

**风 力 摆 控 制 系 统**

**结题报告**

**摘 要：**

风力摆控制系统（以下简称本系统），由数据采集模块，处理终端以及控制模块，风力摆模块,三个部分组合而成。整个系统通过STM32微处理器进行全局控制；采用MPU6050整合性6轴运动处理组件采集实时加速度，角速度以及欧拉角；并通过IIC通信协议传输数据，当微处理器接收到数据之后采用PWM脉冲宽度调制结合PID闭环控制达到自动、稳定、精确控制的目的。本系统采用两组轴流风机为其提供动能，使用万向轴提供自由方向，红外激光笔表现其结果。通过理论及验证，本系统可以达到良好的自动控制目的，并且拥有较高的控制精度。在实验基础之上，通过调试测量与创新，达到了相关题目的要求。

**关键字：**风力摆控制系统、PID、STM32微处理器、MPU6050、轴流风机

**Abstract：**

Wind pendulum control system (hereinafter referred to as the system ) , the data acquisition module, processing terminal and a control module , wind swing module , a combination of three parts . The entire system is controlled by a global STM32 microprocessor ; 6-axis motion processing components using MPU6050 collect real-time integrated acceleration, angular velocity and the Euler angles ; and by the IIC communication protocol to transfer data , when the microprocessor receives data using the PWM pulse width modulation with PID closed loop control to achieve automatic , stable , precise control purpose . The system uses two axial fan to provide kinetic energy , the use of universal axis direction to provide free , infrared laser pen performance of its outcome. Through theory and verification , the system can achieve good automatic control purposes , and has high control accuracy. In experimental basis, by commissioning measurements and innovation , to the relevant requirements of the subject .

目录

[1、 参赛题目 4](#_Toc427416756)

[1.1 任务 4](#_Toc427416757)

[1.2 基本要求 4](#_Toc427416758)

[1.3 发挥部分 5](#_Toc427416759)

[2、 题目分析与方案选择 6](#_Toc427416760)

[2.1 题目分析 6](#_Toc427416761)

[2.2 方案设计 6](#_Toc427416762)

[2.2.1 风力摆设计以及运动控制设计方案 6](#_Toc427416763)

[2.2.2 数据采集选择方案 7](#_Toc427416764)

[2.2.3 主控芯片选择方案 7](#_Toc427416765)

[2.2.4 轴流风机选择方案 8](#_Toc427416766)

[2.2.5 电机驱动的选择控制方案 8](#_Toc427416767)

[2.2.6 PID闭环控制方案 8](#_Toc427416768)

[2.3 方案论证 10](#_Toc427416769)

[3、 硬件设计《系统结构以及电路设计》 11](#_Toc427416770)

[3.1 轴流风机受力控制 11](#_Toc427416771)

[3.2 总体硬件框图 12](#_Toc427416772)

[3.3 电路设计 13](#_Toc427416773)

[3.3.1 电机驱动设计原理图 13](#_Toc427416774)

[3.3.2 控制终端模块原理图 14](#_Toc427416775)

[4、 程序设计 15](#_Toc427416776)

[4.1 程序框架 15](#_Toc427416777)

[4.2 MPU6050数据采集程序设计 16](#_Toc427416778)

[4.3 PWM脉冲宽度调制 16](#_Toc427416779)

[4.4 PID闭环控制 17](#_Toc427416780)

[5、 调试结果与分析 18](#_Toc427416781)

[5.1 分块测试模块 18](#_Toc427416782)

[5.1.1 电路测试 18](#_Toc427416783)

[5.1.2风力摆无驱动摆动测试 18](#_Toc427416784)

[5.2 整机测试程序以及数据分析 19](#_Toc427416785)

[5.2.1 基本要求1 画直线 19](#_Toc427416786)

[5.2.2 基本要求2 画定长线段 19](#_Toc427416787)

[5.2.3 基本要求4 风力摆5s静止 19](#_Toc427416788)

[5.2.4 发挥部分1 画圆 19](#_Toc427416789)

[5.3 误差分析 19](#_Toc427416790)

[5.3.1 轴流风机的滞后性 19](#_Toc427416791)

[5.3.2 数据采集滞后性 20](#_Toc427416792)

[5.3.3 电源的影响 20](#_Toc427416793)

[5.3.4 其他因素的影响 21](#_Toc427416794)

[6、 创新与发挥 21](#_Toc427416795)

[7、 总结及感想 21](#_Toc427416796)

[8、 参考文献及资料 21](#_Toc427416797)

[附录 22](#_Toc427416798)

[1．系统元器件清单 22](#_Toc427416799)

[2. 部分要求照片 22](#_Toc427416800)

### 1、 参赛题目

* 1. 任务

一长约60cm~70cm的细管上端用万向节固定在支架上，下方悬挂一组（2~4只）直流风机，构成一风力摆，风力摆上安装一向下的激光笔，静止时，激光笔的下端距地面不超过20cm。设计一测控系统，控制驱动各风机使风力摆按照一定规律运动，激光笔在地面画出要求的轨迹。

* 1. 基本要求

（1） 从静止开始，15s内控制风力摆做类似自由摆运动，使激光笔稳定地

在地面画出一条长度不短于50cm的直线段，其线性度偏差不大于±2.5cm，并且具有较好的重复性；

（2） 从静止开始，15s内完成幅度可控的摆动，画出长度在30~60cm间可设置，长度偏差不大于±2.5cm的直线段，并且具有较好的重复性；

（3） 可设定摆动方向，风力摆从静止开始，15s内按照设置的方向（角度）摆动，画出不短于20cm的直线段；

（4） 将风力摆拉起一定角度（30°~45°）放开，5s内使风力摆制动达到静止状态。

1.3 发挥部分

（1） 以风力摆静止时激光笔的光点为圆心，驱动风力摆用激光笔在地面画圆，30s内需重复3次；圆半径可在15~35cm范围内设置，激光笔画出的轨迹应落在指定半径±2.5cm的圆环内；

（2） 在发挥部分（1）后继续作圆周运动，在距离风力摆1~2m距离内用一台50~60W台扇在水平方向吹向风力摆，台扇吹5s后停止 ，风力摆能够在5s内恢复发挥部分（1）规定的圆周运动，激光笔画出符合要求的轨迹；

（3） 其他。

### 2、 题目分析与方案选择

2.1 题目分析

按照题目的要求：“直流风机是驱动风力摆的唯一动力，不得以任何其它方式影响风力摆的运动” 。我们小组想到的是与示例相似的结构，风力摆由万向节、碳素杆、和轴流风机以及底座组成。在基本要求的基础上，我们队想到的方法是通过类似于“秋千摆动”的方式驱动风力摆，通过万向节特性，我们可以画出一条直线。

题目要求“画出长度在30~60cm间可设置的线段”，首先点明了这是一个可以控制的摆动，并且要求画出的线段在30至60cm之间， 所以我们认为要达到这个距离要求，首先应该通过PWM脉冲宽度调制得到一个近似的距离，再加上PID控制算法应该可以实现。

在题目要求中“15s内按照设置的方向（角度）摆动，画出不短于20cm的直线段”，我们想到的是首先找到一个标志角度，再通过传感器返回的风力摆平台姿态数据，通过比例计算可以达到控制的目的。

在使风力摆平台静止的要求中，我们可以通过传感器判断风力摆当前的姿态，并且通过控制轴流风机使之停止运动，“在5S内使之保持静止”，这个条件还是较有难度。

在这次比赛中，长度均以厘米（cm）为单位，角度以10°为最小单位。赛题中要求的各项动作完成时间越短越好，超过规定时间1倍时不计成绩。

2.2 方案设计

****2.2.1 风力摆设计以及运动控制设计方案****

方案一：采用两组（四个）轴流风机呈 “ **+** ” 型安装，这样设计，有利于本系统在硬件上基本上基于平衡；在题目要求的设计上，可以实现直线摆动以及圆周摆动的要求。

方案二：采用三个轴流风机呈“ T ”型安装，可养的设计，有利于减小风力摆平台的重量，减轻惯性对于风力摆的影响；在题目要求的设计上，同样可以实现直线摆动以及圆周摆动的要求。

由于方案二无法完成单轴流风机的正反控制，若程序出现误差，方案二无法自动快速的恢复，再加上考虑到风力摆在硬件上的平衡，以及更加方便的控制，我们最终决定采用方案一。

2.2.2 数据采集选择方案

方案一：使用L3G4200D角速度传感器，ADXL345加速度传感器，GY-25角度传感器；使用这些传感器可以分别测量到当前的姿态。

方案二：使用MPU6050整合性6轴运动处理组件采集本系统所需要的各种姿态数据。MPU6050整合性6轴运动处理组件，以数字输出6轴或9轴的旋转矩阵、四元数、欧拉角格式的融合演算数据。 具有131 LSBs/°/sec 敏感度与全格感测范围为±250、±500、±1000与±2000°/sec 的3轴角速度感测器(陀螺仪)。 可程式控制，且程式控制范围为±2g、±4g、±8g和±16g的3轴加速器。

经过比较，我们最后决定采用方案二；MPU6050整合性6轴运动处理组件体积小，便于安装，可以一次性读取角度、角速度，加速度九组数据。虽然三组传感器可以安装在最佳的位置读取数据，但是其需要过多的IO口和程序响应时间，在数据实时性和准确性上难以的到保证，所以我们最终放弃方案一。

2.2.3 主控芯片选择方案

方案一：采用美国德州仪器（TI）公司生产的MS430系列单片机作为系统的控制器。MS430是16位单片机。其针对实际应用需求，把许多模拟电路、数字电路和微处理器集成在一个芯片上，以提供“单片”解决方案。片上资源较丰富，功能强大，速度快，且低功耗。

方案二：采用ST公司的STM32F103系列单片机作为系统的控制器。STM32F103是基于Cortex-M3内核的是新型的32位嵌入式微处理器，兼容SPI，IIC等多种通信协议，可以输出30路PWM波，无需外加芯片,经常用于高速信号采集系统和控制领域。

通过比较我们的题目，STM32微处理器更加适合我们的，所以我们选择方案二，STM32型微处理器，使用其PWM脉冲宽度调制控制轴流风机，使用IIC协议读取MPU6050得到的数据。

2.2.4 轴流风机选择方案

方案一：使用NMD-MAT 2415KL-04W-B79(12V/1.5A直流电机)，该电机为直流电机，直径为6cm，质量102g，上电启动时间0.62S达到最快转速，电机本身无法进行PWM脉冲宽度调制。

方案二：使用AVC-DF04056B12U(12V/1.88A直流电机)，该电机为直流电机，直径为4cm，质量145g，上电启动时间1.80S到达最快速度，电机自带PWM脉冲宽度调制。

经过比较，为了系统的时效性要求比较高，所以需要快速的响应，因此我们选择方案一。

2.2.5 电机驱动的选择控制方案

方案一：使用L298N电机驱动，L298N是一种高电压、大电流电机驱动芯片，最高工作电压可达46V，峰值电流可达3A，持续工作电流为2A。可以通过I/O口提供信号方便地控制直流电机，但是容易发烫。

方案二：使用TB6612FNG电机驱动，工作电压在4.5V~15V，峰值电流可达3.2A，单通道工作电流可以到1.2A。无需加散热片，可以直接用I/O口驱动。

综合以上方案和我们的电机选择方案，我们决定使用TB6612FNG，使用单通道可以达到电流的要求。

2.2.6 PID闭环控制方案

PID控制器：（比例-积分-微分控制器，闭环控制），由比例单元P、积分单元I和微分单元D组成。通过Kp，Ki和Kd三个参数的设定。PID控制器主要适用于基本上线性，且动态特性不随时间变化的系统。

PID控制器是一个在工业控制应用中常见的反馈回路部件。这个控制器把收集到的数据和一个参考值进行比较，然后把这个差别用于计算新的输入值，这个新的输入值的目的是可以让系统的数据达到或者保持在参考值。PID控制器可以根据历史数据和差别的出现率来调整输入值，使系统更加准确而稳定。

PID控制器的比例单元P、积分单元I和微分单元D分别对应目前误差、过去累计误差及未来误差。若是不知道受控系统的特性，一般认为PID控制器是最适用的控制器。借由调整PID控制器的三个参数，可以调整控制系统，设法满足设计需求。控制器的响应可以用控制器对误差的反应快慢、控制器过冲的程度及系统震荡的程度来表示。不过使用PID控制器不一定保证可达到系统的最佳控制，也不保证系统稳定性。

若定义u(t)为控制输出，PID算法可以用下式表示：

其中：

K_p：比例增益，是调适参数

K_i：积分增益，也是调适参数

K_d：微分增益，也是调适参数

e：误差=设定值（SP）- 回授值（PV）

t：目前时间

* ：积分变数，数值从0到目前时间t

方案一：只使用比例控制。

方案二：使用比例控制和积分控制。

方案三：使用比例，积分，微分控制。

通过对题目的分析，我们发现在本系统中存在各种各样的延迟作用，就如轴流风机的响应延迟，读取MPU6050存在延迟，单纯的比例控制很难保证调节得恰到好处，完全消除误差，所以我们组决定采用方案二、方案三。

2.3 方案论证

最终根据题目的要求和本组人员的讨论，本系统采用STM32ZET6型微处理器作为控制核心，采用MPU6050整合性6轴运动处理组件采集角速度、加速度和欧拉角，使用两组轴流风机作为唯一平台动力驱动，主要使用学生电源系统供电。STM32微处理器处理MPU6050采集到的数据并进行分析，然后通过PWM脉冲宽度调制控制两组轴流风机的转速以控制方向和摆动幅度。通过PID算法逐渐对本系统进行精确的控制。

通过MPU6050采集到的数据，我们可以使用其反馈的角度值和角速度值对本系统进行控制；若当角速度的值趋近于正负零的时候可以判断轴流风机的姿态接近于最高点，当角度值与标记量进行对比可以判定轴流风机编号，当确定了摆动方向与轴流风机编号之后，就可以通过控制占空比来控制轴流风机的转速达到基本要求。

通过前期制作对应表的方法，可以较为简单快捷的完成基本要求二、三，前期通过手动测试数据并记录成数组的方式可以获取轴流风机PWM占空比与摆动长度和摆动方向的比例关系；那么在维持本系统的稳定方面可以通过PID算法进行调节，使之逐渐趋近于稳定。

在第二、三题上也可以通过对欧拉角合成力矩方向与摆动长度和摆动方向形成的线性关系建立函数计算轴流风机PWM占空比与摆动长度和摆动方向的比例关系；若这样设计在系统的稳定性方面将会有较为大的影响，如果对于轴流风机编号定位不准确的话，在摆动的上升或者下降方面将会有很大的影响。当然我们依然可以用PID算法对本系统进行调节，在调节的难度上将会更难。

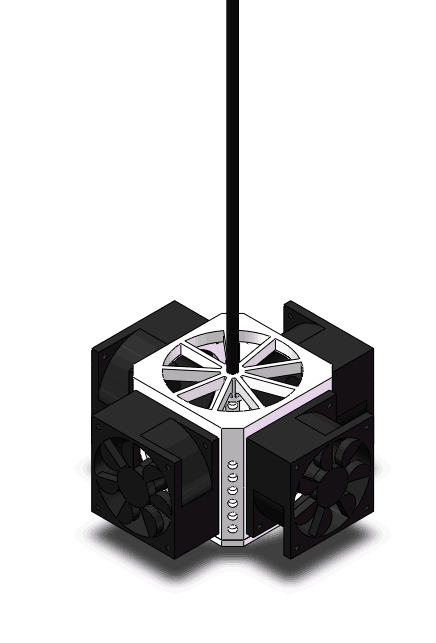
采用两组风机设计，在摆动静止的过程上会有难度的提升，质量与惯性成正比例，在改变运动状态上我们需要更多的力改变运动方向做工，首先，我们考虑通过MPU6050回传的数据读取当前的姿态，判断出当前的角度值，设定反方向上的风机以最大的PWM占空比进行输出，当初始状态为两个轴流风机的合成方向，就要沿着合成角度的反方向其他两个轴流风机进行PWM占空比最大输出。在本次基础要求中，轴流风机的判定姿态问题依然是作重要的，首先，要求在初始状态下判定位置，并且程序立即执行PWM控制，在风力摆将达到对角最高时，要立即对轴流风机进行停止，并且立即对运动方向上的轴流风机加速如此往复达到目的。但是这样在沿着“ **+** ”方向上的风力摆来说达到静止要求的时间更长。所以，在对于沿着“ **+** ”方向上的风力摆达到静止采用另一种算法，首先在第一次下降时，就使用两个与运动方向相同并呈 “ **𠃍** ”形态的轴流风机对系统进行减速操作，如此往复可使得风力摆以更快的速度达到静止状态。

### 3、 硬件设计《系统结构以及电路设计》

3.1 轴流风机受力控制

在轴流风机的的设计中，我们采用 “ + ” 的设计方式，那么两组四个轴流电机以十字形进行出风，由于轴流风机是驱动风力摆的唯一动力，所以在控制上采用PWM脉冲宽度调制信号与PID算法联合进行控制和微处理以及风力摆的姿态调节。在风力摆上可以实现前后左右，以及圆周，单摆等的控制。

风力摆以及轴流风机的安装如图所示：



3.2 总体硬件框

硬件总体设计分为三个部分：数据采集模块，处理终端以及控制模块，风力摆模块。

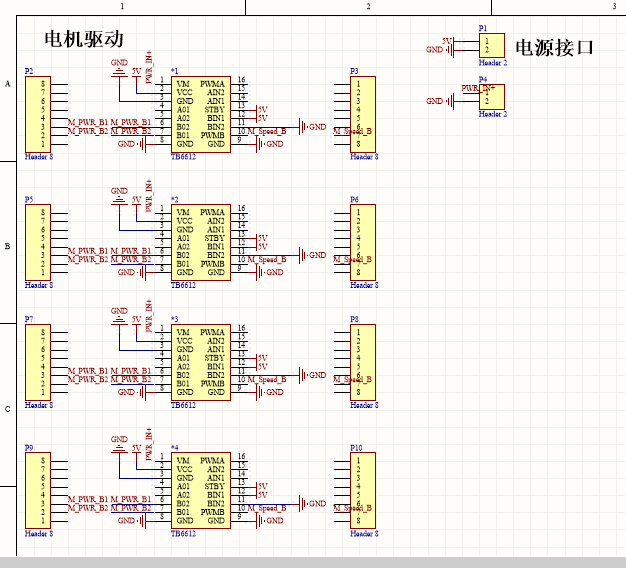
数据采集模块主要由MPU6050组成，使用数据线与STM32连接，采用IIC协议进行数据通信；

处理终端以及控制模块由STM32与OLED，L298N电机驱动组成，STM32定时器产生PWM占空比信号通过控制轴流风机从而控制转速（或者通过控制L298N电机驱动间接控制轴流风机转速）；

风力摆模块由轴流风机和激光探头模块组成，轴流风机是风力摆动力的唯一来源。

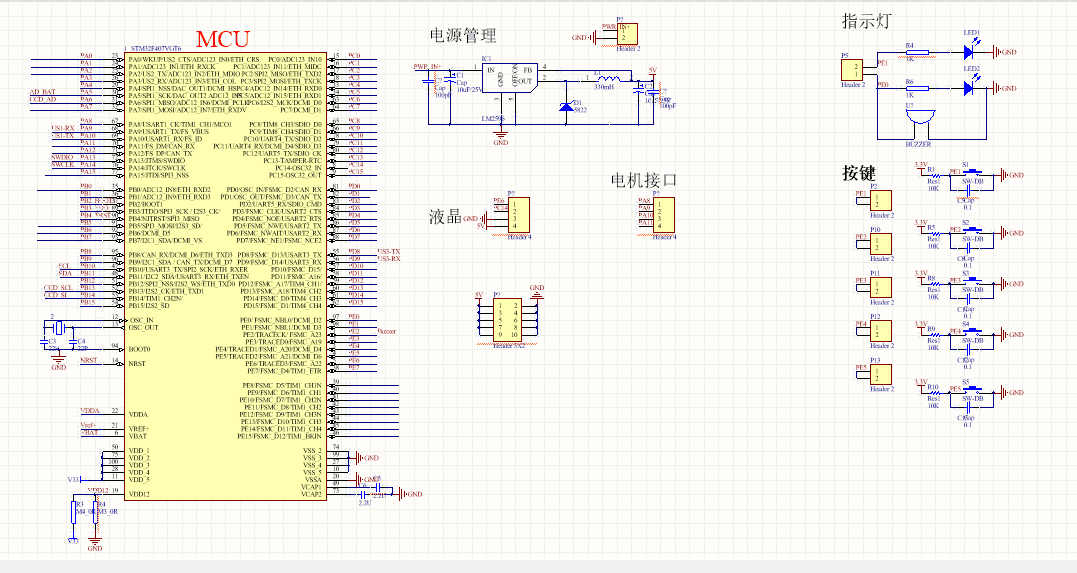
3.3 电路设计

3.3.1 电机驱动设计原理图



电机驱动板由4个TB6612FNG组成，TB6612FNG单个驱动可达15V/1.2A稳定电流。

3.3.2 控制终端模块原理图



终端控制模块由STM32FT03微处理器、电源管理模块、OLED显示器，指示器和蜂鸣器组成。

### 4、 程序设计

4.1 程序框架



用户通过不同的按键选择不同的功能，完成一次功能后复位电路即可选择第二个功能。

4.2 MPU6050数据采集程序设计

MPU6050数据采集的核心代码为Read\_Filter\_MPU6050（）函数，该代码功能为读取MPU6050的加速度、角速度以及欧拉角的值，如果是第一次调用该代码（即开机调用该代码），本段代码会自动计算零偏，而在以后使用则不会计算零偏，该段代码添加滑动滤波功能，以提高程序的稳定性和准确性。

数据采集流程图：



4.3 PWM脉冲宽度调制

Motor\_speed\_ctrl()函数为PWM脉冲宽度调制控制函数，当姿态判断函数得到一个风力摆当前的姿态，那么在此时我们会得到一个相应轴流风机的转速以及将要调制的转速，在调用此函数后，每个风机会自动进行判断并输出PWM占空比。

PWM脉冲宽度控制流程图：



4.4 PID闭环控制

PID闭环控制流程图：



### 5、 调试结果与分析

5.1 分块测试模块

5.1.1 电路测试

5.1.2风力摆无驱动摆动测试

风力摆在原理上是否可行，需要我们进行验证和测试，在我们搭好硬件平台之后，立即对平台进行了手动操作，首先我们尝试一个轴流风机单吹，但是发现并不能使风机摆动起来；在思考之后，我们立即决定采用 “ 秋千摆 ” 的方式将风力摆平台驱动起来，通过依次给轴流风机上电并沿着运动方向推动的方式，在尝试之后，我们发现这样是可行的，那么之后所需实现的功能都应该能实现。

5.2 整机测试程序以及数据分析

5.2.1 基本要求1 画直线

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **基本要求1：画直线** | | | | | |
| **要求：15s内，不短于50cm直线，线性度偏差<2.5** | | | | | |
| **编号** | **时间（s）** | **长度（cm）** | **线性度偏差（±cm）** | **测试用电压（V）** | **备注** |
| 1 | 10.2 | 50 | 1.57 | 11.7 | 10次取平均 |
| 2 | 12.2 | 60 | 1.89 | 11.7 | 10次取平均 |
| 3 | 14.8 | 70 | 1.78 | 11.7 | 10次取平均 |
| 4 | 15.8 | 80 | 1.64 | 11.7 | 10次取平均 |

数据分析：在11.7V的电源下，我们从长度50cm至长度70cm每组测试10次，一共测试了40次，具有一定的代表性。从数据可以看出，我们可以在规定的时间内完成题目一的相关要求，并且在精度控制上还算是不错的。随着摆动距离的上升，摆动的时间上升并且所需要的上升时间也是递减的；在线性偏差度上，由于万向节和硬件平台上可能存在误差，所以存在一定的误差，但是在题目规定之内。

5.2.2 基本要求2 画定长线段

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **基本要求2：画线段** | | | | | |
| **要求：15s内，30~60cm线段，长度偏差<2.5** | | | | | |
| **编号** | **时间（s）** | **长度（cm）** | **长度偏差（±cm）** | **测试电压（V）** | **备注** |
| 1 | 3.4 | 30 | 1.0 | 11.6 | 10次取平均 |
| 2 | 5.1 | 35 | 1.8 | 11.6 | 10次取平均 |
| 3 | 7.6 | 40 | 1.1 | 11.6 | 10次取平均 |
| 4 | 8.4 | 45 |  | 11.6 | 10次取平均 |
| 5 | 10.4 | 50 | 1.5 | 11.6 | 10次取平均 |
| 6 |  |  |  |  | 10次取平均 |

数据分析：在11.6V的电压下，我们从长度30cm至长度50cm每组测试10,次，一共测试50次。从数据上可以看出，长度30cm起摆达到要求时间很快，精度也高，随着长度的增加，长度偏差也会有增加，但是依然在要求范围之内。

5.2.3 基本要求4 风力摆5s静止

5.2.4 发挥部分1 画圆

5.3 误差分析

5.3.1 轴流风机的滞后性

经过测试我们发现，每一款轴流风机都有启动时间并且，不同的轴流风机启动时间是不相同的。我们小组经过测试测出我们轴流风机的时延情况。

以下是时延表：



通过上表，我们可以发现，在独立给每个风机供电的时候，风机一定会有一个启动时间，而且是很难避免的，所以我们小组采用了启动快速的轴流风机，这样在程序的控制上会变得简单，在风机断电之后，风机也不是立即停止，所以通过推算，我们在算法上有改良，即每次减速不减到零，而是保持一个比较小的值，这样可以快速恢复到正常运行状态。

5.3.2 数据采集滞后性

我们所使用的是MPU6050当速度达到某一个值以上的时候，角速度和角度采集反应滞后，若存在这样的误差，那么在PID闭环调节的时候，风力摆的姿态会受到极大的影响。

以下是MPU6050数据采集滞后情况表：



通过上表我们可以的到，在完成半个周期时间小于s时候，角度值和角速度值发生改变存在s的滞后。

5.3.3 电源的影响

我们在实验的时候发现，电源的稳定性会很大的影响系统最后的结果，单电源同时给控制系统和风机供电，方案虽然简单易操作，但风机转动过程中会给电源带来纹波，使得MPU6050读取速度完全不准确；而且单电源在供电的时候可能会有供电不足的情况。

5.3.4 其他因素的影响

试验场地的原因，可能会有不确定的风向，会使得传感器采集的数据不准确；在摆杆和万向节衔接的部分不稳定，会产生抖动；万向节的

### 6、 创新与发挥

### 7、 总结及感想

参加这次大学生电子设计竞赛，让我们收获满满，首先是前期的集训，我是有一点点的基础的，在集训期间我发现我所掌握的知识完全不够用，于是在这段时间我们大量学习并掌握了Altium Designer14这款软件，在微处理器上，学习并掌握了STM32ZET6这款微处理器的大部分功能，在参加比赛的这段日子里，我们感觉时间过得非常的快。在比赛器件清单下发的那段时间里，我们各种猜想题目，在自己的猜想上做自己比赛方向的研究甚至包括猜想平台的制作，在这段时间里，我们按照自己对猜想题目的理解，又学到了自己对于控制方面的理解，在滤波算法，PWM脉冲宽度调制，PID算法实现都有了很大的提高。在比赛题目下来以后，我们小组探讨了方案如何实现，于是在方案和要求上制定了我门自己的解决方案，前后我们一共换了四种解决方案，每一的更换和尝试，我们都会有更清晰的认识。在比赛的这四天三夜里，我们经历过程序错误平台乱摆的失落，经历过要求达成时的欣喜，经历过目不转睛看着激光头研究算法，也经历过各种改造平台的繁忙。总之在这次比赛中，我在嵌入式软件编程方面和控制方面有了较大的能力提升。在这次比赛中十分感谢各位老师给予我们的各种指导，也十分感谢队友给予的帮助和鼓励。

### 8、 参考文献及资料

<https://zh.wikipedia.org/wiki/PID%E6%8E%A7%E5%88%B6%E5%99%A8>（维基百科PID控制器介绍）

<http://d.wanfangdata.com.cn/Periodical_zdhyy201005012.aspx>（万方数据PID参数的意义与整定方法）

<http://baike.baidu.com/view/8879632.htm?fromtitle=mpu6050&type=syn>（百度百科MPU6050整合性6轴运动处理组件）

<https://zh.wikipedia.org/wiki/%E8%84%88%E8%A1%9D%E5%AF%AC%E5%BA%A6%E8%AA%BF%E8%AE%8A>（维基百科，PWM脉冲宽度调制）

<http://www.proewildfire.cn/>（野火论坛，STM32）

<http://bbs.elecfans.com/>（电子发烧友论坛）

<http://wenku.baidu.com/view/71950c6b011ca300a6c390af.html>（百度文库，STM32中文参考手册）

<http://wenku.baidu.com/view/7be6bda989eb172dec63b73e.html>（百度文库，MPU6050datasheet）

### 附录

1. 系统元器件清单

|  |  |  |  |
| --- | --- | --- | --- |
| **系统元器件清单** | | | |
| **编号** | **名称** | **数量** | **备注** |
| 1 | 轴流风机 | 4 |  |
| 2 | STM32FT03最小系统 | 1 |  |
| 3 | TB6612FNG电机驱动 | 4 |  |
| 4 | 激光笔 | 1 |  |
| 5 | MPU6050 | 1 |  |
| 6 | 自制电路 | 2 |  |
| 7 | OLED | 1 |  |
| 8 | 坐标图纸 | 1 |  |
| 9 | 碳素杆 | 1 |  |
| 10 | LED | 1 |  |
| 11 | 蜂鸣器 | 1 |  |
| 12 | 万向节 | 1 |  |

2. 部分要求照片