

# 南昌航空大学科技学院

## 专业综合设计报告

题 目： 板球控制系统

姓 名： 闫 佳 伟

专 业： 自 动 化

班 级： 1682022

学 号： 168202238

同 组 人： 曹聪 管意峰 李朋澔

李怡元 张帅

指 导 教 师： 王 振

成 绩：

信 息 电 子 学 部

2019 年 11 月 20 日

# 南昌航空大学科技学院

## 专业综合设计（论文）任务书

### I、专业综合设计(论文)题目：

板球控制系统

### II、专业综合设计(论文)使用的原始资料(数据)及设计技术要求：

板球控制系统，是让一个球从一个点精确的到达另一个点。这里我们采用 USB 摄像头获取数据，反馈给电脑。代码端采用 OPENCV 来进行图像处理，对小球当前位置进行解析并使用 PID 来调控对舵机的角度，然后下发串口指令到舵机，进而使得小球滚动，摄像头在获取位置信息，从而形成一个闭环。用手柄进行模式之间的选择构成人机交互界面，用于控制。

实现中心稳定模式：小球可以保持在板的中心位置保持稳定。

实现手柄控制模式：可以通过手柄来控制小球的运动。

实现红点跟踪模式：小球可以跟着红点进行运动。

实现矩阵运动模式：小球在板上运动，运动轨迹是矩阵形状。

实现圆周运动模式：小球在板上做圆周运动轨迹。

实现窗口控制模式：电脑上有一个窗口，鼠标点击哪里，小球便运动到哪里。

实现依次定点模式：小球在板上的固定好的位置，进行依次运动。

### III、毕业设计(论文)工作内容及完成时间：

2019 年 10 月 23 日-10 月 28 日 确定题目、设计思路、设计要求，进行可行性分析。

2019 年 10 月 29 日-11 月 08 日 任务分工，进行物理框架搭建，实现中心稳定模式。

2019 年 11 月 09 日-11 月 15 日 实现手柄控制模式、红点跟踪模式、矩阵运动模式。

2019 年 11 月 16 日-11 月 21 日 实现圆周运动模式、窗口控制模式、依次定点模式。

---

#### IV、主要参考资料:

---

[1] 庄淼. 计算机 C++ 语言编辑程序中常见问题及编程技巧[J]. 信息技术与信息化  
2019. 10

---

[2] 徐红云, 高英, 曾健. C++ 程序设计基础探究式教学模式研究与实践[J]. 计算机教育, 2019. 09

---

[3] 宋成章, 李哲, 张天凡, 何传邦, 何振. 基于 OpenCV3 的运动对象识别和跟踪研究[J]  
福建电脑, 2015. 10

---

[4] 李健, 潘松峰. WindowsAPI 实现 PC 机与单片机的串行通讯程序设计[J]. 工业控制计算机, 2009. 09

---

信息电子    学部    自动化    专业    1682022    班

学生 (签名):

填写日期:    2019   年   11   月   21   日

指导教师 (签名):

助理指导教师 (并指出所负责的部分):

学部主任 (签名):

## 摘 要

板球控制系统，是一个以图像作为反馈信息，通过电机转动平板的倾斜角度，从而控制小球运动轨迹达到完成自己事先设置好的路径或者图形等运动的实验研究设备。电脑通过从 USB 摄像头中获取的滚球装置中的小球和平板的图像，然后通过电脑对图像的处理算法从而处理所得到的板子与小球的图像，然后通过坐标的变换转换为小球在平板上的所在位置的信息，电脑处理好小球所在位置的位置信息后，通过上位机来控制在平板的两个相互正交方向上分别进行 PID 控制算法的舵机驱动平板在这两个相互正交方向上相互配合地倾斜，从而间接控制小球直线、绕环等运动。再通过 USB 摄像头重新获取板子和小球的图像，循环上述的操作从而达到随心所欲的控制小球完成自己想要的运动轨迹。采用 USB 摄像头，USB 舵机驱动板分别获取图像和控制舵机。系统功能完善没有明显 BUG，测试结果理想可以随时随地地反复进行试验不会出现需要在特殊情况下才能实现试验的情况，很好的实现了事先设置好的各个功能以及后续在试验进行中又新加的几个功能。

**关键词：** PID 控制算法 图像处理 USB 摄像头 USB 舵机驱动

目录

1 方案设计与论证 .....1

    1.1 控制方案描述 .....1

    1.2 机械结构方案设计 .....1

    1.3 摄像头选择方案 .....2

    1.4 控制器选择方案 .....3

2 硬件设计 .....4

    2.1 理论分析与计算 .....4

3 软件设计 .....5

    2.1 总体软件框图 .....5

    3.2 小球识别及处理 .....6

    3.3 控制方法：PID 算法 .....7

4 系统调试与测试 .....8

    4.1 系统调试 .....8

    4.2 功能测试结果 .....8

    4.3 测试分析与结论 .....9

5 总结 .....10

附录 1 .....11

附录 2 .....13

参考文献 .....14

# 1. 方案设计与论证

## 1.1 控制方案描述

图 1.1 是滚球系统的构成框图，主要由控制器、执行器、板和球、摄像头、图像处理模块构成。

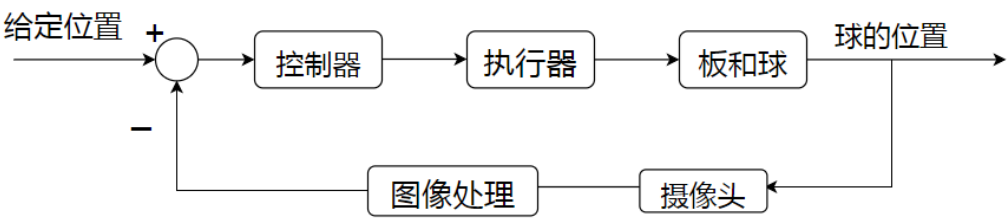


图 1-1 滚球系统构成框图

具体的工作过程为：通过摄像头采集小球的运动图像，在图像处理单元，利用图像处理算法对图片进行处理，获取球相对于板的位置，将位置信息传送给控制器，在控制器内计算控制量，通过控制执行机构来控制平板运动，进而控制小球的运动。

## 1.2 机械结构方案设计

板材采用磨砂亚克力板与舵机传动轴进行刚性连接，既能保证连接处的稳定，又可达到灵活目的。

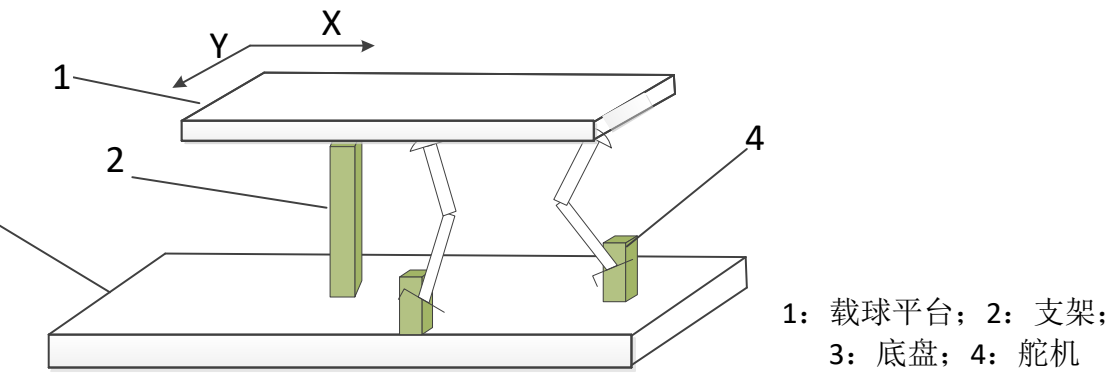


图 1-2 平台机械结构示意图

电机选择方面，既要保证推力够大，能够实现题目基本要求中的轨迹移动、快速制动静止。采用高精度的串行总线舵机，具有大扭力以及迅速。接线方便，可以使用串口指令控制，并且舵机角度可以回读。金属齿轮，具有智能防堵转功能，最大程度上减少舵机因为堵转而烧坏。

小球的运动控制是通过处理器的计算将结果反馈给舵机来 X 方向和 Y 方向的舵机转动，以实现载球平台不同方向的倾斜，从而使小球按照预期的轨迹和规定运动参数在平台上完成相应的动作。

注意：小球的底盘尽量不能出现在垂直 Z 轴方向的转动；底盘与相机画面尽量保持平衡；小球选材时，选择表面越光滑的小球，可控制程度越高。比如设置底盘倾角为  $5^\circ$ ，乒乓球在上面可能会保持静止，而氧化硅小球或铁球可能在  $1^\circ$  的时候就会发生运动。如果可控度低，在控制算法上就要用大量算法解决这个问题。

在此之前规划的整体方案如下：

1. 采用图像处理计算小球位置，来处理舵机转动，使得小球移向下一理想坐标。该方案图像处理方面稍微复杂，但实现效果较好。
2. 采用陀螺仪使得小球定点在某一坐标，目标坐标通过计算角度得到小球滚动速度和时间来定点，再进行稳定。此方案实现过程简单，但精度不够，不能保证精确定点。

故，整体方案选择第一种图像处理的方法，虽然复杂，但实现的准确性更高。

### 1.3 摄像头选择方案

方案 1：OV7725 摄像头，OV7725 型可以硬件二值化的鹰眼摄像头，像素 30 万，传输速率达到 60fps，视场角达到  $63^\circ$ ，拥有很好的低照度。可以满足本系统所需。

方案 2：ov7670 图像传感器，体积小，工作电压低，基本与 ov7725 相同。但是视场角只有  $23^\circ$ ，由于板子边长较大，视场角小的话需要把摄像头抬高，这样影响模型的稳定。并且 ov7670 不能进行硬件二值化处理摄像头采集的图像。

方案 3：USB 摄像头，USB 摄像头像素高，但是需要处理器图像处理快，控制迅速、精准。

## 1.4 控制器选择方案

方案 1：采用 STM32F103RC 单片机，STM32 系列基于专为要求高性能、低成本、低功耗的嵌入式应用专门设计的 ARM Cortex-M3 内核。STM32 单片机程序都是模块化的，接口相对简单些。并且有 STM32 使用经验，开发起来容易上手，但图像处理和控制在一定程度上略显卡顿。

方案 2：采用 K60 单片机，芯片使用多组电源引脚分别为内部电压调节器、I/O 引脚驱动、A/D 转换电路等电路供电，内部电压调节器为内核和振荡器等供电。并且开发环境非常容易搭建，2 路 PWM 控制两个舵机，单片机可控制舵机转动，从而实现小球的移动。

方案 3：采用上位机+USB 舵机驱动，上位机可通过串口来控制总线舵机，也可以读取舵机的角度，设置舵机的工作模式等。控制方便。

综合考虑，由于图像处理需要高速的处理速度，单片机对图像处理略显卡顿，为方便实现功能，采用上位机+USB 摄像头来完成，并且实现控制比单片机效果好，精度高。



## 2 硬件设计

本系统硬件采用 USB 摄像头获取图像反馈给电脑。USB 来下发串口指令到舵机进而来控制小球的滚动。

### 2.1 理论分析与计算

板球系统的机械示意图如图所示。

在板球系统中，要实现小球位置控制，必须明确系统所涉及的四个坐标系：即世界坐标系，球盘坐标系，摄像机坐标系和图像坐标系。球盘坐标系为二维坐标系，是小球位置控制的基准坐标系，板球系统的世界坐标系为三维坐标系。

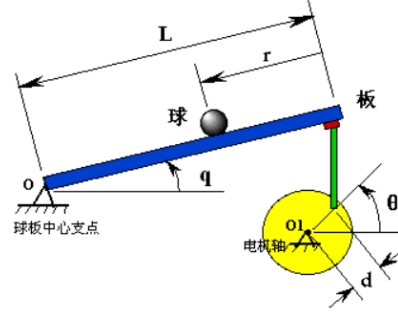


图 2-1 理论分析示意图

对此系统模型，利用牛顿定律或者拉格朗日方程，通过动力学分析，可得：

X 方向：

$$\begin{aligned} & m_b g r_b \sin q_2 \cos q_1 - m_b r_b [(h + r_b) \ddot{q}_2 - y_b \ddot{q}_1 \sin q_2 \\ & - x_b \dot{q}_2^2 - x_b \dot{q}_1 \sin^2 q_2 + (h + r_b) \dot{q}_1^2 \sin q_2 \cos q_2 \\ & - 2\dot{y}_b \dot{q}_1 \sin q_2 + \ddot{x}_b] - I_b ((\ddot{x}_b / r_b) + \ddot{q}_2) = 0 \end{aligned}$$

Y 方向：

$$\begin{aligned} & (m_b g r_b \sin q_1) + m_b r_b [x_b (\ddot{q}_1 \sin q_2 + \dot{q}_2 \dot{q}_1 \cos q_2) \\ & - (h + r_b) (\ddot{q}_1 \cos q_2 - \dot{q}_2 \dot{q}_1 \sin q_2) \\ & + \dot{q}_2 \dot{q}_1 (h + r_b) \sin q_2 - y_b \dot{q}_1^2 + x_b \dot{q}_2 \dot{q}_1 \cos q_2 + 2\dot{x}_b \dot{q}_1 \sin q_2 + \ddot{y}_b] \\ & + I_b ((\ddot{y}_b / r_b) - \ddot{q}_1 \cos q_2 + \dot{q}_2 \dot{q}_1 \sin q_2) = 0 \end{aligned}$$

式中各量的含义如下：

$m_b$  - 小球的质量

$R_b$  - 小球半径

$I_b$  - 小球绕质心的转动惯量

$h$  - 球盘表面距球盘回转中心的距离（见图）

$q_1$  - 球盘绕 X 轴旋转的角度

$q_2$  - 球盘绕 Y 轴旋转的角度

$x_b$ 、 $y_b$  - 小球在球盘坐标系中的位置

以上的非线性模型在采用线性控制理论设计控制器时很难得到应用。当在板球的工作点附近时，我们可以将其线性化，得到以下方程：

$$\begin{aligned} \frac{7}{5} \ddot{x}_b + (\frac{7}{5} r_b + h) \ddot{q}_2 &= g q_2 \\ \frac{7}{5} \ddot{y}_b - (\frac{7}{5} r_b + h) \ddot{q}_1 &= -g q_1 \end{aligned}$$

值得注意的是，线性化的过程同时也是一个解耦的过程。

### 3 软件设计

软件设计是把许多事物和问题抽象起来，并且抽象它们不同的层次和角度。在进行软件设计时，通常把整个过程分成若干个部分，每一部分叫做一个模块。所谓“模块”，实质上就是所完成一定功能，相对独立的程序段，这种程序设计方法叫模块程序设计法。模块程序设计法将复杂的问题分解成可以管理的片断会更容易。将问题或事物分解并模块化这使得解决问题变得容易，分解的越细模块数量也就越多，它的副作用就是使得设计者考虑更多的模块之间耦合度的情况。

#### 2.1 总体软件框图

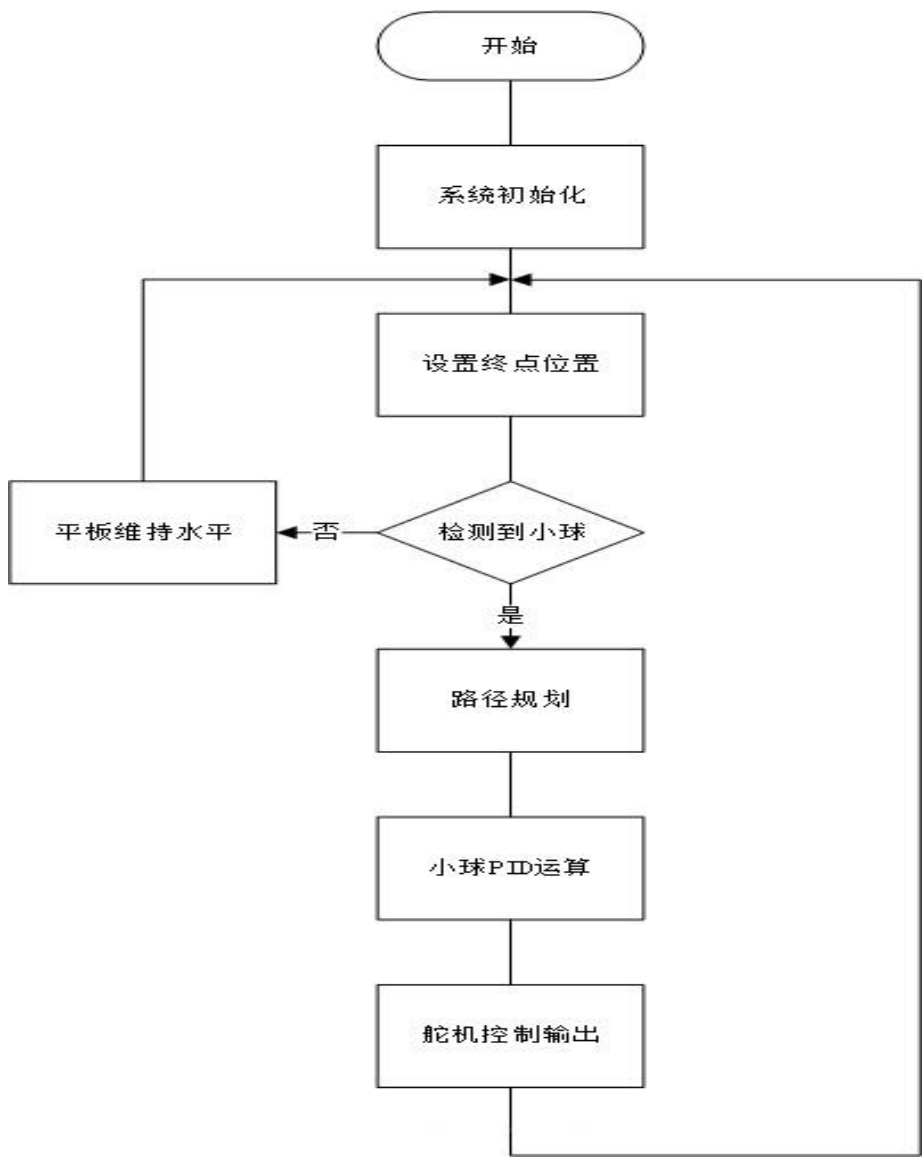


图 3-1 系统流程图

## 3.2 小球识别及处理

小球位置和圆形的检测是控制滚球系统的基础。在图像处理以及模式识别中，经常需要获取图片中圆和矩形的信息，将采集到的图片首先进行灰度化处理，再进行边缘检测、转化为二值图像等操作，具体流程如图 2-1 所示。

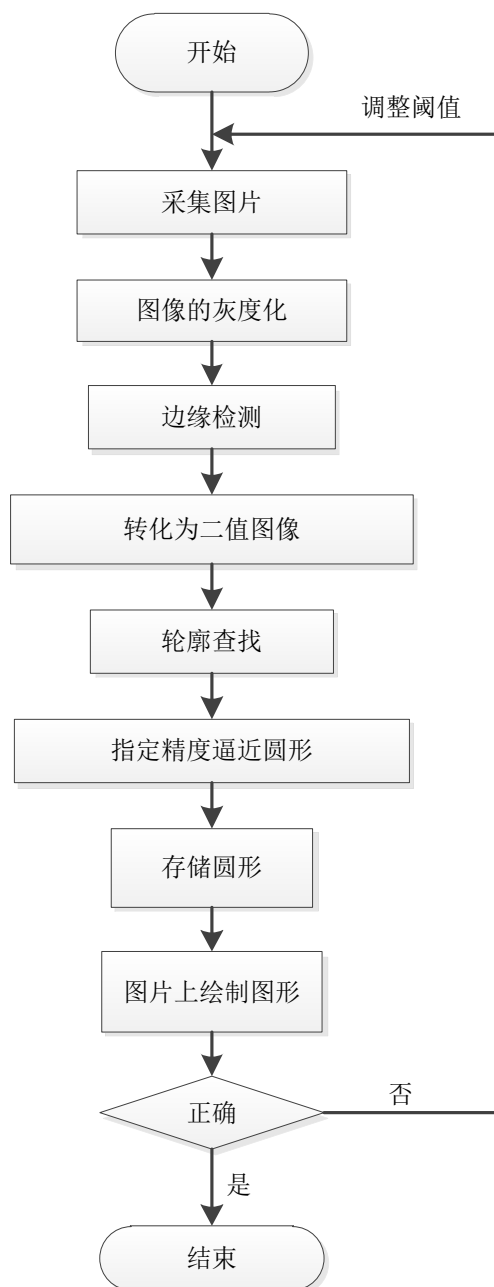


图 3-2 小球识别流程图

找到圆形后，经过转化算得出小球的 X 轴坐标和 Y 轴坐标。小球的速度则是通过两次坐标差算出的。

### 3.3 控制方法：PID 算法

PID 调节器是一种线性调节器，它将给定值  $r(t)$  与实际输出值  $c(t)$  的偏差的比例(P)、积分(I)、微分(D)通过线性组合构成控制量，对控制对象进行控制。

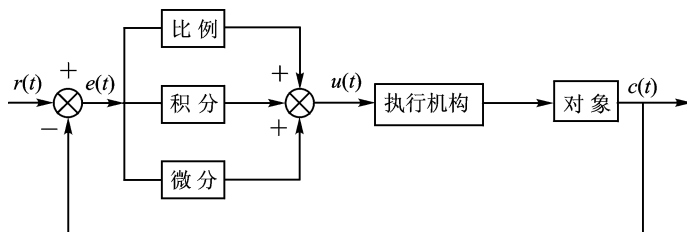


图 3-3 PID 控制系统原理框图

本系统采用的是“X/Y 坐标位置外环——速度内环串级 PID 控制”与“X 轴——Y 轴并级 PID 控制”，结构框图如下。速度环主要起着限速的作用，所以无需积分，控制精度由位置环的积分环节发挥主要作用。调试过程是先把解决速度 PD 环，首先加入比例参数，使关闭微分作用，速度设定值为 0，把小球推置底盘中，观察系统是否可把小球平衡至静止状态，调节幅度过大舵机反复震荡则把比例系数减小，调节幅度过小直接按照原速度方向滚下底盘则把比例系数增大。当系统可把小球稳定在底盘上时则开始调节微分系数。微分的作用是加快系统稳定速度，减少超调量。

外环的调试顺序是 P-D-I，比例系数和微分系数还是根据上述情况去调试，在积分部分加上了积分分离算法和防止积分过饱和算法，积分是为了提高位置控制精度的，如果积分作用有过大的负面作用，可以适当降低精度要求。圆周运动，是在原有控制上加了一个半径外环 PI 算法

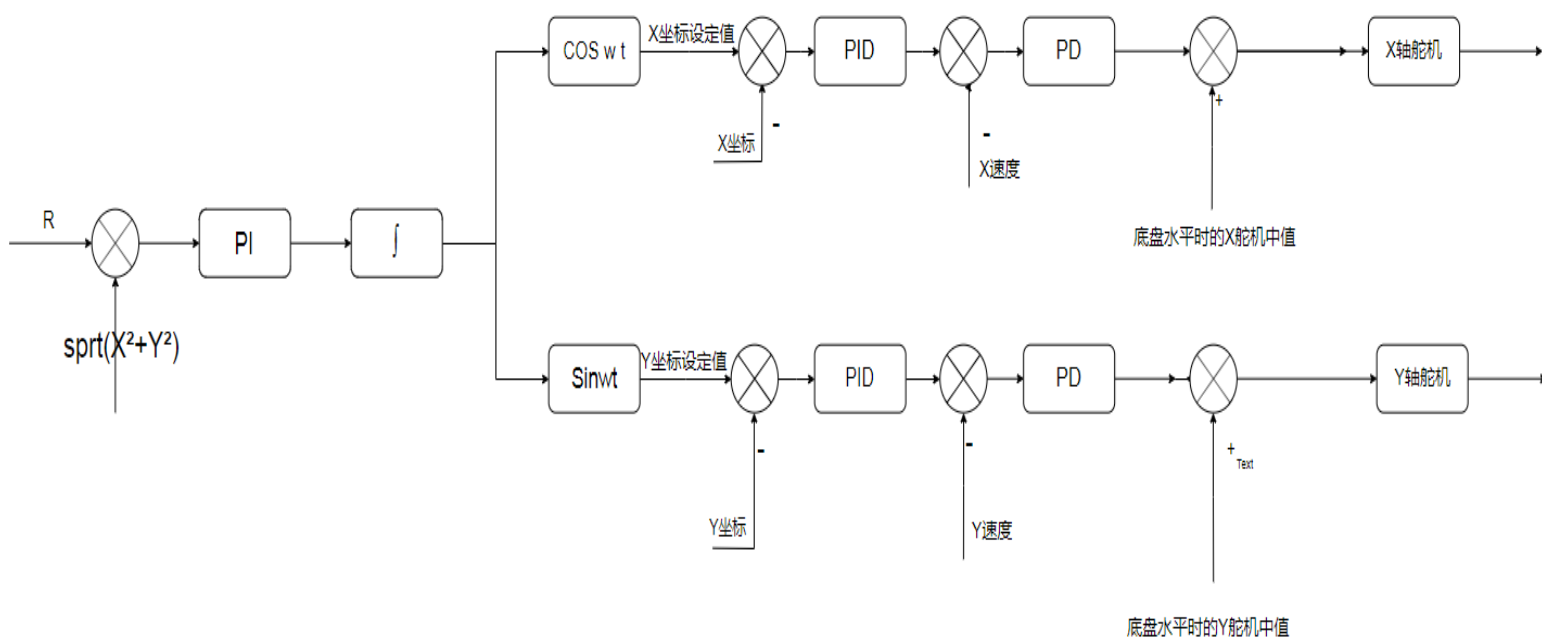


图 3-4 PID 结构框图

## 4 系统调试与测试

### 4.1 系统调试

本程序较大且复杂，因此采用 C/C++ 语言编写，通过软件的不断修改，采用模块调试方法。在进行软件设计时，通常把整个过程分成若干个部分，每一部分叫做一个模块。所谓“模块”，实质上就是所完成一定功能，相对独立的程序段，这种程序设计方法叫模块程序设计法。模块程序设计法将复杂的问题分解成可以管理的片断会更容易。将问题或事物分解并模块化这使得解决问题变得容易，分解的越细模块数量也就越多，它的副作用就是使得设计者考虑更多的模块之间耦合度的情况。

调试步骤：一、先确定好摄像头采集图像数据正常，再确定 OpenCv 的图像处理正不正确。二、确定串口输出数据正确，并且可以正常控制舵机的工作状态。三、确定代码流程正确，没有逻辑问题之后，调整 PID 参数。四、确定模块各模式工作正常。

### 4.2 功能测试结果

按照任务要求，分步测试板球系统的功能。

表 4-1 测试结果

功能模式	试验次数	成功次数	备注
中心稳定模式	10	10	小球比较稳定在 2 号区域
手柄控制模式	10	9	排除干扰和误差，试验结果好
红点跟踪模式	10	9	达到指标且有富余程度
矩形运动模式	10	10	响应迅速，但反光会产生影响
圆周运动模式	10	10	基本上形成圆周运动
窗口控制模式	10	10	每次精准定点在目标坐标
依次定点模式	10	10	和矩形运动原理相似

### 4.3 测试分析与结论

通过一系列的测试结果发现，质量越大的球惯性越大，算法收敛时更趋向稳定，但当算法发散时则容易滚出平板外，如铁球便属于这种材质。而质量小的球如木球，惯性较小，角度稍大容易导致失控，所以我们调节时，比例系数要设定的很低。

系统的主要误差源于摄像头的图像的采集，底架不够稳定会使摄像头跟着平板摆动而摇晃，从而导致获取小球的位置有一定的误差。

## 5 总结

课程设计是教学活动中一个重要环节，课程设计可以使我们了解到一些实际与理论之间的差异。课设是一门锻炼学生综合能力的学习项目，通过课程设计不仅可以巩固专业知识，为以后的工作打下了坚实的基础，而其还可以培养和熟练使用整理资料，运用工具书的能力，把我们所学的课本知识与实践结合起来，起到温故而知新的作用。

在刚刚得到用 PID 的设计任务时，觉得应该很容易！但是此次课程设计题目看似简单，做起来并不十分容易，机械结构的设计要严丝合缝不能有一点误差，所有搭建结构也会比较费时间。必须通过真值表到卡诺图到驱动表达式的步骤，避免死循环的产生！还有设计中各个模块的功能可能是正常的，当电路整体连接起来时，必须考虑到模块间的互相影响，如之前的两个舵机并联，控制不灵敏，当串联起来时结局问题。这些都是设计中需要考虑到问题！

不光是设计过程，在最后写报告的过程中我也深深感受到了这次课设的意义。作为学术性论，内容格式是有很多规定的。即使内容很精辟，格式不过关也不算一个好报告。

通过本系统的设计，使我充分发挥了主观能动性，将大学四年所学的知识以毕业设计的形式应用到了实际当中。这其中，虽然我遇到了很多问题，但通过查阅相关资料、请教老师和与同学讨论将这些问题解决，从而使我对大学四年所学到得知识有了进一步的加深和拓展，相信这一定会对以后走上工作岗位的我有很大的帮助。

总之，通过此次课程设计，我学到了很多专业相关的东西，使自己的能力提升了一个档次，但是还是发现有很多的不足，希望在以后的学习中可以弥补！

# 附录 1

## 小球识别部分代码：

```

/* 完成一次小球查找工作
   查找圆心
   位置赋值
   返回结构体
*/
bool BoardBallControl::GetBallPosition(Mat &frame2, double &x,double &y)
{
    Mat frame;
    cvtColor(frame2, frame, COLOR_RGB2GRAY);      //彩色图像灰度化
    threshold(frame, frame, THRESH_BINARY_C_VALUE,
    THRESH_BINARY_H_VALUE, THRESH_BINARY);      //二值化
    Point position;
    for (int I = 0; I < frame.rows - 1; i++)
    {
        unsigned char *data1 = frame.ptr<uchar>(i);
        for (int j = 0; j < frame.cols - 1; j++)
        {
            if (data1[j] >= 250)
            {
                int begin = j;
                while (data1[j++] >= 250 && (j < frame.cols - 2));
                if ((double)(j - begin)*exchange_value_x <= mindiameter ||
                ((double)(j - begin)*exchange_value_x >= maxdiameter))//不是球，扫描下一行
                {
                    continue;
                }
                position.x = (j + begin)/2;
                position.y = I;
                double radiu = j - begin;
                radiu = exchange_value_x* radiu;
                ball_state = true;
                x  = (double)position.x*exchange_value_x;
                y  = (double)position.y *exchange_value_y;

                cout << "半径=" << radiu << " ";
                cout << "当前坐标: (x=" << x << " , " << "   y=" << y <<
                ") "<<endl;

                //if (fabs(x - position_x) <= 3)x = position_x;

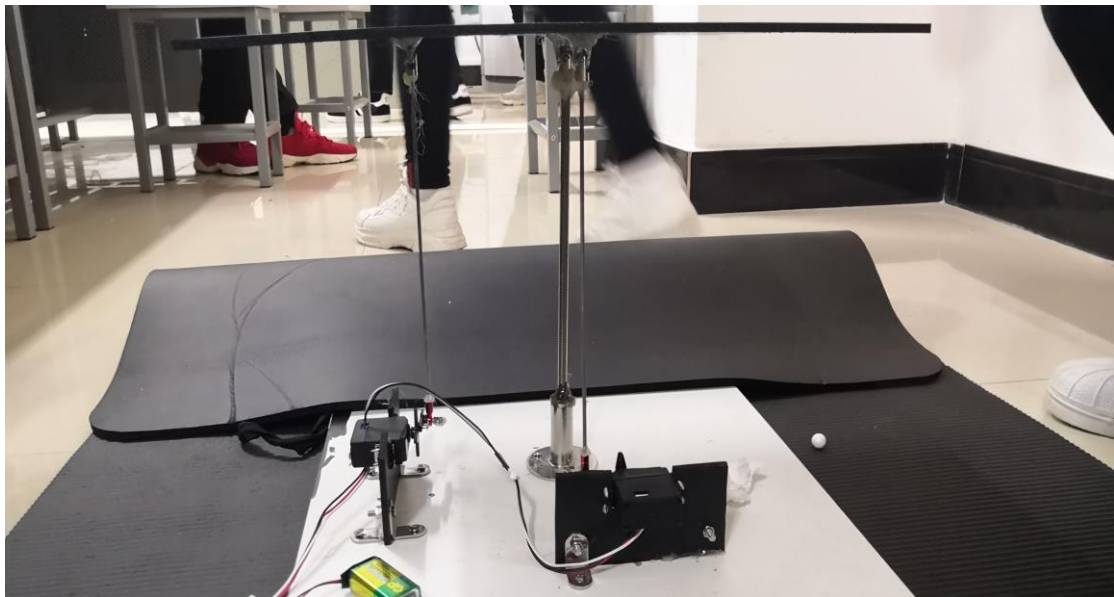
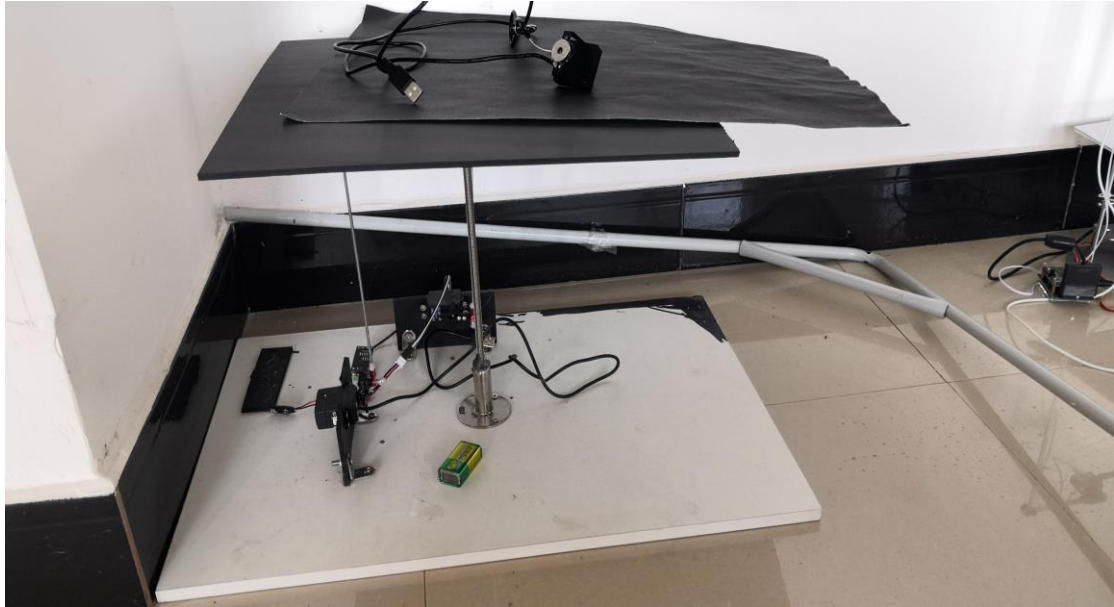
```



```
        //if (fabs(y - position_y) <= 3)y = position_y;
        time_t newtime = clock();
        double time = (double)newtime - lasttime;
        /*假设每次运算时间都相等为 0.5s*/
        time = 1;
        _x_speed = (x- position_x) /time;
        _y_speed= (y- position_y) /time;
        position_x = x;
        position_y = y;
        lasttime = newtime;
        return true;
    }
}
ball_state = false;
return false;
}
```

## 附录 2

实物图:



## 参考文献

- [1]庄淼.计算机 C++ 语言编辑程序中常见问题及编程技巧[J].信息技术与信息化,2019(10):87-89.
- [2]徐红云,高英,曾健.C++程序设计基础探究式教学模式研究与实践[J].计算机教育,2019(09):99-102.
- [3]宋成章,李哲,张天凡,何传邦,何振.基于 OpenCV3 的运动对象识别和跟踪研究[J].福建电脑,2015,31(10):19-20.
- [4]李健,潘松峰.WindowsAPI 实现 PC 机与单片机的串行通讯程序设计[J].工业控制计算机,2009,22(09):51-53.