成。

这些任务包括：

(1) 电机测速脉冲计数器读取与清除。累积电机转动角度。累积电机速度，为后面车模速度控制提供平均数；

(2) 启动 AD 转换。进行 20 次模拟量采集，然后计算各个通道的模拟量的平均值。这个过程是对于模拟信号进行低通滤波。

(3) 车模直立控制过程。包括车模角度计算、直立控制计算、电机 PWM 输出等。

(4) 车模速度控制：在这个时间片段中，又进行 0-19 计数。在其中第 0 片段中，进行速度 PID 调节。因此，速度调节的周期为 100 毫秒。也就是每秒钟调节 10 次。

(5) 车模方向控制：根据前面读取的电磁场检波数值，计算偏差数值。然后计算电机差模控制电压数值。

以上每个中断任务都占用不同的时间，在程序调试时，需要保证它们都不会超过 1 毫秒。每个任务具体执行时间可以通过单片机的端口显示出来。如下就是通过单片机的 LED 控制端口 A6，显示出每个任务执行的时间。在进入中断时，A6 输出高电平，推出中断时，A6 输出低电平。如图 5-3 中波形所示所示。

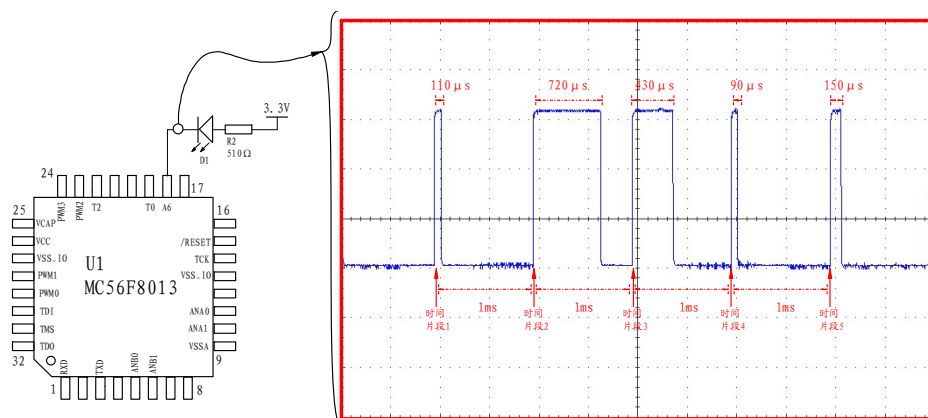


图 5-3 任务中断时间波形

从上面任务中断时间波形可以得到每个任务消耗的时间：

表 5-1 中断任务执行时间表

时间片段	任务执行时间 (微秒)	任务名称	备注
1	110	速度脉冲	

2	720	AD 转换	进行了 20 次了 6 个模拟通道采集并进行平均。
3	430	车模角度控制	
4	90	车模速度控制	速度控制周期是 100 毫秒。
5	150	车模方向控制	

在中断退出之后，程序执行主程序，完成车模运行和监控调试。

## 5.2 DSC的硬件资源配置

下面说明程序所用到 DSC 的模块资源及其配置。如果使用其它系列的处理器，可以参照这些配置进行设置。

### 1) 毫秒定时中断：TI1

硬件模块：TMR2\_Compare

中短周期：Period:1ms

触发事件：Event:Interrupt

### 2) 电机 P W M输出控制：PWMC1

硬件模块：PWM\_Timer

输出频率：Frequence:10Khz

输出模式：PWM0,1,2,3 Independent

死区时间：Dead Time:0

### 3) ADC 采集通道：ADC

硬件模块：ANA0,12,ANB0,1, 2

数值范围：Range:0 - 0x7ff0

转换时间：Conversion Time:1.594us

采集模式：Mode: Sequency

转换分辨率：Resolution:12bit

### 4) 电机测速脉冲计数器：Counter1

硬件模块：TMR0

计数范围：16bit

信号触发沿：Rising Edge

### 5) 电机测速脉冲计数器: Counter2

硬件模块: TMR1

计数范围: 16bit

信号触发沿: Rising Edge

### 6) 监控 UART 串口

硬件模块 SCI

通讯速率: Baud:117647 (实际值)

### 7) 读取 MMA8452Q 的 I2C 总线 (备用)

硬件模块: I2C

中断模式: Interrupt Service:Disable

I2C 地址: Address:4c

通讯速率: SCL Frequency:200kHz

注: 这个 I2C 总线控制器用在读取飞思卡尔公司数字接口的加速度传感器上。

注意: 上面的这些配置主要是通过 CodeWarrior 的 Processor Expert (PE) 来完成, PE 会自动生成这些模块的初始化程序代码, 以及接口程序代码。工程文件中的目录与文件说明参见附录。

## 5.3 主要算法及其实现

在设计和编写控制算法软件之前, 请先确认自己的控制方案。对于初学者可以研读在附录 4 给出的软件算法全图, 根据自己的理解进行裁剪和修改控制方案。同一个控制算法在软件实现的时候由于采取的数据结构不同, 实现的 C 语言的代码也各有差异。初学者可以参考本节所提供的软件函数中的代码, 在此基础上实现和优化自己的代码。

### 5.3.1 算法框图与控制函数关系

完整的软件算法框图参见附录 4。其中与控制相关的算法见下图。该图中将各部分算法与软件中的控制函数之间的关系列写出, 帮助大家理解相关的算法。

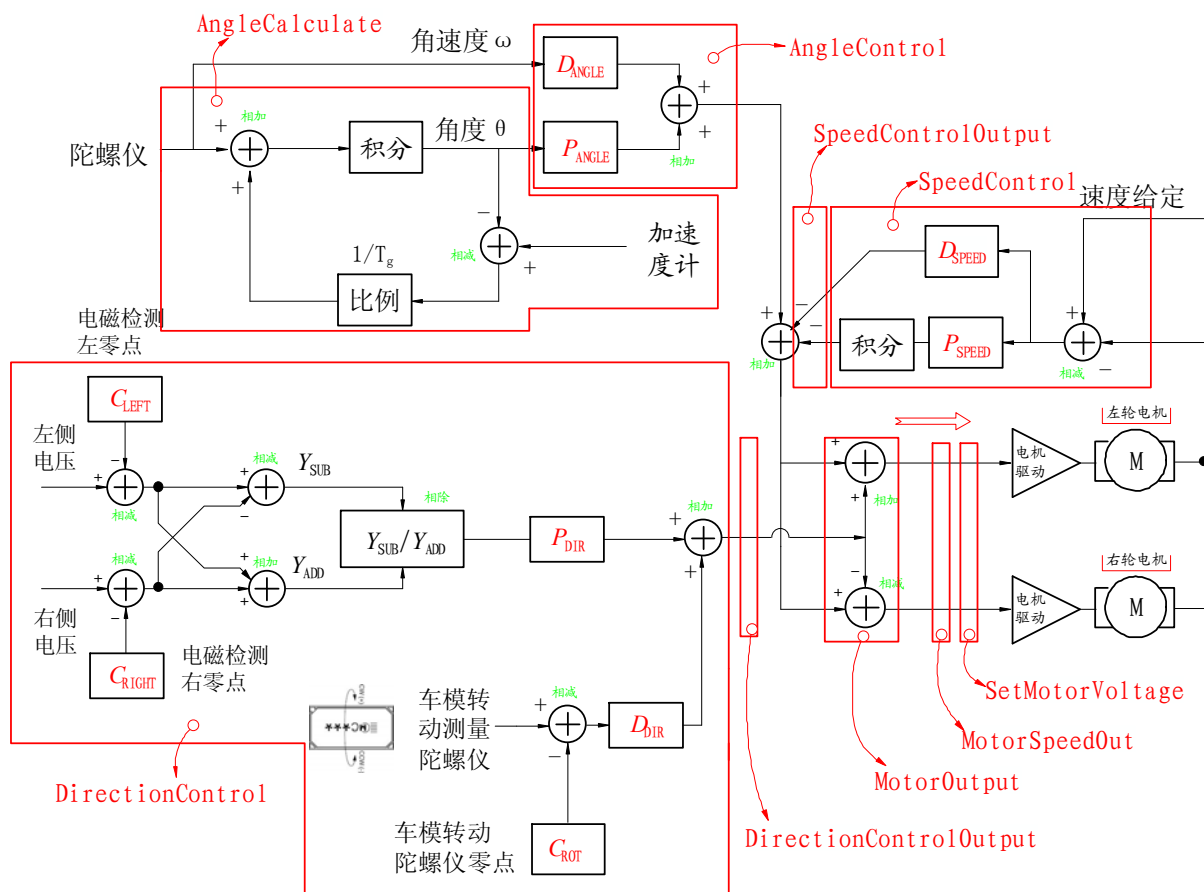


图 5-4 算法框图中与控制相关的软件函数

图 5-4 与控制相关的软件函数包括：

1. **AngleCalculate:** 车模倾角计算函数。根据采集到的陀螺仪和重力加速度传感器的数值计算车模角度和角速度。如果这部分的算法由外部一个运放实现，那么采集得到的直接是车模的角度和角速度，这部分算法可以省略。该函数是每 5 毫秒调用一次。
2. **AngelControl:** 车模直立控制函数。根据车模角度和角速度计算车模电机的控制量。直立控制是 5 毫秒调用一次。
3. **SpeedControl:** 车模速度控制函数。根据车模采集到的电机转速和速度设定值，计算电机的控制量。该函数是 100 毫秒调用一次。
4. **SpeedControlOutput:** 速度输出平滑函数。由于速度是每 100 毫秒进行一次计算。为了使得速度控制更加平滑，该函数将速度输出变化量平均分配到 20 步 5 毫秒的控制周期中。
5. **DirectionControl:** 方向控制函数。根据车模采集到的左右两个电磁传感器的数值计算出角度控制的量。该函数每 10 毫秒调用一次。

6. **DirectionControlOutput**: 方向控制函数输出平滑函数。将方向控制的输出变化量平均分配到 2 步 5 毫秒的控制周期中。
7. **MotorOutput**: 电机输出量汇集函数。根据前面的直立控制、速度控制和方向控制所得到的控制量进行叠加，分别得到左右两个电机的输出电压控制量。对叠加后的输出量进行饱和处理。函数调用周期 5 毫秒。在此请大家注意速度控制量叠加的极性是负。
8. **MotorSpeedOut**: 电机 PWM 输出计算函数。根据左右两个电机的输出控制量的正负极性，叠加上一个小的死区数值，克服车模机械静态摩擦力。函数调用周期 5 毫秒。
9. **SetMotorVoltage**: PWM 输出函数：根据两个电机的输出量，计算出 PWM 控制寄存器的数值，设置四个 PWM 控制寄存器的数值。函数调用周期 1 毫秒。

以上 9 个函数都是在 1 毫秒中断服务中进行被相互调用的。下图显示了这些函数之间的调用与参数传递关系。在个函数附近也表明了调用的周期。

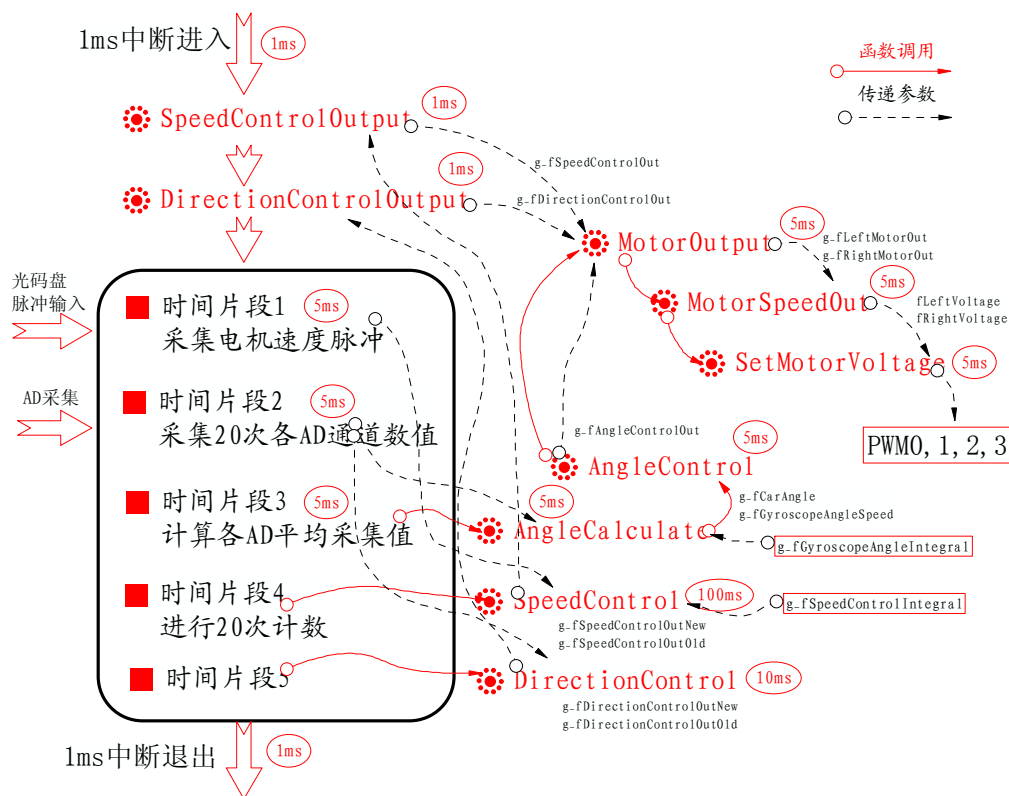


图 5-5 控制函数调用与参数传递关系

上面函数之间的参数传递有的是通过函数调用参数变量，有的是通过全局变量传递的。全部的参数的类型都是单精度浮点数(float)。AngleCalculate, SpeedControl 中存在

积分计算，这两个函数还保留两个全局积分变量。在时间片段 1,2 中采集到的速度脉冲信息和模拟量信息经过累积平均后，为后面的角度、速度和方向控制提供车模的状态变量。

为了方便阅读后面的代码，下面将程序中定义的变量的命名规范进行简单地介绍，如图 5-6 所示。

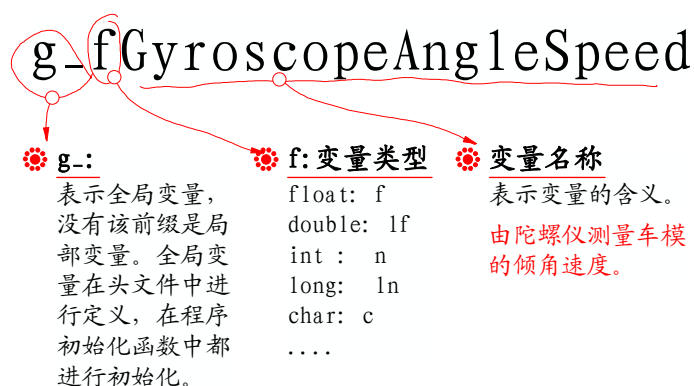


图 5-6 程序中变量命名规范

理解和遵循一定的变量命名规范，使得程序代码具有“自明”特性，书写程序就可以减少注释的工作，从而可以避免混淆。在后面程序介绍中，每段代码都与控制算法框图进行了一一对应，便于控制算法理解和熟悉，帮助学生阅读代码，因此程序代码无需任何注释都应该是非常清楚。

### 5.3.2 主要控制函数说明

本节将对主要控制函数进行说明。函数代码均采用通用的 C 语言，单精度浮点运算。方便初学者进行理解和移植。

#### 1. AngleCalculate: 角度计算函数

本程序读取 DSC 的 AD 采用数值，然后计算车模的倾角。

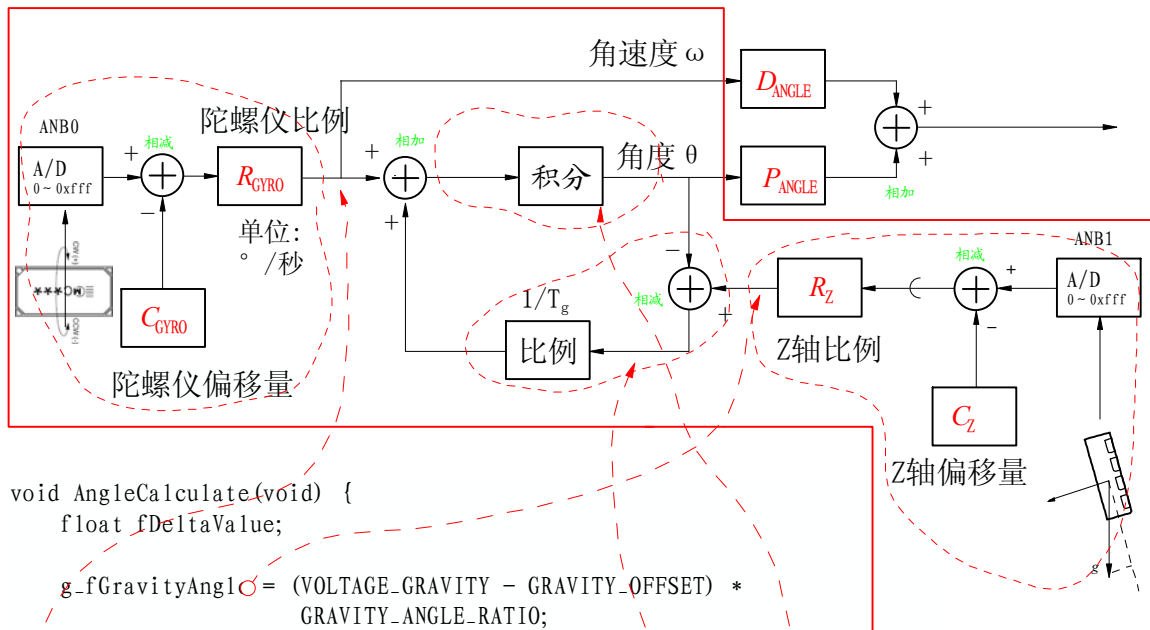
对于读取的陀螺仪和加速度计的数值需要减去零偏值。这个数值需要通过实验确定。在车模保持直立静止时，读出两个通道的数值，便是相应的零偏值。这个数值会带有一定的误差，往往会使得车模往一个方向加速行驶。这个误差可以通过后面的速度控制加以消除。

对于加速度计所得到的数值，通过一个比例系数(GRAVITY\_ANGLE\_RATIO)将它归一化到-90 至 90 之间，这个宏定义请大家阅读工程文件中的头文件。同样，后面的陀



螺仪的读出的数据也需要通过一个比例系数进行归一化。这个数值也是通过实验确定的。具体的方式为参见后面的调试过程。

由于本函数调用的时间周期为 5ms。详细程序请参见程序-1 代码：



```

void AngleCalculate(void) {
    float fDeltaValue;

    g_fGravityAngle = (VOLTAGE_GRAVITY - GRAVITY_OFFSET) *
        GRAVITY_ANGLE_RATIO;
    // g_fGravityAngle = 0;

    g_fGyroscopeAngleSpeed = (VOLTAGE_GYRO - GYROSCOPE_OFFSET) *
        GYROSCOPE_ANGLE_RATIO;

    g_fCarAngle = g_fGyroscopeAngleIntegral;
    fDeltaValue = (g_fGravityAngle - g_fCarAngle) /
        GRAVITY_ADJUST_TIME_CONSTANT;

    g_fGyroscopeAngleIntegral += (g_fGyroscopeAngleSpeed + fDeltaValue) /
        GYROSCOPE_ANGLE_SIGMA_FREQUENCY;
}

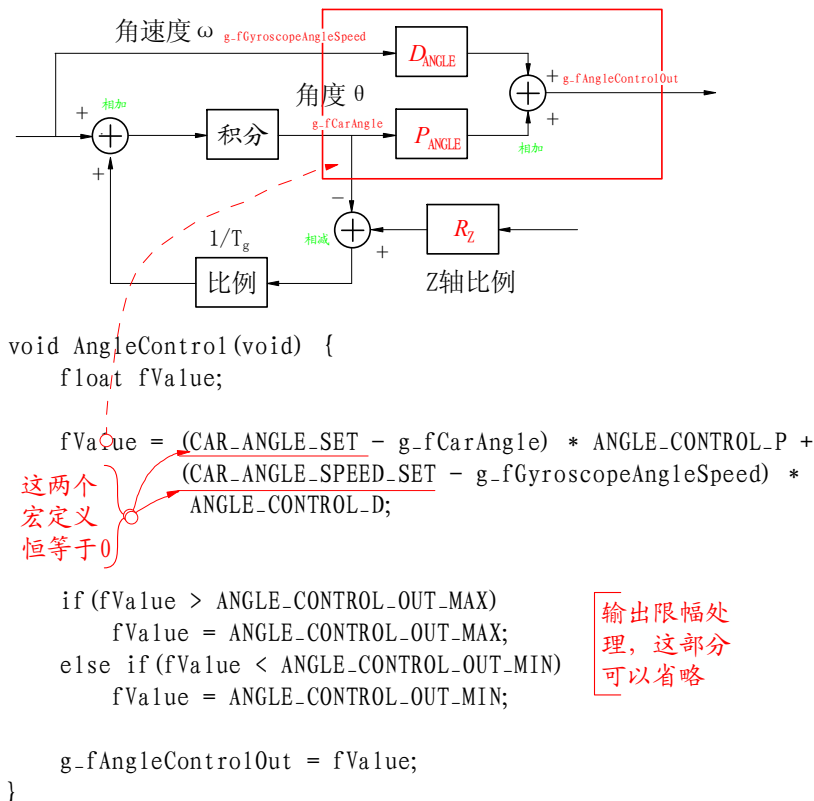
```

程序-1 角度计算函数

值得说明的是角度计算函数可以由硬件来完成，请参阅图 3-25 简化的角度和角速度处理电路，这个电路可以帮助大家进一步理解角度处理算法。如果实际电路采用了图 3-25 的方案，这个角度计算函数就可以省略，直接通过采集就可以得到相应的车模角度和角速度。

## 2、AngleControl:直立控制函数

将车模角度和角速度乘以各自相应的系数就可以得到直立控制输出量。算法参见程序-2 的内容。



程序-2 直立控制函数

在这个函数中有两个宏定义 `CAR_ANGLE_SET`, `CAR_ANGLE_SPEED_SET` 都在头文件中定义成 0。

角度控制输出量限幅部分的代码可以去掉。

### 3、SpeedControl:速度控制函数

速度控制周期为 100 毫秒。读取两个电极的光码盘脉冲进行累加，利用 100 毫秒的脉冲数量可以反映车模的电机的转速。车模速度采用两个电极速度的平均值。

为了使得速度具有一定的物理含义，对于所取得的电机速度值需要进行单位转换。根据定义单位的物理含义，可以确定速度单位转换的比例值。在本程序中，定义速度的单位是电机每秒钟的转动圈数。因此单位比例值(`CAR_SPEED_CONSTANT`)定义为：

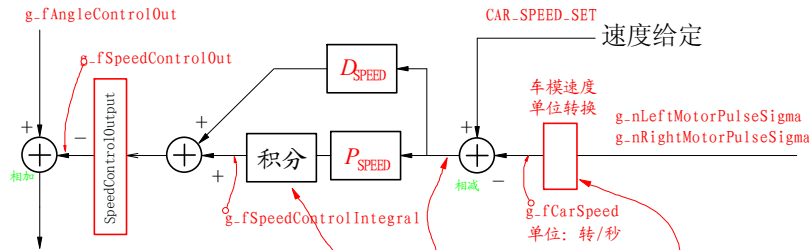
```

#define OPTICAL_ENCODE_CONSTANT 100 //光电编码盘的刻槽数量
#define SPEED_CONTROL_PERIOD 100 //速度控制周期，ms
#define CAR_SPEED_CONSTANT // 单位转化比例值
1000.0/SPEED_CONTROL_PERIOD/OPTICAL_ENCODE_CONSTANT

```

为了减少速度控制对于车模直立控制的影响，速度控制输出量的变化被函数

SpeedControlOut 均匀分配在 20 个角度控制函数周期内，这样可以保证车模的稳定性。



```

void SpeedControl(void) {
    float fDelta;
    float fP, fI;

    g_fCarSpeed = (g_nLeftMotorPulseSigma +
                  g_nRightMotorPulseSigma) / 2;
    g_nLeftMotorPulseSigma = g_nRightMotorPulseSigma = 0;
    g_fCarSpeed *= CAR_SPEED_CONSTANT;

    fDelta = CAR_SPEED_SET - g_fCarSpeed;
    fP = fDelta * SPEED_CONTROL_P;
    fI = fDelta * SPEED_CONTROL_I;

    g_fSpeedControlIntegral += fI;

    g_fSpeedControlOutOld = g_fSpeedControlOutNew;
    g_fSpeedControlOutNew = fP + g_fSpeedControlIntegral;
}

void SpeedControlOutput(void) {
    float fValue;
    fValue = g_fSpeedControlOutNew - g_fSpeedControlOutOld;
    g_fSpeedControlOut = fValue * (g_nSpeedControlPeriod + 1) /
        SPEED_CONTROL_PERIOD +
        g_fSpeedControlOutOld;
}

```

计算两个电极脉冲数的平均值，再进行单位转换。单位：转/秒

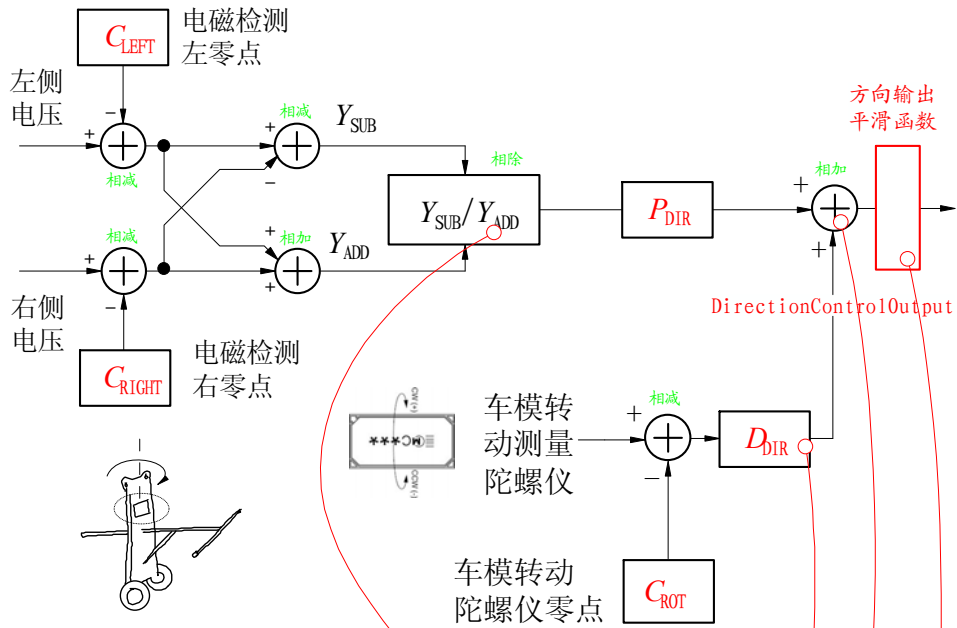
该函数将速度输出变化量平均到20个控制周期内。

程序-3 速度控制函数

#### 4、DirectionControl:方向控制函数

车模方向控制利用左右两个感应电磁线圈的电压差值来计算车模电机差值驱动电压。根据前面控制原理介绍，具体程序实现参见程序-4。该函数调用周期为 10ms。

同样为了减少对直立控制的影像，方向控制量的变化也被均匀分配在 2 个直立控制周期内。通过函数 DirectionControlOut 来实现。



```

void DirectionControl(void) {
    float fLeftRightAdd, fLeftRightSub, fValue;
    float fDValue;
    int nLeft, nRight;

    nLeft = (int) (g_fLeftVoltageSigma /= DIRECTION_CONTROL_COUNT);
    nRight = (int) (g_fRightVoltageSigma /= DIRECTION_CONTROL_COUNT);
    g_fLeftVoltageSigma = 0;
    g_fRightVoltageSigma = 0;

    fLeftRightAdd = nLeft + nRight;
    fLeftRightSub = nLeft - nRight;

    g_fDirectionControlOutOld = g_fDirectionControlOutNew;

    if (fLeftRightAdd < LEFT_RIGHT_MINIMUM) {
        g_fDirectionControlOutNew = 0;
    } else {
        fValue = fLeftRightSub * DIR_CONTROL_P / fLeftRightAdd;

        fDValue = DIR_CONTROL_D.VALUE - DIRECTION_OFFSET;
        fDValue *= DIR_CONTROL_D;

        fValue += fDValue;

        g_fDirectionControlOutNew = fValue;
    }
}

void DirectionControlOutput(void) {
    float fValue;
    fValue = g_fDirectionControlOutNew - g_fDirectionControlOutOld;
    g_fDirectionControlOut = fValue * (g_nDirectionControlPeriod + 1) /
        DIRECTION_CONTROL_PERIOD + g_fDirectionControlOutOld;
}

```

● g\_fLeftVoltageSigma  
g\_fRightVoltageSigma  
已经在AD采集通道中减去了  
零点偏移值，同时进行了  
累加平均

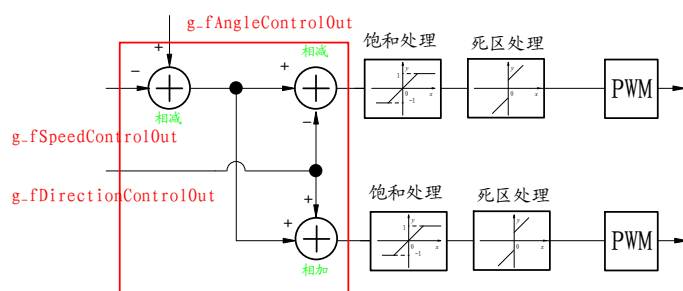
{ 如果左右电磁检测不到  
感应电压，方向控制量为0

程序-4 车模方向控制

### 5、MotorOut:电机输出函数

电机输出函数非常简单，只是将前面直立控制、速度控制、方向控制的输出量进行

叠加形成左右两个电机的驱动控制量。参见程序-5 电机输出函数。



```

void MotorOutput (void) {
    float fLeft, fRight;

    fLeft = g_fAngleControlOut -
            g_fSpeedControlOut -
            g_fDirectionControlOut;
    fRight = g_fAngleControlOut -
            g_fSpeedControlOut +
            g_fDirectionControlOut;

    g_fLeftMotorOut = fLeft;
    g_fRightMotorOut = fRight;
    MotorSpeedOut ();
}

```

程序-5 电机输出函数

在该函数中需要注意的是各个控制量的极性。不同的物理坐标系定义决定了前面各个控制量在叠加中的正负关系。比如定义车模的速度往前为正，车模角度往后倾角为正，车模旋转角度逆时针旋转为正。在车模控制过程中应该坚持统一的物理坐标系的定义。在图 5-3 图，规定了一种车模运动参数的坐标系，如果参照了图 5-3 运动坐标系的定义，在上述函数中，控制速度的输出量的极性是“负”，与角度控制的量的极性相反。

## 6、MotorSpeedOut:电机 PWM 计算函数

该函数是对左右两个电极的输出量增加死区常量和输出饱和处理。

由于车模传动机构存在这静态摩擦力，增加一个死区常量可以提高车模在静态下的稳定性。这个数值需要通过实验来确定。图 5-7 显示了死区补偿的关系。

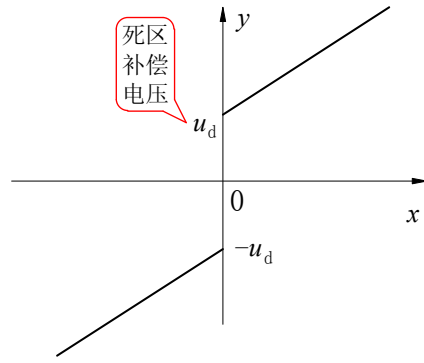
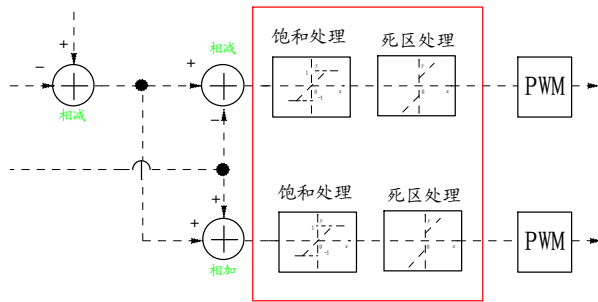


图 5-7 电机死区补偿

输出饱和处理是保证输出量不会超出 PWM 的满量程范围。

改程序调用周期为 5 毫秒，参见程序 5-6.



```

void MotorSpeedOut (void) {
    float fLeftVal, fRightVal;

    fLeftVal = g_fLeftMotorOut;
    fRightVal = g_fRightMotorOut;

    if (fLeftVal > 0)
        fLeftVal += MOTOR_OUT_DEAD_VAL;
    else if (fLeftVal < 0)
        fLeftVal -= MOTOR_OUT_DEAD_VAL;

    if (fRightVal > 0)
        fRightVal += MOTOR_OUT_DEAD_VAL;
    else if (fRightVal < 0)
        fRightVal -= MOTOR_OUT_DEAD_VAL;

    if (fLeftVal > MOTOR_OUT_MAX)
        fLeftVal = MOTOR_OUT_MAX;
    if (fLeftVal < MOTOR_OUT_MIN)
        fLeftVal = MOTOR_OUT_MIN;
    if (fRightVal > MOTOR_OUT_MAX)
        fRightVal = MOTOR_OUT_MAX;
    if (fRightVal < MOTOR_OUT_MIN)
        fRightVal = MOTOR_OUT_MIN;

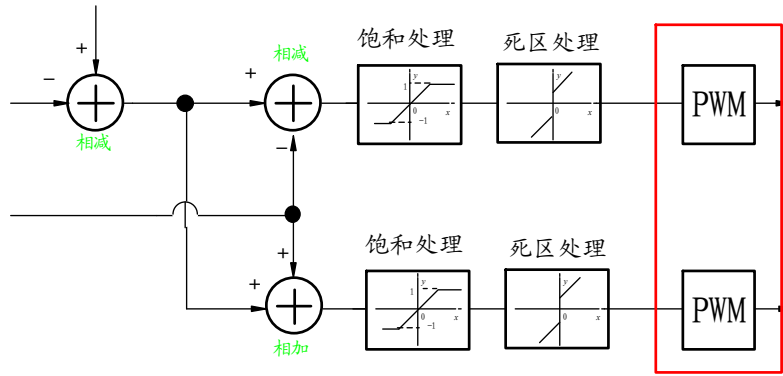
    SetMotorVoltage (fLeftVal, fRightVal);
}

```

程序-6 电机 PWM 计算函数

## 7、SetMotorVoltage:PWM 输出设置函数

最终电机驱动信号是通过单片机的 PWM 模块产生的。本函数计算两路电机的 PWM 输出占空比，并设置 PWM 模块的寄存器。



```

void SetMotorVoltage(float fLeftVoltage, float fRightVoltage) {
    short nPeriod;
    int nOut;

    nPeriod = (short) getReg (PWM_PWMCM);

    if(fLeftVoltage > 0) {
        setReg (PWM_PWMVAL1, 0);
        nOut = (int) (fLeftVoltage * nPeriod);
        setReg (PWM_PWMVAL0, nOut);
    } else {
        setReg (PWM_PWMVAL0, 0);
        fLeftVoltage = -fLeftVoltage;
        nOut = (int) (fLeftVoltage * nPeriod);
        setReg (PWM_PWMVAL1, nOut);
    }

    if(fRightVoltage > 0) {
        setReg (PWM_PWMVAL2, 0);
        nOut = (int) (fRightVoltage * nPeriod);
        setReg (PWM_PWMVAL3, nOut);
    } else {
        setReg (PWM_PWMVAL3, 0);
        fRightVoltage = -fRightVoltage;
        nOut = (int) (fRightVoltage * nPeriod);
        setReg (PWM_PWMVAL2, nOut);
    }

    MOTOR_SETLOAD; // Reload the PWM value
}

```

程序-7 PWM 输出函数

## 8、中断服务程序

中断服务程序完成了车模的角度、速度和方向控制的周期调用。请参见图 5-5 阅读以下程序。



```

void TI1_0nInterrupt (void) {
    int i;

    g_nSpeedControlPeriod ++;
    SpeedControlOutput ();

    g_nDirectionControlPeriod ++;
    DirectionControlOutput ();

    if(g_n1MSEventCount >= CONTROL-PERIOD) {
        g_n1MSEventCount = 0;
        GetMotorPulse ();    读取两个电极脉冲计数值
    } else if(g_n1MSEventCount == 1) {
        for(i = 0; i < INPUT-VOLTAGE-AVERAGE; i ++)
            SampleInputVoltage ();
    } else if(g_n1MSEventCount == 2) {
        GetInputVoltageAverage ();
        AngleCalculate ();
        AngleControl ();
        MotorOutput ();
    } else if(g_n1MSEventCount == 3) {
        g_nSpeedControlCount ++;
        if(g_nSpeedControlCount >= SPEED-CONTROL-COUNT) {
            SpeedControl ();
            g_nSpeedControlCount = 0;
            g_nSpeedControlPeriod = 0;
        }
    } else if(g_n1MSEventCount == 4) {
        g_nDirectionControlCount ++;
        DirectionVoltageSigma ();
        if(g_nDirectionControlCount >= DIRECTION-CONTROL-COUNT) {
            DirectionControl ();
            g_nDirectionControlCount = 0;
            g_nDirectionControlPeriod = 0;
        }
    }
}

```

程序-8 中断服务程序

为了使得核心控制函数容易理解，在上面给出的所有函数代码中，都将与控制算法无关的语句进行了适当的删减，详细的代码可以参阅附件给出的 DSC 工程文件中 CarSub.C, .H 文件。大家对于核心控制算法研究，会发现电磁组直立车模的算法相对还是非常简洁的。

实现了核心算法只是为车模直立行走打下了基础。车模运行是否稳定高速还需要进行参数的整定与调试。软件代码中还应该包括为了车模调试的相关程序。比如车模运行

状态的显示，与调试上位机之间的通讯，车模运行异常的检测与保护，程序参数的设置与存储等等。这些辅助程序是否完善，直接影响到后期的车模调试的准确性和效率。这部分的内容可以参见秘书处给出的 DSC 工程文件的相关函数，在这儿就不一一介绍了。

## 第六章、车模调试篇

车模制作与软件开发为车模运行打下了基础，是否运行的稳定和快速需要通过车模精心的调试才能够完成。调试过程同时也可以使得同学们加深对于控制算法物理过程深刻理解，提高解决实际工程问题的能力，为寻找更加优化的车模设计方案打下基础。调试过程对于竞赛车模制作非常重要。

车模调试分为调试准备、静态参数整定、动态参数整定、车模机械调整和竞赛策略制定等各个环节。整个调试中涉及到的参数和部件非常多，而且这些参数之间还有着紧密的相互影响。如果对于其中的物理过程认识不清，有没有正确的调试步骤，那么在调试过程中出现的各种错误现象就会掩盖正确的原因，影响整个调试进程，甚至会动摇制作的信心。

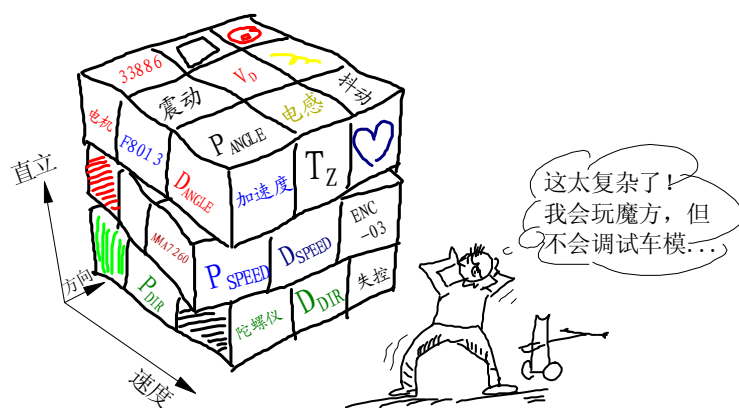


图 6-1 调试车模参数复杂而关键

本节将介绍车模参数调试的步骤作为参考。

### 6.1 调试参数

前面给出的算法程序存在很多参数，虽然从理论上可以对这些参数进行优化计算。但是由于受到车模模型精度的影响，计算所得到的参数也只能作为参考值——调试的

起始范围。实际优化参数需要通过一定的工程步骤最终确定，这个过程称为参数整定。

首先确定需要整定的车模控制参数。在附录 5 给出的控制算法框图中，将所有需要整定的参数都标记为红色。这些参数包括两大类：

### 1、传感器参数：

- $R_{GYRO}$ ：陀螺仪比例。将陀螺仪 AD 采集数值转换成角速度信号，单位是  $^{\circ}/\text{秒}$ 。
- $C_{GYRO}$ ：陀螺仪零点偏移量。陀螺仪静之时 AD 采集量。
- $C_Z$ ：重力加速度 Z 轴零点偏移量。重力加速度传感器在车模垂直静止时的 AD 采集量。
- $Z_{MAX}$ ：重力加速度 Z 轴最大采集值。在车模反面朝上时 Z 轴 AD 采集值。
- $Z_{MIN}$ ：重力加速度 Z 轴最小采集值。在车模正面朝上时 Z 轴 AD 采集值。
- $C_{LEFT}$ ：左边电磁感应线圈零点偏移量。在外部 20kHz 电源关闭时，左边电磁感应线圈通道的 AD 采集值。
- $C_{RIGHT}$ ：右边电磁感应线圈零点偏移量。在外部 20kHz 电源关闭时，右边电磁感应线圈通道的 AD 采集值。
- $C_{DIR}$ ：车模转动速度陀螺仪零点偏移值。在车模静止时，测量车模转动速度陀螺仪的 AD 采集值。

### 2、运动控制参数：

- $P_{ANGLE}$ ：角度比例控制参数。
- $D_{ANGLE}$ ：角度微分控制参数。
- $P_{SPEED}$ ：速度比例控制参数。
- $D_{SPEED}$ ：速度微分控制参数。
- $P_{DIR}$ ：方向比例控制参数。
- $D_{DIR}$ ：方向微分控制参数。
- $T_Z$ ：重力加速度补偿时间常数。。

- $V_D$ : 电机输出死区参数。

标定这些参数需要考虑输入输出参数的单位量纲。下图给出了各个参数的输入输出单位。

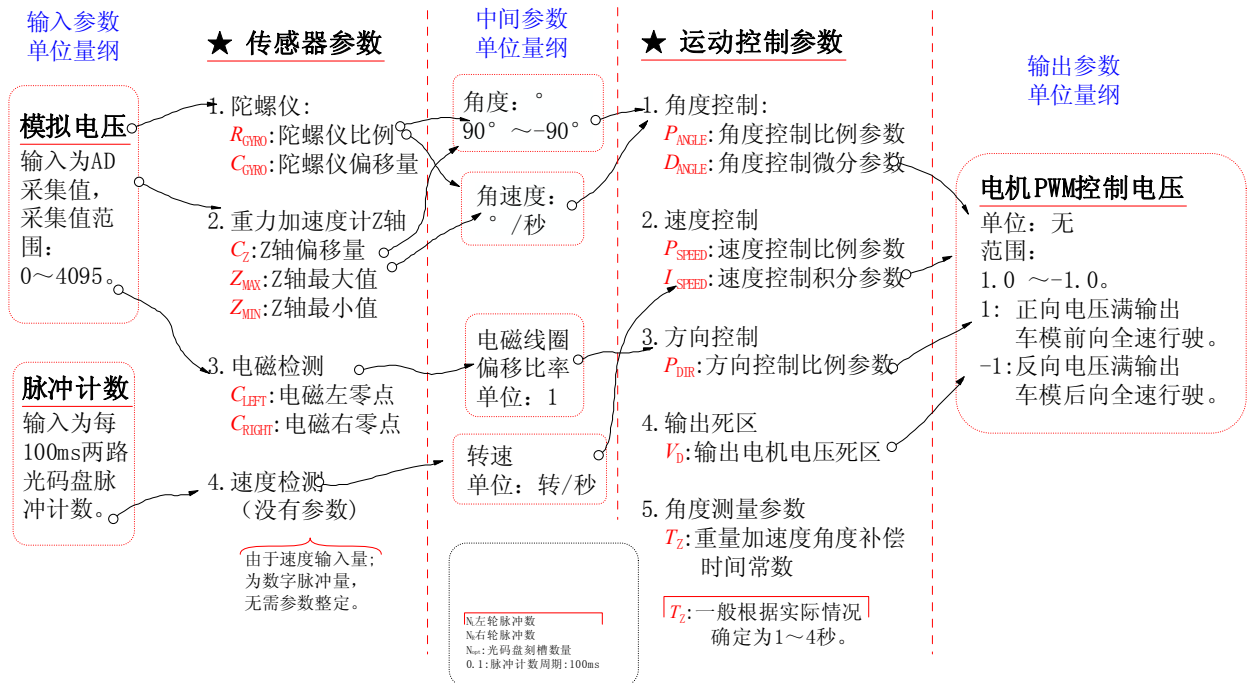


图 6-2 需要调整的参数和相关的单位

如果设计的方案中的单位和上述的单位不一致, 相应的参数需要进行单位换算。比如电机的 PWM 控制电压的数值范围是-128~127, 那么相应的运动控制参数的取值都需要扩大 128 倍左右。

各个参数的极性 (正负号) 需要考虑车模运动坐标系的定义。车模运动包括三种运动: 车模平动、车模仰动和车模转动。

车模正面方向规定为车模平动的正方向。车模往后倾斜规定为车模仰动的正方向。车模逆时针旋转规定为车模转动的正方向。如图 6-3 所示。

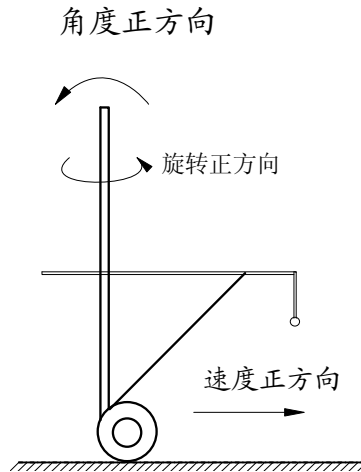


图 6-3 车模运动坐标定义

当车模的坐标规定与图 6-3 不同时，相应的参数的极性需要改变。

## 6.2 调试条件

车模在安装组装完毕之后，正式调试之前需要准备一下条件：

- 1、车模电路初步调试：控制电路可以正确上电，单片机可以下载程序。
- 2、车模调试环境：进行车模调试必须的场地和辅助的 20kHz 的电源。
- 3、车模调试辅助硬件和软件：车模运动保护和参数监控软件。

上述条件可以在车模开发制作、调试过程中逐步进行完善。下面分别介绍这些调试条件的要求和准备情况。

### 6.2.1 车模电路初步调试

车模电路进行初步调试为后面的车模软件参数打下基础。基本包括五个步骤，分别进行介绍。：

#### 1、上电检查电源

电路加上 7.2V 电池电压之后，测量电路的工作电流和各个稳压电路的输出值是否在安全范围之内。

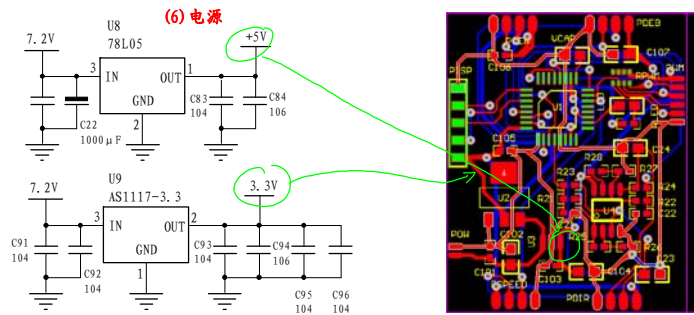


图 6-4 电源检查

参考设计电路中包括有 7.2V, 3.3V, 5V 三种电源。在静态时，总的工作电源应该在 150mA 左右。

## 2、单片机程序下载和通信：

DSC 56F8013 具有在系统编程(ISP)能力。该单片机在出场后内部就有一个自举编程 (Bootloader)程序，通过 UART 串口便可以将用户程序下载下去。如果第一次下载下去的程序本身又具有自举的功能，将来所有的程序下载便可以通过 UART 进行。详细的介绍请参见 Freescale 公司网站提供的资料。UART 不仅可以提高程序的下载，同时在上位串口监控程序的协助下来显示程序运行过程中发送的串口字符，便于调试程序。建议 F8013 的 Bootloader 下载功能和串口监控联合开发成一个上位机软件，由此便可以通过串口完成整个车模程序的开发与调试。

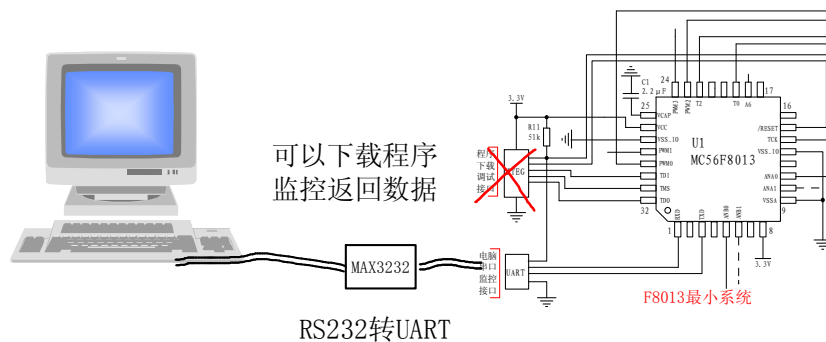


图 6-5 单片机串口通信

单片机开发也可以借助于 JTEG 等开发工具来实现。

## 3、PWM 输出

在单片机软件可以进行开发下载基础之上，编写单片机 PWM 输出控制程序。参见 SetMotorVoltage 函数代码。通过控制电机转速，验证驱动电路的正确性。确定两个电极

输出电压的极性。保证输出 PWM 正电压时，电机带动车模前行；PWM 输出负电压时，车模后行。

输出PWM驱动电机运行。调整电机接口极性，保证输出正电压，车模前行；输出负电压，车模后行。



图 6- 6 PWM 信号输出

#### 4、光码盘脉冲采集

通过示波器观察电机转动时光码盘的输出波形，确定波形的幅度和频率。编写脉冲采集程序并进行显示。

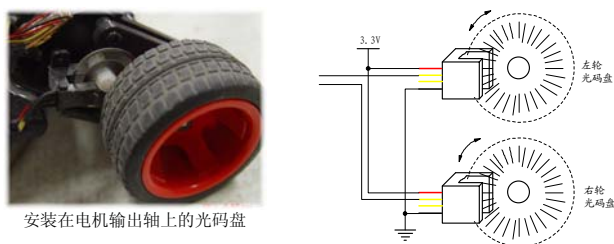


图 6- 7 采集电机光电码盘信号

利用 PWM 信号的输出驱动电机转动，获得电机脉冲数量。通过以上两个步骤，可以编写一个电机恒速运行的程序。通过 PID 的调节，是的电机在不同的外面阻力的情况下，都能够保持恒定转速。通过这个练习可以锻炼 DSC 程序的开发调试的能力，并且进一步验证车模输入输出通道的正确性。

#### 5、模拟量采集

F8013 具有六个通道的 AD 转换。参考设计方案应用了其中五个通道。编写 AD 转换程序，将采集的数值发送给上微机进行显示。通过观察各通道的采集值，确认各传感器工作状态，信号动态范围以及各通道噪声的影响。最好能够将各通道采集的数值通过曲线显示，可以比较直观的观察信号的变化和噪声的幅度。如下便是通过上微机显示的陀螺仪和加速度计的采集波形。

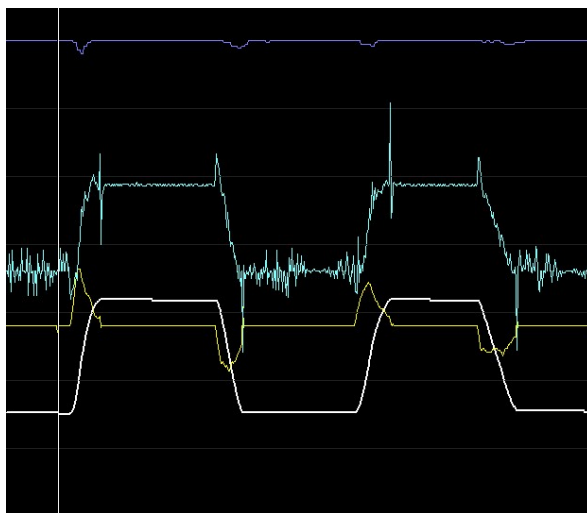


图 6-8 陀螺仪、加速度传感器 AD 采集信号

每个模拟量都会具有不同比例的噪声，需要对每个采集量进行低通滤波。简单的方法就是通过多次采集进行平均来实现。图 6-8 显示的波形数据是通过 20 次采集平均后的数据。

## 6.2.2 车模调试环境

### 1、静态参数调试桌面

车模静态参数调整可以在桌面完成。因此需要准备铺有 20kHz 电磁导引线的调试桌面。

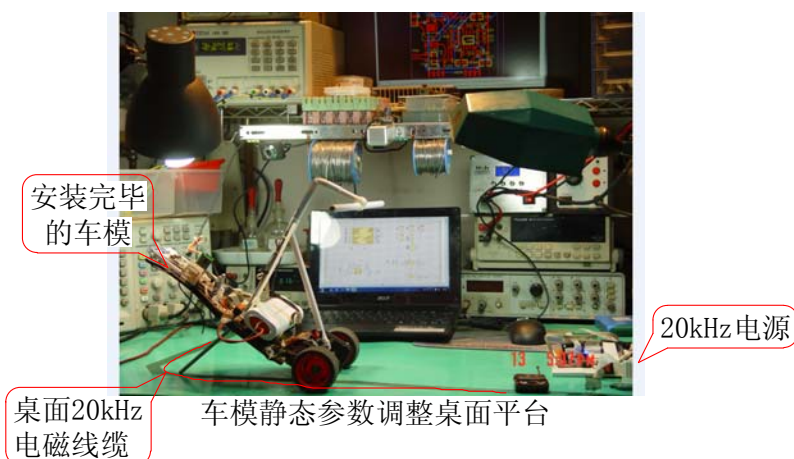


图 6-9 车模静态参数调整桌面

### 2、动态参数调试场地

在车模运行过程对各个控制参数进行优化组合，同时对车模的机械结构和传感器安



装进行调整。需要准备带有电磁线缆的跑道。跑道的形状和复杂程度在比赛规定的范围内，尽量设计有丰富的赛道元素，其中至少包括由十字交叉路口和各种弯道。



图 6-10 车模动态参数调试场地

### 6.2.3 车模调试辅助硬件和软件

俗话说，欲善其功，必先利其器。良好的辅助调试工具可以大大加快调试过程，能够给出现的故障准确定位。

#### 1、参数设定与状态监控软件

为了保证调试顺利，一般需要配合上位机串口监控程序，能够实时显示程序运行采集到的各种数据，通过曲线或者数字显示出来，帮助确定一些待定参数，判断程序 BUG，加快程序调试，确定控制参数的优化数值。

监控软件功能包括：

- 1) 显示车模实时返回的参数；
- 2) 设定车模各种运行参数；
- 3) 具备 Bootloader 下载车模程序。

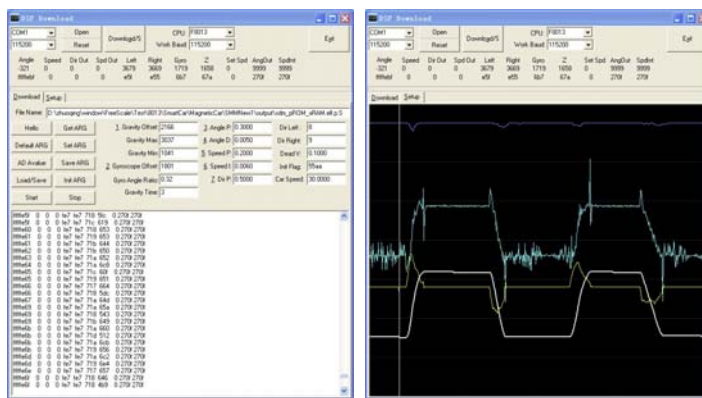
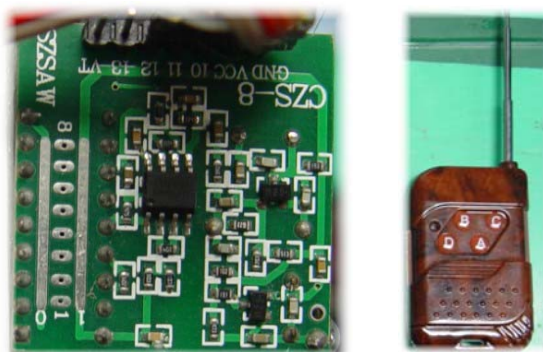


图 6-11 监控软件界面

监控软件可以选择自己熟悉的 Windows 开发工具进行开发。

## 2、无线遥控开关

车模在初期调试过程中容易冲出赛道。为了避免车模因撞击而损坏，在车模上安装一个无线遥控开关，能够及时停止车模运行，避免车模冲出赛道撞击损坏。



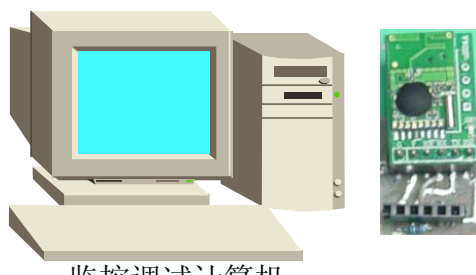
无线遥控开关

图 6-12 无线遥控开关

无线遥控开关可以直接控制车模电机驱动电源，也可以给控制单片机一个开关状态信号，由单片机检测该信号启动或停止运行。

## 3、无线通信模块

现在可以选择的无线串口通信模块非常多，应用无线通信模块非常方便对于动态运行的车模参数进行监控。同时也可以动态改变车模的各种运动控制参数。



无线通信模块进行参数监控

图 6-13 无线通信模块进行参数监控

车模运行状态主要是有个控制参数决定的。因此将车模的全部运动参数定义成一个数据结构存储在单片机内部 FLASH，每次修改仅仅对这些参数进行改动，加快了单片机更新的速度。图 5-14 显示了 F8013 内部 FLASH 的一种应用划分。其中控制参数存储了 54 字节的控制参数。这些参数可以直接通过无线通信模块下载到单片机内存储，避免了繁琐的程序更新的过程。

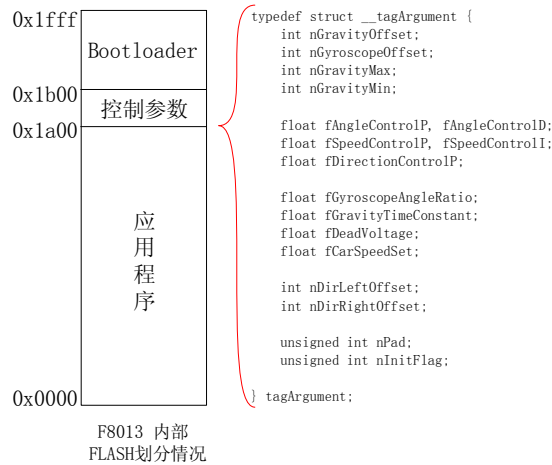


图 6-14 F8013 内部 FLASH 应用划分

### 6.3 桌面静态参数调试

桌面静态参数包括传感器参数以及在车模静止状态下的控制参数整定。

#### 6.3.1 传感器参数

车模传感器参数包括了测量车模运动状态各传感器在车模静止时的零点偏移量以及单位换算的比例值。在图 5-15 中将参数标记出来了。

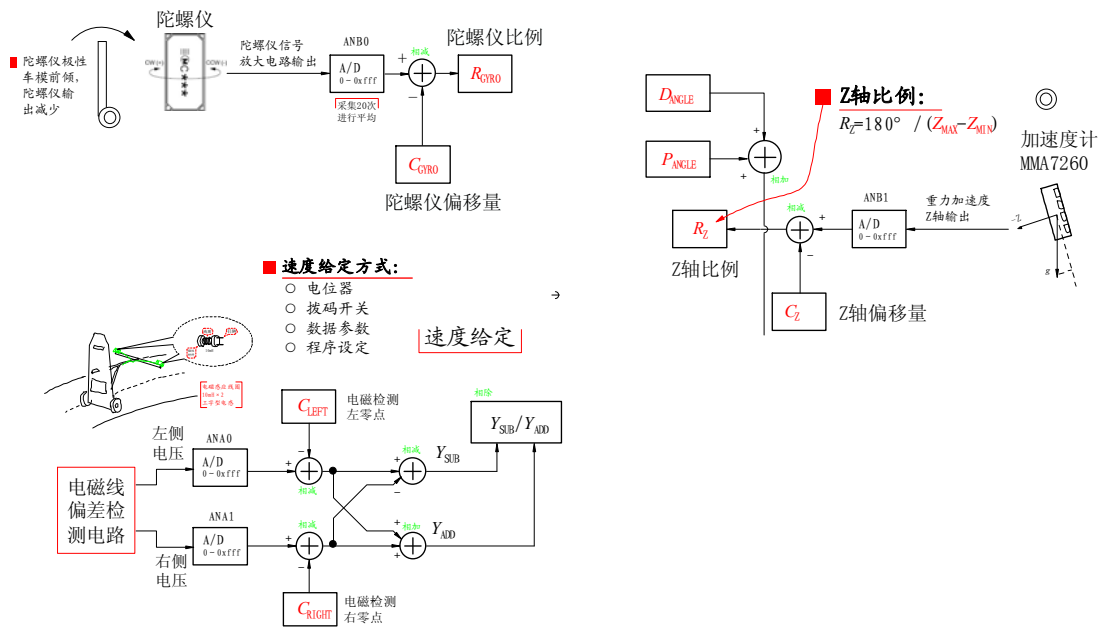


图 6-15 需要整定的传感器参数

#### 1、测量传感器零点偏移量

在调试桌面，保持车模垂直静止状态，关闭 20kHz 电磁电源。通过单片机软件采集

各模拟通道 AD 值，传递给上位计算机进行显示。确定各传感器的零点偏移量：

$C_{ANGLE}, C_Z, C_{LEFT}, C_{RIGHT}, C_{DIR}$ 。

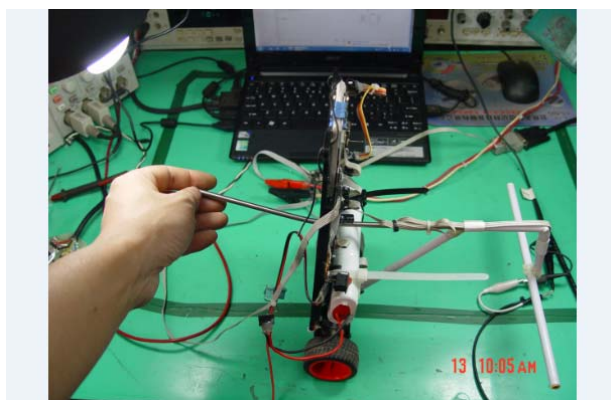


图 6-16 车模保持垂直静止

通过多次采集平均可以获得更加稳定准确的。在参考方案中，采集了 1024 次各通道数值进行平均作为各通道的零点偏移量。

## 2、测量加速度传感器 Z 轴最大值、最小值

保持车模正面朝上和反面朝上静止可以测到 Z 轴最大值和最小值。

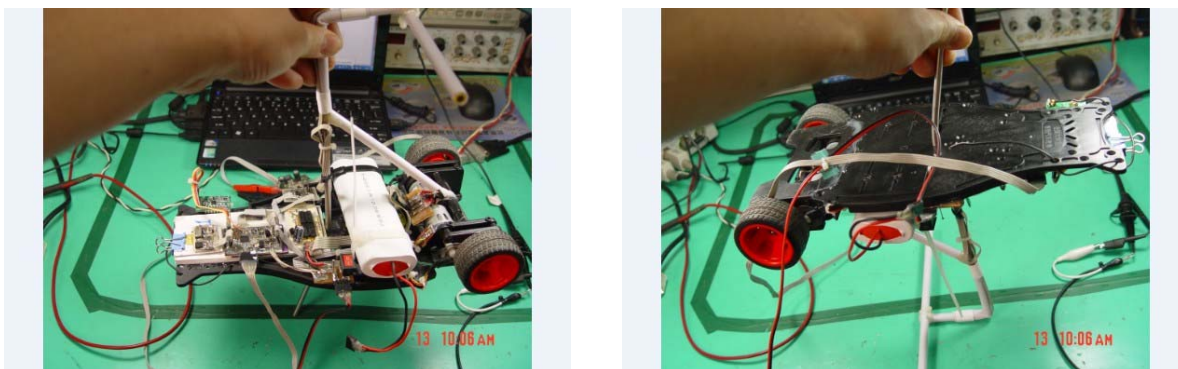


图 6-17 测量加速度传感器的极值

加速度传感器的最大值和最小值可以用来确定加速度传感器 Z 轴对应角度的比例值： $R_z = \frac{180}{Z_{MAX} - Z_{MIN}}$ 。通过这个比例可以将加速度传感器 Z 轴信号转换成角度信号，单位是度。

### 3、标定车模陀螺仪比例值

陀螺仪比例值  $R_{GYRO}$  决定了陀螺仪速度积分得到的角度单位是度，和重力加速度传感器是一致的。这个比例可以通过公式计算出来： $R_{GYRO} = \frac{R_{AD}}{R_G \cdot K}$ 。其中  $R_{AD} = 3.3V / 4096$ ，表示 AD 转换因子。 $R_G = 0.67mV \cdot \text{deg}/\text{sec}$  为陀螺仪比例因子，通过陀螺仪数据手册可以查到。 $K = 5.1$  是陀螺仪信号放大倍数，根据前面电路参数可以计算出这个放大倍数。将上述数值代入公式可以计算得到陀螺仪比例因子： $R_{GYRO} = 0.24$ 。由于上述存在着参数误差，上述计算得到的比例因子只是一个参考值，需要通过实验测量最终确定实际的正确比例值。

实验的基本方法就是通过改变车模一定的角度，对比陀螺仪积分得到的角度变化量和重力加速 Z 轴的变化量是否相同。具体步骤如下：

1) 首先将算法中角度补偿回路断开；

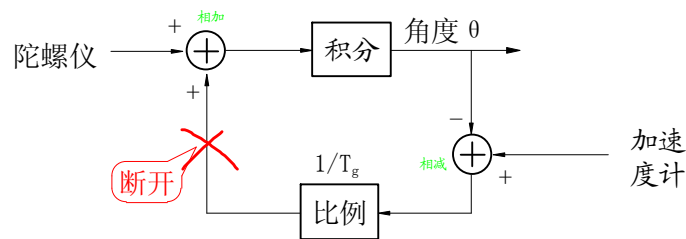


图 6-18 测量陀螺传感器比例因子，角度补偿回路断开

2) 保持车模一定角度，测量出 Z 轴角度输出和加速度积分角度输出；

3) 改变车模角度，测量 Z 轴角度输出变化量和加速度积分角度变化量；

如果 Z 轴角度输出变化量与陀螺仪角度变化量是一样的，则说明现在的陀螺仪比例因子是正确的。否则，将 Z 轴角度变化两除以陀螺仪角度变化量，然后再乘以现在的比例因子就可以得到正确的陀螺仪比例因子了。

由于陀螺仪积分存在漂移，所以上面矫正过程需要快速。通过多次矫正，便可以得到准确的陀螺仪的比例因子。

也可以不将角度补偿回路断开，通过陀螺仪输出角度曲线也可以判断陀螺仪的比例因子是否合适。如果陀螺仪比例因子小了，则陀螺仪角度输出就会有一个缓慢跟踪过程。相反，如果陀螺仪比例因子大了，陀螺仪角度输出就会有一个过冲。通过几次调整，可

以得到陀螺仪的正确比例因子，使得陀螺仪的角度输出能够正确跟踪重力加速度 Z 轴输出信号。图 6-19 显示了几种不同的陀螺仪比例因子下的陀螺仪角度输出情况。

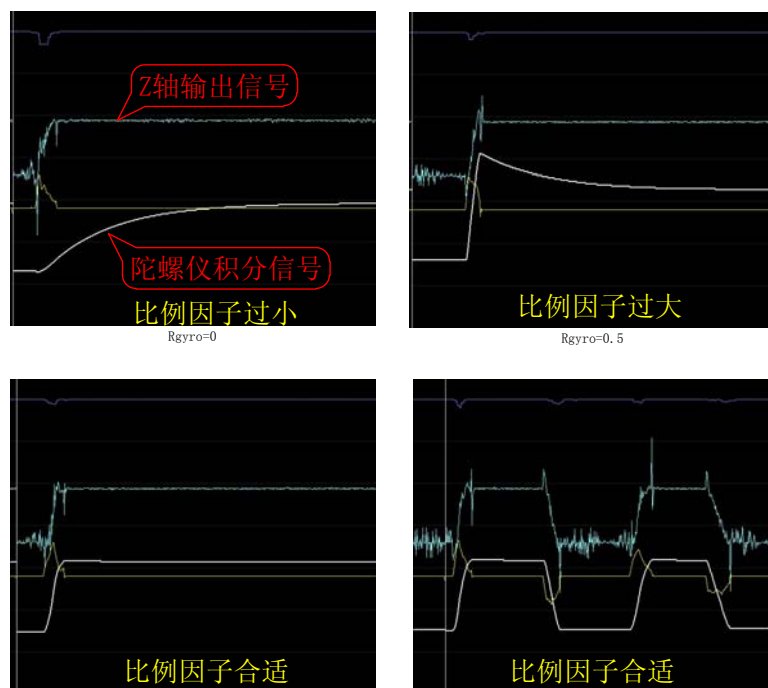


图 6-19 几种不同陀螺仪比例因子角度输出

如果采用图 2-32 所示的无重力加速度的控制方案，则可以大大简化传感器参数的整定。没有了重力加速度计传感器，对于测量车模倾角的陀螺仪和车模转角的陀螺仪就可以不用进行比例因子矫正，最终通过控制参数来补偿不同传感器的性能。

### 6.3.2 车模控制参数静态调试

车模在设定速度为 0 时，对于车模各个控制参数进行调试称为车模静态调试。需要调试的控制参数如图 6-20:

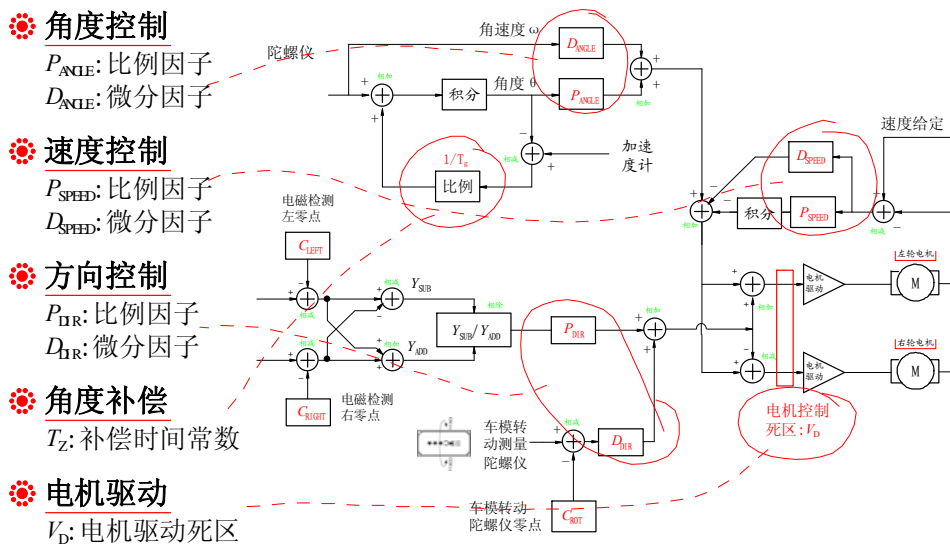


图 6-20 车模控制参数

这些参数调整的顺序遵循着，角度、速度、方向调整的顺序。在这三组参数调整的过程中，角度补偿时间常数和电机死区参数都可以首先设定为一个缺省值。比如，角度补偿时间常数  $T_Z$  可以选择 3 秒。电机死区  $V_D$  选择为 0。在前三个参数整定完之后在进一步优化这两个参数。需要特别强调有两个方面：第一个方面是角度补偿常数  $T_Z$  不能够过小，原因在后面调整该参数时进行分析。第二个方面是对于各传感器的零点偏移量尽可能的修正准确。如果在下面调试过程中有的传感器的零点偏移量发生了漂移，需要及时修正。

### 1、角度参数整定

角度控制参数包括比例和微分两个参数。比例参数 ( $P_{ANGLE}$ ) 相当于倒立摆的回复力。这个参数需要大于重力加速度所产生的效果才能够使得车模保持直立。随着这个参数逐步增大，车模开始能够保持直立。该参数进一步加大，车模开始出现来回的摆动现象。微分参数 ( $D_{ANGLE}$ ) 相当于倒立摆中的阻尼力，它可以有效的抑制车模的摆动。当该参数过大时，会引起车模本身的震动。这是由于车模本身不是一个刚体，车体具有一定的共振频率。微分参数过大时会使得车模在电机的驱动下产生车体的共振。

调节这两个参数可以遵循先比例后微分的过程。首先调整比例参数，使得车模能够保持直立并且开始来回摆动了。说明此事比例参数已经可以克服重力的影像。然后逐步

增加微分参数，车模逐步保持直立稳定。增大微分参数直到车模开始共振，这样就可以确定微分参数的最大值了。然后适当减小微分参数，然后逐步增加比例参数，直到车模又开始震荡，这样便可以确定比例参数的最大值。然后适当减小比例参数。在这些参数附近进行试验，直到寻找到一组车模直立的控制最优参数。

具体过程可以参见下面图 6-21 图形所示。



图 6-21 角度参数调整过程

车模保持直立的比例和微分控制参数有一个相当大的区间。这为后面进行车模动态参数调整留下了修改空间。

由于陀螺仪和重力加速度传感器安装误差已经信号的漂移，经过调整角度控制参数的车模不会精致停留在某一点。它会朝着车模角度倾斜的方向加速行驶。在这个过程中，对于车模重力加速度和陀螺仪的零点参数进行多次调整，可以逐渐消除车模倾斜角度，但是最终随着传感器的漂移，车模还是会加速移动。保持车模静止或者恒速运行需要通过速度控制来实现。

## 2、速度参数整定

在角度参数初步确定之后，便可以进行速度控制参数的整定。速度控制参数包括比例 ( $P_{SPEED}$ ) 和微分 ( $D_{SPEED}$ ) 两个控制参数，这里的比例和微分定义是指反馈速度而



言。如果相对于车模的反馈位置（速度的积分量），这两个参数也可以分别对应称作积分（ $I_{SPEED}$ ）和比例（ $P_{SPEED}$ ）参数。

首先请注意速度的调制本质上是通过调整车模的角度来实现的。图 2-38 中显示了速度控制环节。比例控制参数决定抑制速度误差的能力，但是只有比例控制，车模速度控制就会出现双积分负反馈控制。第一个积分是误差的积分，第二个积分是车模倾角所产生的加速度积分。因此车模速度就会出现震荡现象。通过微分控制可以消除这个震荡现象。

在调整时，由于是进行静态调整，故此先将速度控制闭环的速度设定值置为 0。然后开始后面的比例和微分参数的调整。具体的调试过程可以首先逐步比例参数，车模能够在一定平衡点附近来回运动。然后在逐步增加微分参数，车模就会很快停留在平衡点出。使用外力冲击车模，车模能够很快趋于静止。然后再逐步增加比例和微分控制参数，使得车模抵抗外部干扰冲击的能力逐步增强。调试过程见图 6-22 所示。



图 6-22 速度参数调整过程

当比例和微分参数达到一定的数值后，车模开始出现摆动现象。至此可以知道比例

和微分的大体范围。在此范围内通过几次调整测试，可以初步确定速度控制参数的最佳组合。

### 3、方向参数整定

方向控制包括两个参数：比例控制参数（ $P_{DIR}$ ）和微分控制参数（ $D_{DIR}$ ）。在调整方向参数是，设置速度参数为 0，打开桌面 20kHz 的电磁电源。比例控制参数可以使得车模方向恢复正确位置。当该参数逐步增加时，车模方向回复速度逐步加快。当比例参数增加到一定数值之后，车模的方向回复由于过快，会出现方向过冲现象。这个现象可以通过图 5-23 显示。此时通过增加微分控制参数可以抑制这种方向过冲的现象。通过几次组合测试，可以选择一组合适的比例控制和微分控制参数使得车模方向控制即迅速又不会出现过冲现象。



图 6-23 方向参数调整过程

方向控制参数如果调整的过大，就会影响车模角度的控制。因此在此调试过程中，需要兼顾车模的方向和直立控制两方面的稳定性。

以上三个步骤调整了车模的角度、速度和方向控制。这些参数是在车模静止的时候进行调试的。它们之间具有一定的相互影响。在后面车模运动过程中根据前面调整所遵循的原则和它们之间相互影响的体验，逐步优化这些参数，使得车模在运动过程中保持稳定、快速的特性。

### 4、补偿时间常数整定

车模的角度补偿时间常数（ $T_z$ ），决定了抑制陀螺仪积分漂移的能力，也决定了车模速度控制中角度跟踪的速度，同时决定了抑制重力加速度中干扰信号的能力。虽然它

只是一个参数，但是它的调整同时会影响到角度和速度的控制。其中对于角度控制影响尤为明显。

一开始在调整前面三个控制参数的时候，可以选择补偿时间 3 至 4 秒左右。如果陀螺仪零点漂移很小，可以适当增加该补偿时间常数。如果陀螺仪零点漂移大，那么可以逐步减小这个补偿时间常数。在减小时间补偿常数时，会发现车模会出现来回摆动的现象。这个现象和车模角度控制时，比例值过大时产生的车模摆动现象一致。因此此时可以适当增加角度控制参数中微分参数 ( $D_{ANGLE}$ ) 来抑制车模摆动的现象。调试过程参见图 6-24 所示。

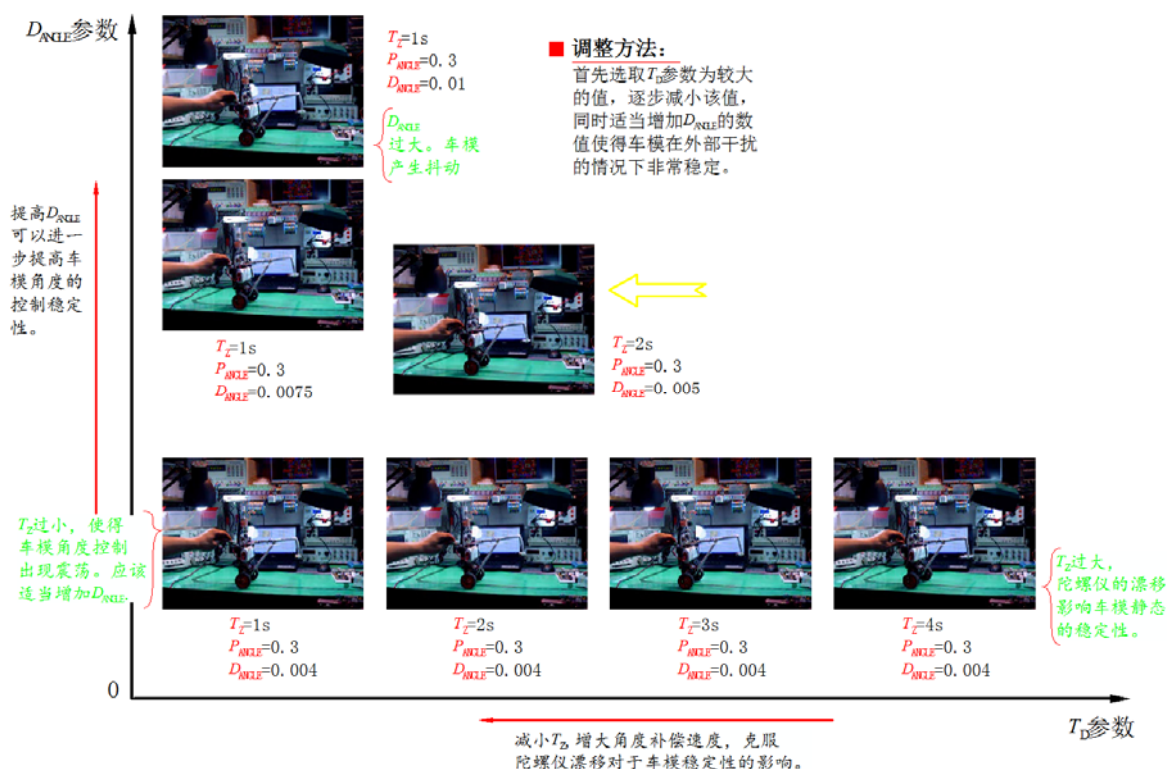


图 6-24 角度补偿时间常数调整

为什么角度补偿时间常数过小会引起车模摆动呢？对图 5-25 中加速度上的进行受力分析，可以知道重力加速度传感器安装具有一定的高度，此时车模倾角发生变化，就会在重力加速度传感器的 Z 轴出现加速度分量。这个分量与车模倾角加速度成正比。如果此时重力加速度角度补偿时间常数过小，对应角度计算回路分析可以知道，此时 Z 轴中由于车模角度的加速度所引起的信号正好会抵消原来角度控制中的微分分量，使得车模会出现前后摆动，故此需要增加角度控制中的微分分量来抑制车模的摆动。图 5-25 显示了重力加速度传感器的受力分析。

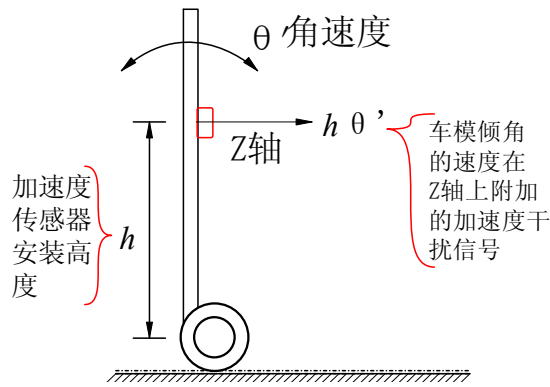


图 6- 25 Z 轴附加信号分析

消除车模的摆动可以通过两个方法实现，一种是增加补偿时间常数  $T_z$ ，这也就解释来前面要求在调整参数时为什么补偿时间常数不能过小的原因。第二种方法就是通过降低角速度传感器的高度来减少车模摆动

### 5、电机死区常数整定

在车模动力传动系统中，各个环节都存在着静止摩擦力，它会降低车模静态稳定性。可以通过给电机控制电压增加一个死区常数来克服。

调整死区常数时，需要保持车模的直立静止。死区常数为 0 时，车模会在静止状态出现来回小幅度的来回运动。通过逐步增加死区常数，车模来回运动现象逐步消失。当死区常数过大时，会出现车模电机震动现象。如图 5-26 所示。



图 6- 26 死区常数调整

通过逐步调整可以确定一个相对比较适中的死区常数。

在智能汽车竞赛是，由于是进行速度的比赛。电机的死区常数可以就设置成 0，它对于车模在高速运行没有影响。

## 6.4 现场动态参数调试

通过上面几个步骤，逐步确定各个控制环节中的参数。但是这些参数是在车模速度保持为 0 的状态下进行调试的。究竟这些参数在车模高速运行下是否配合良好，还需要“是骡子是马，拉出来溜溜”。因此将车模在比赛赛道上进行高速运行，重新对于各个控制参数进行优化，使得车模跑得又快又稳定。

动态调整一般是在静态调整所得到的各个参数基础之上，逐步提高车模速度的设定值，观察车模在不同的运行速度设定值下，它的角度、速度以及方向控制的性能。在保证车模的角度和方向稳定的条件下，逐步提高车模的速度，观察出现的各种现象，根据前面调整控制参数所得到它们之间的相互影响的关系，通过调整参数，使得车模能够兼顾稳定性和速度。

在车模运行时，调整控制参数非常复杂。这个过程需要反反复复进行测试。测试过程中，为了避免车模失控冲出赛道撞击障碍物而损坏，因此建议在车模上增加无线遥控开关，可以在紧急时停止车模的运行。

测试车模的赛道需要在条件允许的情况下尽可能的复杂，对于可能出现的赛道元素希望都能够进行测试，找到车模在各种极限赛道情况下的最优参数，以及可能最快的速度设置，这为后面参加现场比赛制定比赛策略打下基础。在现场比赛时，往往可供现场调试的时间不允许选手对比赛赛道进行完整的测试，故此需要选择有特征的赛道部分进行测试，最后给出合理的控制参数。选择什么样的赛道部分进行测试，需要平时动态调试的经验积累。

车模动态调试不仅是对于各种控制参数进行调整，同时对车模的机械结构也进行测试和调整。在比赛规则规定范围之内，合理安排车模的重心位置以及传感器的安装位置，提高车模的稳定性。在第四章介绍了陀螺仪安装误差会造成车模在转弯时速度增加或者减少。因此通过车模动态运行，可以对陀螺仪安装角度进行调整。另外再车模经过电磁交叉的十字路口时，通过车模方向控制是否稳定判断电磁传感器安装是否水平。

## 6.5 方案改进与车模整体水平提高

车模经过制作、编程、调试之后，在真实环境中进行运行，进一步发现存在的错误，

优化各个控制参数，逐步提高车模的运行速度。通过不断的磨合、修改获得一组最优的控制策略。

在运行测试中，也可以对控制算法做进一步的改进。增加车模状态观测量，修正其中非线性环节，提高车模在高速运行时的稳定性和抗干扰能力。

比赛无极限，追求无止境。在已经取得的调试结果之上，仔细观察，认真思考，对于发现的问题，逐步完善控制方案，突破现有观念的限制，寻求更加先进完美的方案，从而在未来的竞争中获胜。在比赛规定的技术范围之内，追求卓越。在不断的改进与实验中，收获的不仅是最终的比赛成绩，而且还有自己动手的能力和勇于探索的精神。

## 第七章、结束语

本文使用了尽量少的理论知识介绍了车模直立行走的参考控制方案。通过这个方案，同学们会觉得制作直立行走的车模参加竞赛不是困难的事情，而是相当容易和有趣的过程。

通过制作，同学们可以基本上实现完成比赛的车模。完成之后，同学们会逐步发现该方案的不足之处，进而萌发改进的想法。为了简化控制方案设计，本文中在很多方面都进行了简化。比如实现车模控制需要通过控制电机的加速度来实现，在开始介绍的时候，对于电机的加速度控制只是改变了电机的驱动电压。但是电机的加速度应该是通过其中的电流所产生的电磁力矩来实现的。为此需要在电机控制回路中应该增加上电流控制这样可以保证车模有着更加稳定性能。根据自己的知识和制作的经验，同学们可以在电路设计、控制算法、机械调整中发现许许多多可以改进的地方。为了实现这些改进的想法，取查阅更多的相关文献，学习先进的现代控制理论知识，提高自己的理论水平和实践能力，最终产生对于自动控制专业的强烈兴趣，这正是同学们应该得到的收获。

在制作竞赛车模的方案中，应该尽量选择价格便宜，容易购买到的器件。为了避免参赛队伍在选择传感器方面使用昂贵的器件，竞赛规则中限定加速度传感器和陀螺仪使用的型号。考虑到将来传感器市场供货情况的变化，组委会将扩大传感器选择的范围。请参赛队员及时留意竞赛网站公布的信息。

由于今年车模直立行走竞赛是首次设立，同学们制作的电磁组直立行走的车模运行

速度可能相对比较慢，故此在竞赛规则中限制了比赛赛道的长度和难度。同时，在比赛中，不要求车模从静止两秒钟自动开始比赛，也取消了比赛结束时车模停止在起跑线后三米内赛道上的要求。竞赛组委会将会根据同学们制作车模的具体情况，对电磁组比赛赛道、比赛规则做进一步的修订，修订信息会及时发布在竞赛网站上。

## 附录：

### 1、版权声明

《第七届全国大学生“飞思卡尔”杯智能汽车竞赛电磁组参考设计方案》文章版权属于智能汽车竞赛秘书处。

### 2、参考文献

- (1)《电磁组竞赛车模路径检测设计参考方案》竞赛秘书处 2010-1, 版本 1.1
- (2)《第五届全国大学生智能汽车竞赛 20kHz 电源参考设计方案》竞赛秘书处技术组, 版本 1.1

### 3、直立车模参考设计方案视频

可以在竞赛网站下载到本文介绍的参考设计方案车模演示视频：

<http://www.smartcar.au.tsinghua.edu.cn/web/index.jsp>: “直立车模参考设计方案视频”

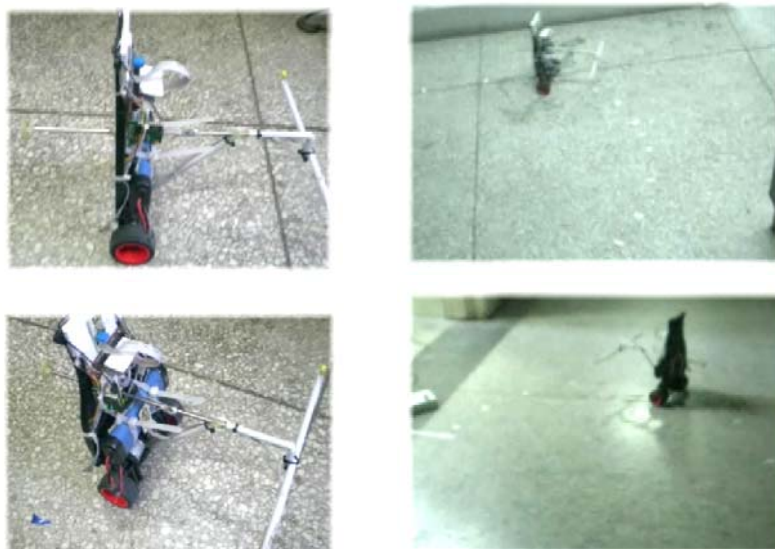


图 7-1 参考设计方案视频截图

#### 4、电磁组直立车模调试指南及视频

竞赛网站可以下载本参考方案中直立车模调试制作指南手册以及车模调试过程的讲解视频文件，观看车模调试过程中的车模运行的动态情况。



图 7-2 参数整定与调试指南





## 5、软件算法全图

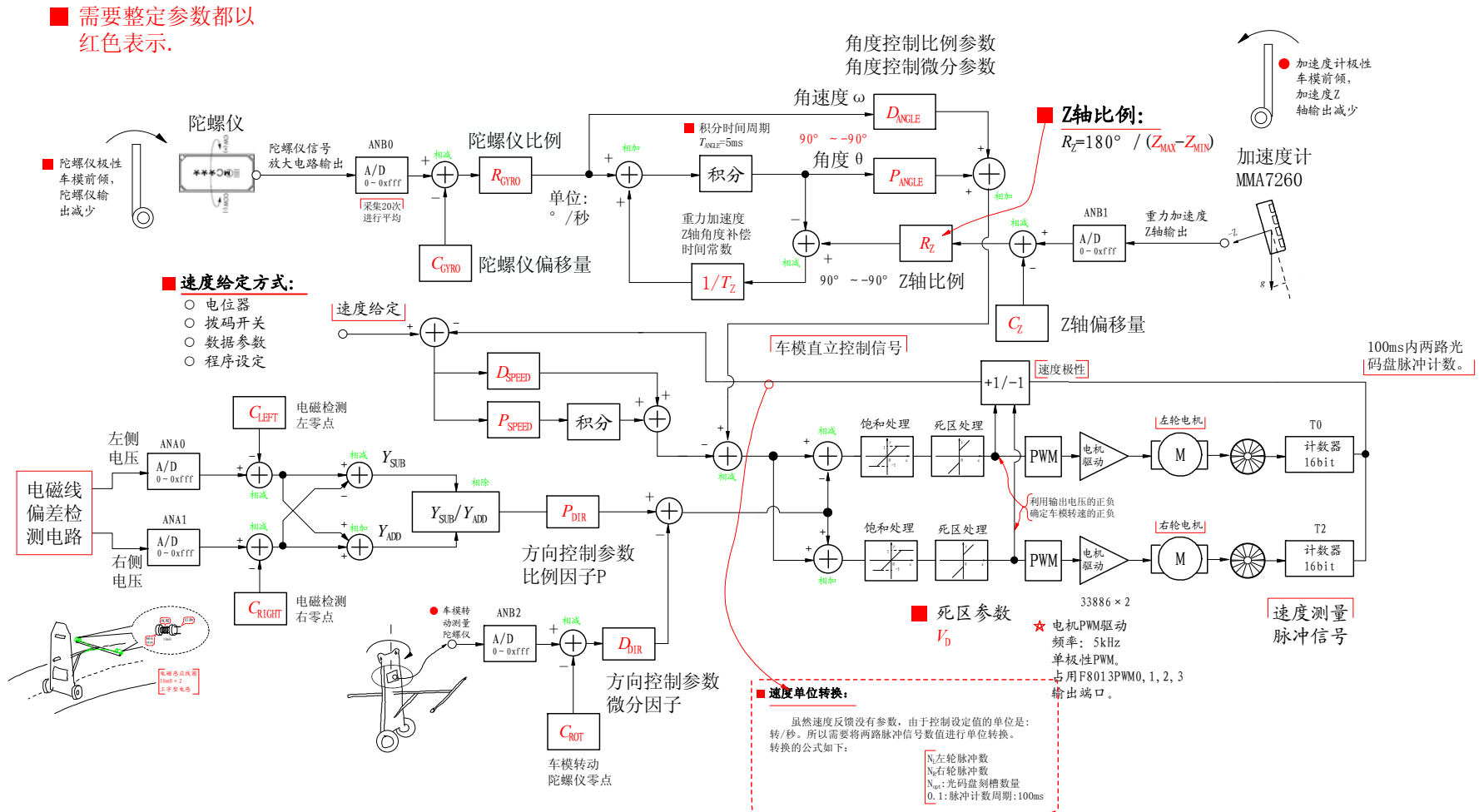


图 7-3 软件控制算法全图

## 6、参考设计方案电路全图

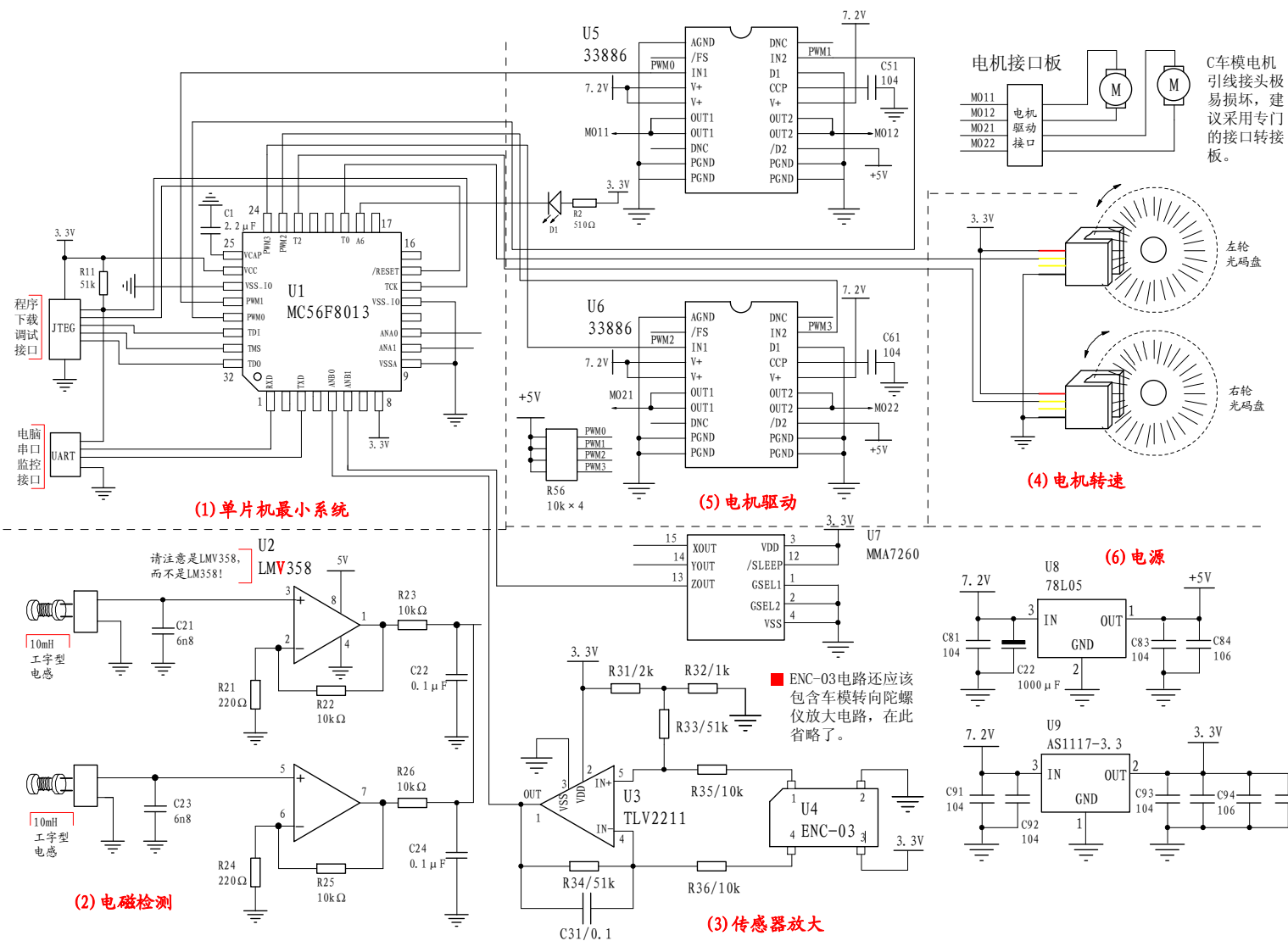


图 7-4 参考方案电路全图

## 7、参考设计工程设计文件

可以在竞赛网站下载到本文介绍参考设计方案软件工程文件。“直立车模参考设计方案 DSC 软件工程文件.RAR”。

## 8、DSC 工程目录与文件说明

(1) SMMNew1\Code 目录：保存由 Codewarrior 中 PE 自动生成 DSC 的各模块的驱动程序文件。

(2) SMMNew1\AppCode 目录：保存车模控制算法文件。其中：

CARSUB.C, .H: 车模全部控制函数文件，头文件。

CONFIG.C, .H: 车模控制参数 FLASH 存储函数文件，头文件。

DEBUG.C, .H: 车模串口调试函数文件，头文件。

F8013.C, .H: F8013 底层函数文件，头文件。