板球系统设计及其控制算法的研究

摘要

本文介绍了一种基于视觉的板球控制系统，并重点讨论了其控制算法。该系统通过摄像头采集小球的位置信息，计算得到小球的坐标，通过相应的算法来控制电机的运行。用陀螺仪测量木板的倾斜角度，通过两个直线电机的运行来驱动木板的倾斜，从而驱动小球滚动。其目的是实现小球的滚动控制以及轨迹追踪。其中涉及到的研究内容有：智能控制、运动控制、图像处理等。

文中介绍了系统的整体结构与制作过程，详细讨论了其控制的算法的实现。对单片程序的设计提出了独特的见解，并总结出了一种比较通用的单片程序开发框架。针对装置的特点使用多组PID,相互串联或并联使用来控制系统。对于实际制作和调试过程中所遇到问题及解决方法进行了深入的探讨。

关键词

单片机；板球系统；串联PID控制；单片机程序通用框架；图像处理；电机控制

Design of Ball & Plate System and Research of Its Control Arithmetic

Student: Zhan Kang, School of Physics & Information Engineering

Teacher: Qin Gong, School of Physics & Information Engineering

Abstract

This paper introduces a ball & plate system based on computer vision and discusses its control algorithm. This system collects the location information of the ball through the camera, and then calculates the coordinates of the ball, and controls the operation of the motor through the corresponding algorithm. Using a gyroscope to measure the tilt angle of the board, the tilt of the board is driven by the operation of two linear motors to drive the ball rolling. The goal is to achieve the ball rolling control and track tracking. The research contents involved include: intelligent control, motion control, image processing, etc.

This paper introduces the overall structure and production process of the system, and discusses the realization of the control algorithm in detail. This paper presents a unique view to the design of a single program and summarizes a common single - chip programming framework. In order to control the system, multi-group PID is used, in series or in parallel for the features of the device. The problems and solutions in the process of actual production and debugging are discussed in depth.

Keywords

MCU; ball & plate system; series PID control; frame of MCU program; image processing; motor control.

目 录

[1 绪论 1](#_Toc513989929)

[1.1 研究背景和意义 1](#_Toc513989930)

[1.2 国内外研究现状 2](#_Toc513989931)

[1.3 论文内容 3](#_Toc513989932)

[2 板球系统总体设计方案 4](#_Toc513989933)

[2.1 系统总体结构构建 4](#_Toc513989934)

[2.2 STM32控制器 5](#_Toc513989935)

[2.3 动作执行单元 6](#_Toc513989936)

[2.3.1 电机的选择 6](#_Toc513989937)

[2.3.2 直线电机 6](#_Toc513989938)

[2.3.3 电机驱动模块 7](#_Toc513989939)

[2.4 摄像头模块 8](#_Toc513989940)

[2.4.1 摄像头的选型 8](#_Toc513989941)

[2.4.2 openMV图像传感器模块 8](#_Toc513989942)

[2.5 角度测量模块 9](#_Toc513989943)

[2.5.1 角度传感器的选择 9](#_Toc513989944)

[2.5.2 MPU6050六轴传感器 10](#_Toc513989945)

[3 板球系统程序设计 11](#_Toc513989946)

[3.1 板球控制系统的程序构架 11](#_Toc513989947)

[3.2 驱动模块的设计 12](#_Toc513989948)

[3.2.1 openMV驱动程序 12](#_Toc513989949)

[3.2.2 电机驱动程序 14](#_Toc513989950)

[3.3 应用层程序设计 15](#_Toc513989951)

[3.3.1 系统控制程序 15](#_Toc513989952)

[3.4 总调度层程序设计 16](#_Toc513989953)

[3.4.1 前台时间调度 16](#_Toc513989954)

[3.4.2 后台中断服务 18](#_Toc513989955)

[4 板球系统控制算法 19](#_Toc513989956)

[4.1 板球系统物理模型 19](#_Toc513989957)

[4.2 PID控制理论 20](#_Toc513989958)

[4.3 板球系统的PID控制 21](#_Toc513989959)

[4.4 画圆算法的实现 22](#_Toc513989960)

[5 系统调试与评价 24](#_Toc513989961)

[5.1 直线电机调试 24](#_Toc513989962)

[5.2 平板水平的调试 24](#_Toc513989963)

[5.3 系统联合测试 25](#_Toc513989964)

[6 结论与展望 27](#_Toc513989965)

[致谢 28](#_Toc513989966)

[参考文献 29](#_Toc513989967)

# 绪论

## 研究背景和意义

板球控制系统是一个经典的自动控制系统。自上个世纪80年代以来就有很多国内外学者对其做了各种研究。通过板球系统这一平台可以研究验证各种控制算法,然后将其应用到生活中的各个地方。在实验室中学者设计研究自己板球系统,通过触觉传感器(触摸板,电阻屏等),或者视觉传感器(摄像头等),来测量小球在平板上的坐标,然后通过计算该坐标来控制电机等动力机械驱动平板倾斜,从而控制小球在平板上滚动。通过这样的装置得到的研究成果可以应用到生活中的很多方面。例如对于小球位置的检测、预测,需要应用计算机视觉及其相关算法,这可以应用到工业中,例如无人机的目标跟踪,或者是无人汽车的行人检测。同样控制算法也可以应用到方方面面,当前工业机器人和消费级智能机器的发展无一能离开自动控制算法。因此板球控制系统是一个很有价值的研究课题,也是一个很好的研究和验证相关算法的平台。

今天计算机技术正在飞速发展,而计算机视觉一直以来都是计算机科学中的热门话题,人们希望计算机有类似人眼一样看的功能,即使用摄像机和处理器代替人眼使计算机能够有像人类那样能够对目标进行识别,分割,跟踪的功能。如果将计算机视觉与智能控制相结合,计算机别能自行识别环境中的物体,从而根据设定的动作去操控执行机构,完成相应的功能。这已经在工业、农业、制造业、军事上有广泛的应用。随着科学的发展,越来越多的学者投入计算机视觉与智能控制相结合的研究。所以为了研究其理论的好坏,板球系统被设计了出来。国内外不同学者设计了不尽相同的板球控制系统,因为板球系统自身有强耦合、非线性等特征,成为了研究验证控制算法的一个理想的平台。

## 国内外研究现状

对于板球控制系统的研究,国内的一些研究状况如下:吉林大学的田彦涛教授和他的研究生们研究设计了一种基于扩展卡尔曼滤波的状态观测器。该状态观测器能够实时分析评估板球控制系统运行时小球运动过程中所受到的摩擦力,从而得到一系列与摩擦力相关的因素,然后利用这些数据对摩擦力建立数学模型,最终设计了摩擦力补偿器对摩擦力进行补偿。2009年前后,将研究成果加入基于视觉的板球控制系统中加以验证,并且设计了一套的参数自动调节自动控制方案,考虑到控制量可能受饱和特性限制,引入了约束条件,然后利用遗传算法设计并优化模糊规则,从而有效地降低了自动控制中的位置偏差。从2000年前后开始,清华大学自动化系针对板球控制系统的模糊控制方案进行了一系列的研究,最终提出了一种多变量模糊控制方案。在此基础上设计了小球运动轨迹规划和目标跟踪算法,并在利用计算机仿真作了检验。其后教研组设计并制作了自己的基于视觉的板球控制系统物理仿真平台,并将研究成果应用于该平台,最终实现了对板球实验平台的稳定控制。大连理工大学的李洪兴教授及其研究生在 simulink 环境下设计了变论域模糊控制器,其仿真效果明显好于应用位置式 PID 和增量式 PID的控制。最后他们将其应用在了自己研发的仿真平台上,在小球对正方形和圆形的轨迹跟踪上取得了很好的控制效果。中南大学张修如教授及其研究生也研究了板球控制系统的视觉系统,能够准确地获取了小球的实际位置。台湾成功大学的C.C.Ker教授运用触摸板构建了板球实验平台,通过触摸板来获取小球的位置,从而对实验装置进行控制。上海交通大学的学者研究开发了台面球系统,其利用气缸伺服控制机构作为执行机构进行控制。多年来,国内很多大学都设计了自己的板球控制系统物理仿真平台,其不再只是在simulink环境下进行的仿真实验,并是在板球控制系统的实物平台上验证控制算法。

国外的一些研究的状况如下:伊朗德黑兰大学的 Carolucas研究出了一套基于遗传内插算法的模糊控制器,并在CE151板球实验平台上进行验证,取得了较为理想的结果,最终实现了比较准确的小球的定位控制。美国伦斯勒理工学院也自主设计了板球系统,其使用触摸面板来检测小球的坐标。设计了一套简化的系统线性数学模型。其利用状态反馈控制器对系统进行实时控制,最终实现了小球的定位控制误差小于5mm,跟踪圆轨迹的半径误差小于18mm,控制的运动速度误差小于4.2mm/s。美国知名的控制理论方面的学者Hauser.J 于1992年前后,对板球控制系统的输入输出变量运用线性化理论进行了局部线性化处理,得到了简化后板球系统的线性化模型,并在实物仿真平台上获得了较为理想的控制效果。韩国汉阳大学的学者设计了一种基于变结构自适应控制器,其核心控制原理是分层次的模糊CMAC神经网络控制,该方法实现了小球的比较精确的定位控制。YUBAZAKIN 研究了一种动态单输入模糊规则群的控制方法,并将其应用在板球系统的椭圆和圆形的轨迹跟踪控制上面,取得很好的控制效果。FANX 在特定的板球控制系统仿真平台上研究了当环境不变的情况下,小球的运动轨迹的规划及跟踪问题,并取得了一定的成果。

## 论文内容

本论文各个章节的安排如下：

第一章主要分析了板球控制系统的发展现状、研究的结果、研究意义以及其相关研究方向。

第二章主要论述了系统硬件设计方案，对系统的硬件结构进行了介绍，讨论了芯片的选型以及各个模块的硬件电路设计方案。

第三章主要论述了板球系统的软件构架，对单片程序开发作了简单的总结，并对该系统各模块程序作了详细的讨论。

第四章讨论了PID算法在板球系统上的应用，对控制小球直线滚动算法和绕圆滚动算法做了详细说明。

第五章对系统的调试过程中遇到的一些问题作了说明，并提出了解决方案，对最终的运行结果作了总结；

第六章对论文进行了总结，概括了系统的特点和以实现的功能并对不足提出了改进意见，对系统将来的功能扩展作了展望。

# 板球系统总体设计方案

## 系统总体结构构建

本设计的系统构成框图如图2.1所示，主要由控制器、执行器、板和球、摄像机、图像处理单元构成。装置整体的工作过程为：摄像头采集小球在运动过程中的图像，经图像处理单元计算出小球相对于平板的坐标，将此坐标作为测量值与给定的坐标期望值一同送入控制器，控制器计算后将输出结果到执行器，从而控制小球的运动。

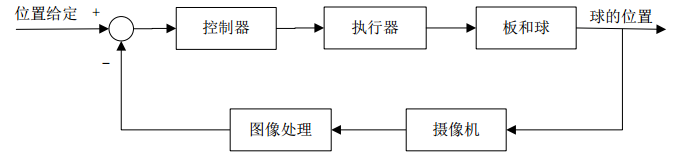


图 2.1 板球系统构成框图

整个系统的硬件组成框图如图2.2所示：使用陀螺仪芯片对平滑平板进行角度测量，并控制其初始状态为水平。当球放到平板上，摄像头对球进行定位和轨迹跟踪，将球位置信息传给STM32，进行相应算法处理，驱动模块使电机相应运动，木板倾斜控制小球滚动。矩阵按键可以选择工作模式，设置小球运动路径，调整参数。相应数据通过液晶屏显示出来。

STM32

**陀螺仪**

**液晶屏**

**摄像头**

**直线电机**

**矩阵键盘**

**电机驱动**

**平板**

图 2.2 板球系统硬件框图

系统实物图如图 2.3所示，白色平板边长60cm，摄像头放置在平板中心正上方120cm处，平板有两个直线电机和一个固定的支点支撑，通过连个直线电机的升降控制平板的倾斜，从而控制小球滚动。角度传感器放置于平板中心的正下方以保证测量角度的准确性。

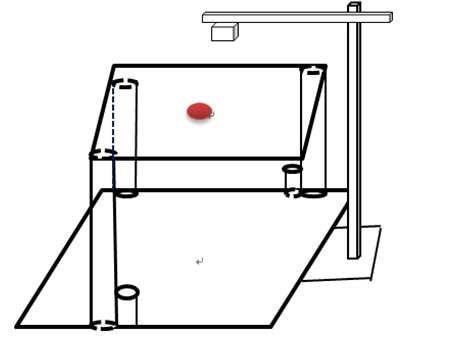


图 2.3 板球系统的模型图和实物图

## STM32控制器

本系统采用的是STM32F103zet6单片机作为主控制器。它采用ARM公司设计的高性能CORTEX-M3内核,具有32位CPU、72MHz主频、内置乘法器、64KB高速SRAM和512KB闪存,以及丰富的IO接口,可满足多种应用需求。作为计算和电机控制的核心,其起着至关重要作用。在此次板球控制系统的应用中使用到了 STM32的定时器、串口和IIC等外设。通过配置串口为中断接收方式,并设置串口参数(波特率,停止位,校验位,数据位),编写串口接收协议,使处理器可以准确的接收上位机发送过来的数据。STM32拥有一个功能多样的定时器,它既可以作为普通定时器使用,还可以结合其他外设,例如DMA、ADC一起使用,而且还可配置为PWM模式。在此次设计中使用定时器的PWM模式来输出PWM信号。应为要控制两个电机的正转和反转,所以使用到了两个定时器(定时器3和定时器5)输出4路PWM信号。这样就能方便地使用程序来控制电机的上升或下降,以及升降的速度。使用IIC接口与角度传感器MPU6050通信以获取平板的倾斜角度。

## 动作执行单元

### 电机的选择

方案一：使用步进电机作为滚球系统的驱动电机,因为步进电机的转动角度是通过程序中所给的PWM脉冲数量确定的,所以在程序中可以精确地计算电机转动角度,从而可以比较精确的计算出平板的倾斜角度而不需要其他的传感器来测量角度。但是在实际应用时步进电机的输出力矩可能达不到要求,而且步进电机在正反转切换时会比较缓慢,这会使平板在快到平衡位置时难以控制。

方案二：使用直线推杆电机作为滚球系统的驱动电机。直线推杆电机主要由电机、推杆和控制装置等机构组成。由于其中的电机为直流电机，所以能实现快速的改变转动速度，内置的减速齿轮也能保证较大推力的输出。

综合以上原因，结合装置模型，选择方案三。

### 直线电机

执行单元所使用的电机如图2.4所示，其最大推理150N，最大推进速度为90mm，额定电压为12V。实际测量中发现，该电机的启动电压（能让电机开始工作的最低电压）约为5V。在5V到12V之间输入电压与电机推杆的行进速度近似为线性关系。



图 2.4 直线电机实物图

### 电机驱动模块

主控制器STM32产生的PWM信号，并不能直接控制电机，需要驱动电路将PWM信号转化为电压后方能驱动电机工作。图2.5是电机驱动模块的接口电路。HCPL2630是高速光耦芯片，能有效隔离单片机控制电路和大电流较高电压的电机驱动电路，有效的保护了单片机。图2.6为电机驱动电路。IR2104是高性能的板桥驱动器芯片，两片一起使用配合IRLR7843构成全桥驱动电路。IRLR7843是N沟道的MOSFET，漏极持续电流可达160A内阻仅为3.3mΩ，完全满足设计的要求。

四路PWM分别控制两个电机的正反转，PWM1与PWM2控制一个电机，同理PWM3和PWM4控制另一个电机。当PWM1信号的占空比大于PWM2的占空比时对电机的输出电压为正，电机正转，反之电压为负，电机反转。两个占空比差值越大，电压越高，实现电压输出从-12V到12V可调，以此来控制电机推杆的升降速度。

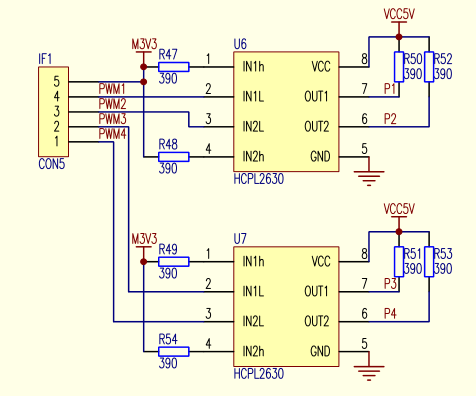


图2.5驱动模块接口电路

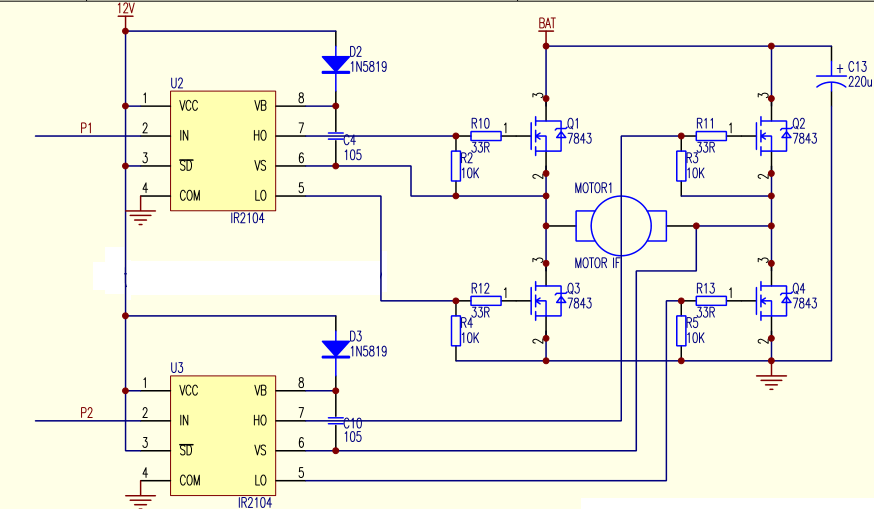


图2.6电机驱动电路

## 摄像头模块

### 摄像头的选型

方案一：采用普通摄像头模块。常用的摄像头传感器有OV7670。该图像传感器体积小方便使用直接输出VGA图像可直接操作。使用SCCB总线可对其控制，可以设置分辨率，从而读取不懂分辨率的8位影像数据，清晰度可以满足设计的要求。但由于本设计中所使用的单片机STM32没有相应的硬件接口，尝试过使用普通IO软件模拟该接口来读取数据，但效率过于低下。

方案二：使用OpenMV模块。OpenMV是利用OpenCV开发的计算机视觉模块。其上面有一块STM32F427芯片，172M的主频可以应付分辨率较小时的图像处理任务。模块使用了OV7725摄像头芯片，最大可以获取720P分辨率的图像，清晰度也能够满足设计的要求。OpenCV利用OpenCV的C语言库移植了一些算法到STM32F427的ROM中，并向外提供了python接口，使用起来非常灵活方便。此外其预留了丰富的硬件资源接口，可以很灵活的和其他流行的模块配合。本着不重复造轮子和更好地利用工具的想法本次设计选择了方案二。

### OpenMV图像传感器模块

OpenMV上的已实现的机器视觉算法有寻找色块、边缘检测、标志跟踪等。可以使用简单的python代码快速完成设计来达到自己的目的，比如说寻找小球算法，只需要借助其现有的找色块算法加以包装就能快速完成。此外其采用的STM32F427芯片，引出了非常多的外围功能接口，所以能方便与其他模块连接。方便地开发环境也是其的一个亮点，只需使用USB接口就能让将其连接到电脑上，并集成开发环境OpenMVIDE，协助完成编程、调试和更新固件等工作。

本次设计中使用到了其找色块算法和硬件接口UART。使用Python语言调用openMV内置的openCV库，进而实现寻找红色小球的算法。计算得到小球坐标后用自己实现的简单打包算法进行打包处理，然后通过UART将信息发送给主控制器。

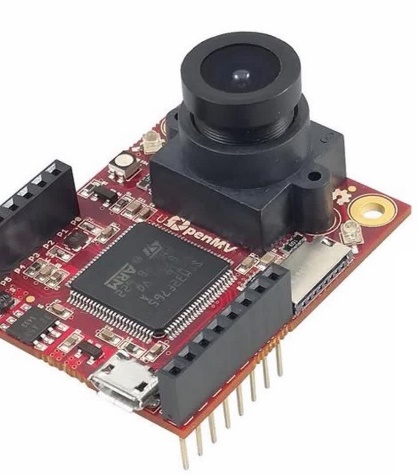
 

图2.7openMV实物图

## 角度测量模块

### 角度传感器的选择

方案一：使用加速度传感器。加速度传感器可以测量重力加速度，当该传感器静止时可以读得其在xyz方向上的加速度数值，由其与重力加速度的关系可以方便地计算出各个方向与重力加速度的夹角，从而计算出平板与水平面的夹角。但是该传感器运动时，由于自身存在加速度，这会与重力加速度相混淆，是的角度计算不准确。

方案二：使用陀螺仪传感器。但陀螺仪直接测量的是角速度，动态新能好，且不受外界其他因素影响，但是角速度积分而得到角速度，没有一个绝对的角度参考点，会由于误差而渐渐偏移。

结合方案一和方案二可以克服两种方案各自的不足。加速度传感器与陀螺仪的结合使用既能保证静态时角度的准确又能保证动态时的性能。

### MPU6050六轴传感器

MPU6050是一种六轴传感器,其将3轴陀螺仪和3轴加速计集成到了一个小小的芯片之内,并且还集成了一个的数字运动处理器 DMP(Digital Motion Processor)。该数字运动处理器可处理MPU6050内部的数据,还以通过IIC连接第三方数字芯片,并处理其数据。这些数据的处理结果可以缓冲在片内FIFO中保证数据的高速输出。此外MPU6050可编程设置片内数字滤波器,还可以编程配置DMP使其实现陀螺仪和加速度传感器数据的互补融合计算,并输出稳定的姿态四元数数据。另外DMP 有权使用 MPU 的一个外部引脚产生中断,保证了数据的实时性。

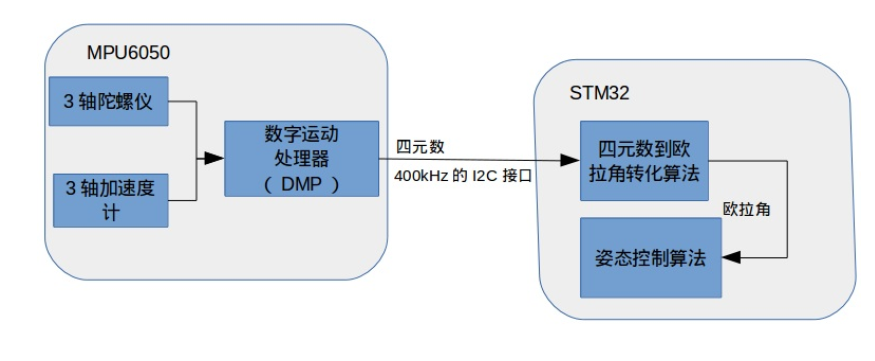


图2.8 DMP硬件解算四元数

# 板球系统程序设计

## 板球控制系统的程序构架

合理的系统构架能够有效地节省CPU资源保证系统运行稳定，利于维护和查错。由于STM32F103单片机的RAM相对较少，所以在资源管理方面如果移植嵌入式操作系统可能引入不必要的麻烦。而且加入实时操作系统必然增加程序的复杂度，也会增加调试的难度。所以针对STM32F103单片机的特点利用定时器设计以下的系统构架。

按键

display

Mpu6050

Motor

Ili9341

Key

角度传感器

电机

液晶屏

Control

前台（main）、后台（中断）

物理层

驱动层

应用层

总调度层

openMV

openMV模块

图3.1板球系统软件框架

物理层：由单片机外设构成，如传感器、键盘、屏幕等。

驱动层：由相关的函数和数据缓冲区、数据结构组成。将对物理层内外设的操作转变为，函数调用和数据更新。

应用层：由各功能模块（用户算法程序）组成。应用层直接调用驱动层的函数来实现各种功能。

总调度层：前台为main函数的主循环，利用时间调度对整个系统综合管理。例如利用定时器让按键扫描程序10ms执行一次，而电机控制部分代码100ms执行一次。后台则工作在中断系统中。这样一来既能有效的控制相关模块程序的运行周期，合理分配时间，节约了CPU资源，又能让各个模块协调运行。而且后台部分的存在也保障紧急任务的实时性。

该框架下每个外设的驱动模块都单独的放在一起（一般是同名的C文件和H文件各一个），这样当单独将每个驱动模块都调通后能方便的拼合在一起。方便调试查错，以及代码移植。由于C语言没有语言级别的封装功能，所以设计程序框架是就约定好了一些规则：在图3.1中的底层模块不能访问上层模块，而上层模块则有权限访问底层模块的公开数据和函数。这样设计的好处如下：1、硬件驱动程序和功能算法程序严格分开，使得代码结构清晰，方便扩展。2、由于应用层相对独立，方便移植。

## 驱动模块的设计

### OpenMV驱动程序

OpenMV作为一个独立的图像传感器模块，对其进行编程后能将小球以特定的数据格式用串口发送给主控器。主控制器需要用串口读数据，然后解析数据包，读出其中的小球坐标值。

使用串口读取数据需要先对串口进行先关初始化配置，设置波特率为115200，允许接收中断。初始化相关代码如下。

//

// 连接openMV相关端口初始化

//

//

void openMVPortInit(void){

USART2\_Init(115200);

}

接收和解析数据包程序工作在串口中断服务函数中，其工作流程如图3.2所示。

接收单个字符

存入缓冲区

接收到一帧数据？

解析数据包

是

否

图3.2位置坐标接收流程图

代码如下所示。

//

// 用在串口中断中处理接收到的openMV数据

//

void openMVDataProess(uint8\_t res){

static uint8\_t rxBuf[200];

static int16\_t rxBufCut = 0;

uint16\_t vas = 0;

uint16\_t i = 0;

// R 为起始标志

if(res == 'R'){

rxBufCut = 0;

}

if(rxBufCut > 199){

rxBufCut = -1;

}

// 读值到缓冲区

if(rxBufCut >= 0){

rxBuf[rxBufCut ++] = res;

}

if(rxBuf[0] == 'R' && res == 'O'){

// 收到一帧数据开始处理

if(rxBuf[1] == '('){

i = 0;

while(rxBuf[2 + i] != ','){

vas = (rxBuf[2 + i] - '0') + vas \* 10;

i ++;

}

spot.x = vas;

vas = 0;

while(rxBuf[4 + i] != ')'){

vas = (rxBuf[4 + i] - '0') + vas \* 10;

i ++;

}

spot.y = vas;

vas = 0;

}

}

}

### 电机驱动程序

电机驱动程序主要作用是将对电机的速度和运转方向的控制简化为函数操作，为简化后续的程序，将两电机的操作抽象为一个函数。代码如下所示。

//

// 电机速度更新

// speed -1000 ~ 1000 负数表示反转

// which : 0 和 1，电机 0， 电机 1

void motoSpeedUpdata(int16\_t speed, uint8\_t which){

uint16\_t speedTemp = 0;

if(speed > 1000){

speed = 1000;

}

if(speed < -1000){

speed = -1000;

}

switch(which){

case 0:

if(speed < 0){

speedTemp = - speed;

tim3PwmSetDutyCycle(speedTemp, 1);

tim3PwmSetDutyCycle(0, 2);

} else {

tim3PwmSetDutyCycle(0, 1);

tim3PwmSetDutyCycle(speed, 2);

}

break;

case 1:

if(speed < 0){

speedTemp = - speed;

tim5PwmSetDutyCycle(speedTemp, 1);

tim5PwmSetDutyCycle(0, 2);

} else {

tim5PwmSetDutyCycle(0, 1);

tim5PwmSetDutyCycle(speed, 2);

}

break;

}

}

## 应用层程序设计

应用层由两个模块组成。Control负责小区滚动控制和系统控制。Display模块负责在液晶屏上显示运行时的相关信息。

### 系统控制程序

系统控制部分主要有两个功能：1、通过按键控制板球系统的功能切换、调整参数等。2、使用控制小球按设定的方式滚动。为了方便统一管理设计了以下数据结构。

typedef struct

{

float xCtrl\_mea; // 左边速度 测量

float yCtrl\_mea; //右边速度 测量

float xCtrl\_exp; //左边速度 期望

float yCtrl\_exp; // 右边速度 期望

float xCtrlOut; // 左边速度输出 单位 m/s

float yCtrlOut; // 右边速度输出

float pitch; // 角度

float roll; // 角度

float zeroPitch; // 水平角度

float zeroRoll; // 水平角度

uint16\_t ballPosition[2]; // 小球的位置

uint16\_t centrePosition[2]; // 板中心位置

uint16\_t thro; // 油门

bool start; // 开始？

uint8\_t item; // 项目

uint8\_t state; // 状态

bool needRst; // 一些状态初始化

bool haveCalCooldinate; // 是否已经校准坐标

} systemControl\_t;

将控制程序的两个功能总结为两个函数，工作与主循环之中，如下图3.3所示，其中systemCtrl()函数通过按键设置系统所要执行的功能，并将相关信息保存于上述结构体中，ballMoveCtrl()函数根据所设置的功能对小球进行滚动控制。

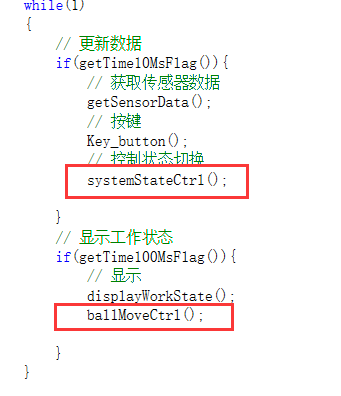


图3.3 主循环代码

## 总调度层程序设计

### 前台时间调度

前台调度模块在程序中的体现是main函数中的主循环。其具体代码如下。

while(1)

{

// 更新数据

if(getTime10MsFlag()){

// 获取传感器数据

getSensorData();

// 按键

Key\_button();

// 控制状态切换

systemStateCtrl();

}

// 显示工作状态

if(getTime100MsFlag()){

// 显示

displayWorkState();

ballMoveCtrl();

}

}

上述代码中getTime10MsFlag()函数如果连续两次调用的时间大于10ms则返回1，否则返回0。所以容易看出第一个if块中的函数大约每间隔10ms会被执行一次。同理第二个if块中的函数大约100ms。当然在这样的框架下需要保证每个功能函数执行一次的时间尽可能的短，不然会导致时间控制偏差较大。

getTime10MsFlag()函数利用单片的定时器实现其功能。在该工程中其代码如下。

void TIM2\_IRQHandler(void) //TIM2中断

{

if (TIM\_GetITStatus(TIM2, TIM\_IT\_Update) != RESET) //检查TIM2更新中断发生与否

{

TIM\_ClearITPendingBit(TIM2, TIM\_IT\_Update ); //清除TIMx更新中断标志

time\_10MsFlag = 1;

time100MsCut ++;

timeSCut ++;

time30MsCut ++;

if(time100MsCut == 10){

time100MsCut = 0;

time100MsFlag = 1;

}

if(timeSCut == 100){

timeSCut = 0;

timeSFlag = 1;

}

if(time30MsCut >= 3){

time30MsCut = 0;

time30MsFlag = 1;

}

timeCut ++;

}

}

//

// 10ms标志

//

u8 getTime10MsFlag(void)

{

if(time\_10MsFlag == 1)

{

time\_10MsFlag = 0;

return 1;

}

else return 0;

}

time\_10MsFlag为全局变量，定时器中断设置为没10ms一次，故time\_10MsFlag每隔10ms都会被赋值为1。在getTime10MsFlag函数被调用后，会首先判断time\_10MsFlag是否为1，如果为1将其赋值为0，并返回1，否则返回0 。如此一来如果两次调用getTime10MsFlag函数的时间小于10ms，则未能进入定时器中断，故time\_10MsFlag为0，返回0。由此便实现了10ms的时间控制。

### 后台中断服务

本次程序中后台中断服务程序主要有两个：1、用于时间调度的定时器中断，上文已经介绍过。2、用于接收和处理OpenMV数据的串口中断。

为了保证数据通信过程中的可靠性，需要对数据严格地实时接收，故需要用到中断服务。其具体代码如下。

void USART2\_IRQHandler(void)

{

u8 res;

if(USART\_GetITStatus(USART2, USART\_IT\_RXNE) != RESET)//接收到数据

{

res = USART\_ReceiveData(USART2);

openMVDataProess(res);

}

}

# 板球系统控制算法

## 板球系统物理模型

板球系统的目的是控制小球在平板上的运动，必须实时获取小球在平板坐标系中的实际坐标值，才能实现对小球位置的闭环控制。通过摄像头计算得到的坐标为图像坐标值，必选先转化到平板坐标系中才能直接使用。如图4.1所示以支点为原点O，支点与两个电机的连线分别作为X轴与Y轴，建立平板坐标系。

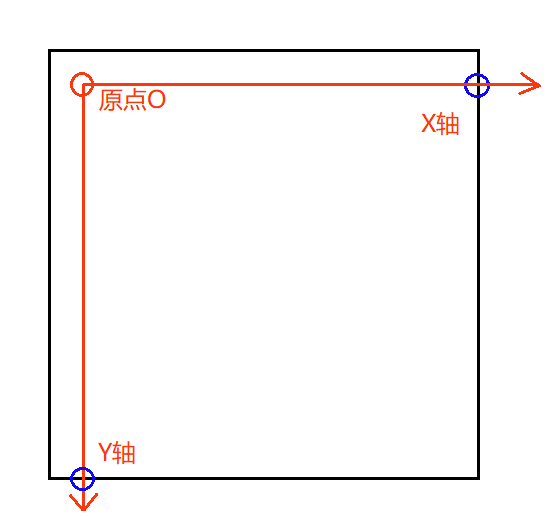
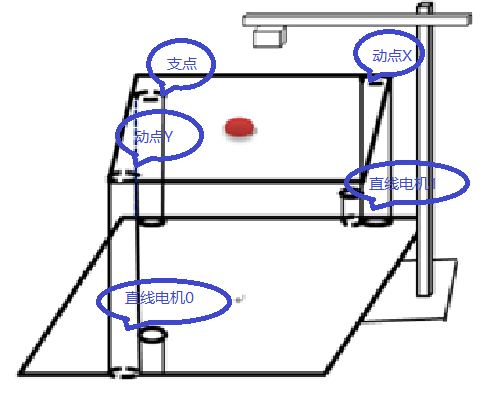


图4.1 板球系统坐标构建

板球系统 X,Y 轴相互垂直，为简化，可以将其看作两个互相垂直方向的球杆，即杆球系统的二维扩展。因为运动的独立性原理，若忽略掉摩擦力，小球在X轴方向上的运动与Y轴方向上的运动彼此独立。所以可以看作两个杆球系统分别控制。

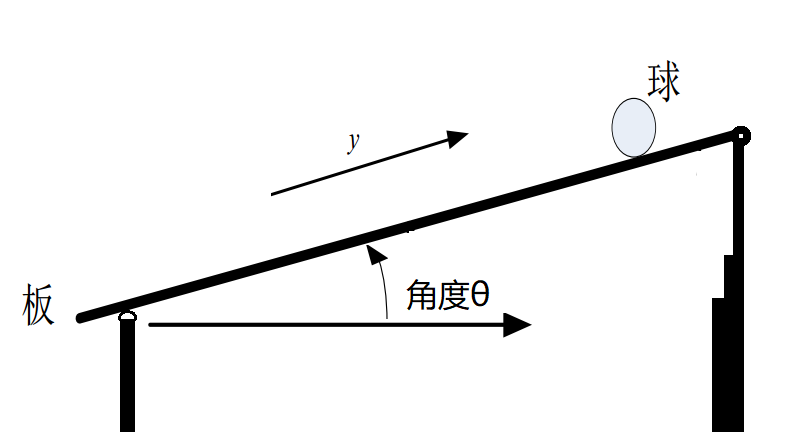


图4.2 板球系统y方向控制模型

如图4.2为小球在Y轴方向上运动的示意图。若不考虑摩擦力容易得出此时小球的加速度a = g\*sinθ，当θ比较小时可做近似计算a = g \*θ。其中g是重力加速度，为常量，因此小球的驱动加速度与平板的倾斜角度之间可看作是近似的线性关系。

## PID控制理论

PID控制是最常用、最广泛使用的自动反馈系统。 PID控制器使用偏差的比例（P，Proportional）、积分（I， Integral）和微分（D，Derivative）来控制被控对象。这里的积分或微分都是偏差相对于时间而言的。

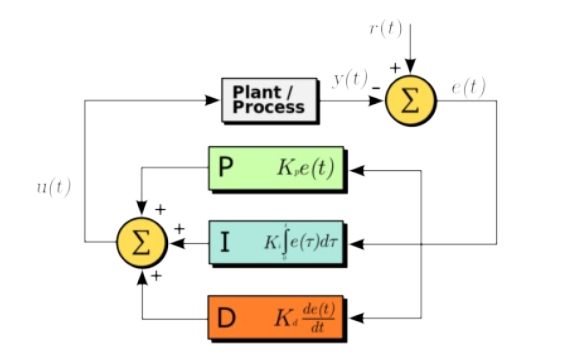


图4.3 PID控制器

对于任何一个自动反馈控制系统来说，有几个基本的性能指标。

1、稳定性(P和I降低系统稳定性,D提高系统稳定性):系统受到某个干扰后,经过一段时间其被控量可以达到某一稳定状态;

2、准确性(P和I提高稳态精度, D无作用):系统处于稳态时,其稳态误差(Steady-state error)。达到稳定状态时与所设定的期望状态之间的误差越小,越准确;

3、快速性(P提高响应速度,I降低响应速度):系统对动态响应的要求。一般由过渡时间的长短来衡量,过渡时间越短，越快速。

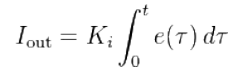
**比例（P）控制**

比例控制是一种最简单的控制方式。其控制器的输出与误差成比例关系。若只有比例控制，系统输出可能存在稳态误差。比例项输出：

 ①

**积分（I） 控制**

只有积分控制时，控制器的输出与误差与时间的积分成正比关系。因此，为了消除稳态误差，常常引入积分项。由于积分项的存在，随着时间的增加，积分项的输出会一直增大。这样以来，即便误差很小，随着时间的积累，积分项的输出结果也会越来越大，直到足以让执行机构运动，从而消除稳态误差。由此，可以设计比例积分控制器，即PI控制器，这样便可以使得系统能够快速进入稳态并消除稳态误差。积分项输出：

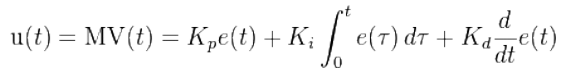
 ②

**微分（D） 控制**

只有微分控制器时输出与误差与时间的微分成正比关系。微分控制的作用对象是误差的变化率，例如在板球控制系统中，需要控制其坐标，而位置坐标的变化率是小球滚动的速度。在这样一个系统中，微分控制的实质是对其速度进行控制，即保证小球运动的速度不至于过大。微分项输出：

 ③

综上所述得到一条公式， 这个就是PID控制数学表达式：

 ④

## 板球系统的PID控制

将板球系统作为PID控制的对象，首先要弄清楚该系统的输入输出及系统特征。本次设计的板球系统，通过OpenMV视觉模块来计算小球坐标，使用直线电机控制平板的倾斜从而控制小球的滚动。也就是说直接驱动的小球滚动的是平板的倾斜角度。因此要想更好地控制小球滚动，必须先控制好平板的倾斜角度。故设计如下图4.4所示的PID串联算法。

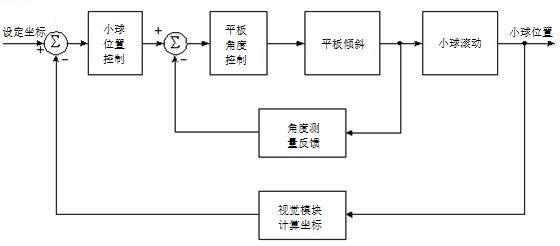


图4.4 板球系统PID控制算法

内环以角度测量作为反馈，其输出通过控制电机来控制平板的倾斜角度。外环将视觉模块计算得到的小球当前坐标作为反馈，与给定坐标进行PID计算，得到的结果为平板倾斜角度的期望值。将其作为内环的输入来控制平板的倾斜，进而控制小球滚动。

在4.1节的讨论中忽略了小球收到的摩擦力的影响，但实际上摩擦力对小球到达目标点的准确性有较大的影响，使得小球难以刚好滚到目标点然后停下。PID中的积分控制项能够消除静态误差，但是将Ki设置过大会导致系统难以稳定，甚至发散，Ki值太小又难以消除误差。为了使小球能够快速且稳定地靠近目标点，然后刚好落在目标点上，设计了如下控制方案：在距离目标点较远时只使用比例和微分控制即KD控制，当到达目标点一定范围内再加入积分控制。这样一来从整体上来说既能保证控制的准确性与稳定性，又能消除稳态误差。

## 画圆算法的实现

上述所讨论的小球控制算法能够轻松地控制小球从起始点大致沿着起始点与目标点的连线滚动到目标点，即直线滚动控制。那么怎么实现弧线滚动控制呢？首先讨论最简单的沿弧线滚动——圆形线路滚动。

要想小球沿着圆形路线滚动可以有两种实现方式：1、将圆近似看作是一个n边形，然后对该n边形的每一条边进行上述的直线滚动控制。2、重新设计算法，在圆的法线方向上进行小球离圆形距离的控制，在圆的切线方向上进行小球滚动速度的控制。

使用方案一可以利用原有的控制算法，但是要想小球滚出的轨迹足够圆，必须将n取得足够大，也就是所对应n边形的边长足够短。在直线滚动的控制中，由于摩擦力的影响，当小球靠近目标点事主要受积分项的控制，应对效率低。所以短距离的滚动控制难以达到快速准确的要求。另外，当小球完成一条边的滚动控制后需要切换目标点的坐标，边长太短时对坐标切换的条件难以判断。故该方案不可取。

方案二的实现方法如图4.5所示。需要控制小球在平板坐标系的滚动轨迹为圆形，不放假设该圆的圆心为（x0, y0），半径为R。为保证R为定值需要在法线方向上用F1加以约束，为了保证小球顺时针滚动就需要在切线方向使用F2控制小球滚动的速度。

x

y

O

(x0,y0)

F1

F2

R

(x1,y1)

图4.5 小球圆形轨迹滚动控制

# 系统调试与评价

## 直线电机调试

在实物调试时发现，在小球距离目标点过近（5cm之内）时，启用直线控制程序电机无反应。后测量中发现，该电机的启动电压（能让电机开始工作的最低电压）约为5V。在5V到12V之间输入电压与电机推杆的行进速度近似为线性关系。而且由于电机的差别每个的电机的启动电机不一致，同一电机上升和下降的启动电压也不相同。所以需要对每个电机进行单独地测试并处理。如图5.1，用宏引入记录电机启动所需要的最小PWM值，然后在程序中加以补偿使电机工作在线性范围内。

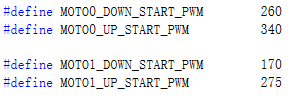


图5.1 电机补偿数值

## 平板水平的调试

将小球放入平板之前必须先将平板调节至水平，但是由于角度传感器的安装和其本身存在的误差，平板手动调整至水平时通过传感器读得的值往往不为0。所以说必须先测量出平板水平时传感器测量的角度值，然后利用该值在程序中加以修正。



图5.2 平板水平时传感器读数记录值

上述记录的值作为水平时角度期望值直接参与PID计算，微小的误差会影响最终控制的准确度，故对其设置的手动调节。在系统调试时小球可能往往没有刚好停在目标点而是偏向固定的一边。这种误差往往是由于记录的水平时角度值与实际偏差导致的。



图5.3 调试时液晶屏显示的数据

如图5.3，图中的zeroAngle即为所记录的平板水平时的角度值，通过按键进行微调，从而减少小球的控制误差。

## 系统联合测试

将参数调节好后进行系统综合测试。将小球放置于起点，坐标为（74,33）设置目标点坐标为（176,135），测得如下表所示数据。已知平板大小为60cm \* 60cm，对应为200\*200的坐标系，故坐标间隔为60 / 200 = 0.3cm。由下表的20组数据可知，装置最终的坐标误差不超过（6,6），即距离误差在6 \* 1.414 \* 0.3 = 2.55cm之内。从起点滚动30.6cm到达并稳定在终点附近用时不超过10秒。基本满足设计的要求。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 起点坐标 | 目的点坐标 | 终点坐标 | 误差 | 时间 |
| 1 | （74,33） | （176,135） | （171,130） | （5,5） | 8s |
| 2 | （74,33） | （176,135） | （172,130） | （4,5） | 9s |
| 3 | （74,33） | （176,135） | （170,133） | （6,2） | 8s |
| 4 | （74,33） | （176,135） | （174,135） | （2,0） | 10s |
| 5 | （74,33） | （176,135） | （174,134） | （2,1） | 8s |
| 6 | （74,33） | （176,135） | （172,134） | （4,1） | 8s |
| 7 | （74,33） | （176,135） | （174,135） | （2,0） | 8s |
| 8 | （74,33） | （176,135） | （175,135） | （1,0） | 9s |
| 9 | （74,33） | （176,135） | （178,133） | （-2,2） | 8s |
| 10 | （74,33） | （176,135） | （179,131） | （-3,4） | 8s |
| 11 | （74,33） | （176,135） | （182,138） | （-6,-3） | 9s |
| 12 | （74,33） | （176,135） | （181,137） | （-5,-2） | 10s |
| 13 | （74,33） | （176,135） | （176,139） | （0,-4） | 7s |
| 14 | （74,33） | （176,135） | （174,134） | （2,1） | 8s |
| 15 | （74,33） | （176,135） | （175,136） | （1,-1） | 9s |
| 16 | （74,33） | （176,135） | （177,135） | （-1,0） | 8s |
| 17 | （74,33） | （176,135） | （179,138） | （-3,-3） | 8s |
| 18 | （74,33） | （176,135） | （180,131） | （-4,4） | 8s |
| 19 | （74,33） | （176,135） | （181,131） | （-5,4） | 9s |
| 20 | （74,33） | （176,135） | （180,132） | （-4,3） | 9s |

# 结论与展望

本设计通过采用现有的理论和单片机技术，通过研究，搭建了一套基于视觉的板球控制系统，并在该系统上设计了一套完整的控制算法。在实验中验证和调试算法，并发现了一些问题，不断地解决问题使系统更加完善，算法的性能也有所提高。最终成功地完成了平台的制作和控制算法的设计，并且控制效果基本满足了设计之初的各项要求。

本文主要完成了以下工作：

1. 对系统的设计要求进行了分析，对整个系统的硬件结构提出了可行的方案；
2. 完成基于视觉的板球控制系统平台的搭建工作，并对各组成模块进行了分析；
3. 提出了一个单片机程序通用设计思路，并将其应用在自己的板球控制系统上。对该系统的程序构架进行了分析；
4. 设计了主控制器与视觉模块的通信协议；
5. 给出了一套完整的板球系统控制思路，并设计且优化了控制算法。分析了板球系统的物理模型，并在该模型上设计了PID控制算法，实现了小球定位控制和圆形轨迹控制。

虽然本系统初步实现了板球控制系统的一些控制算法，但是与国内外的优秀研究成果相比任然有很大的差距，还有许多需要改进的地方：

（1）本系统上实现的控制算法相对简单，使用PID算法不能够应付比较复杂的场景。以后会考虑应用更加先进的控制理论，例如模糊控制，自适应控制等；

（2）本系统采用的电机反应较慢不足以控制小球快速滚动，可考虑优化系统结构加强小球在速度上的控制；

（3）本系统当前实现的功能相对简单，后期考虑加入自动调整参数算法，和按照任意路径滚动算法。