

探测制导与控制技术专业课程设计
二自由度云台运动姿态控制系统设计

学生学号： 2014040828

学生姓名： 张耀坤

班 级： 2014 探测 1

指导教师： 于秀萍

起止日期： 2017.11.10 — 2017.11.24

哈尔滨工程大学自动化学院

课程设计报告撰写内容

一、设计要求

- 1、掌握 MEMS 传感器 MPU6050 的应用方法；
- 2、掌握 STM32 系统应用，脉冲宽度调制（PWM）信号控制舵机的方法；
- 3、二自由度云台运动姿态控制系统设计。

二、设计方案（要求给出详细的设计思路及其必要的论证）

按照控制系统中输入量的变化规律，二自由度云台运动姿态控制系统可设计为程控姿态控制系统和恒值运动姿态控制系统。其中程控姿态控制系统的输入量是按预定规律，随时间变化的函数，该种系统可用于飞行器姿态角的控制，对其设计要求系统的响应迅速，准确度高。而恒值姿态控制系统的输入量是一个固定常值，即运动姿态不受外界载体运动的干扰而维持恒定角度，该种系统可应用在航拍无人机上的自稳云台上，对其的设计要求系统的抗干扰能力强。因此，该设计题目中对两种系统分别进行设计研究。

控制系统的设计包括硬件设计和软件设计两大部分，程控姿态控制系统和恒值姿态控制系统的区别主要体现在输入量的变化规律不同，需采用不同的参数进行调试，因此两者的硬件基础和软件基本框架大致相似，均可统一设计。

1、硬件设计

硬件系统主要包括电源系统、数字控制器系统、角度测量传感器、执行机构等部分，通过对各部分进行合理的器件选型和方案制定，从而组成整个系统。

1) 电源系统

电源部分主要功能是为系统中的其他部分提供合适的供电电压，工程中常用的供电方案有如下两种。

方案一：使用 220V 交流电作为系统总的电源输入，通过 AC-DC 转换模块得到需要的较高电压的直流电源，再通过 DC-DC 稳压模块分别得到其他任意值的直流电压。该种供电方式适用于固定的设备，输出电流大，负载能力强，不足之处是体积较大，不灵活。

方案二：使用可充电电池供电。根据系统所需的动力电压和系统功耗分别选取相应的电池电压和电池容量。通过 DC-DC 稳压模块分别得到其他任意值的直流电压。该种供电方式体积小重量轻，方便灵活，适用于需要移动的系统。缺点是电池成本较高，且使用寿命有限。

由于该姿态角控制系统仅用于原理验证说明，功耗低，体积小，故以上两种方案均可使用，考虑成本和使用范围等方面因素，决定使用方案一，即用开关电源将 220V 交流电转化为 12V 直流电为系统供电。

2) 数字控制器

数字控制器是系统的关键组成部分，通过对其编程可以实现复杂的控制算法，控制执行机构动作，进而实现系统功能，因此控制器的选取在一定程度上决定系统的性能。控制器的选取一般应考虑其运行速度，外设资源，开发的难易程度等方面。目前较常用的数字控制器有以下几种：

方案一：51 单片机。STC89C51 单片机价格便宜，但运算速度较慢，内部资源较少，功能单一，逐渐被市场淘汰。而增强型的 51 单片机 STC12C5A，较 STC89C51 在性能方面有很大改善，16 位字长，运行频率 35MHz，同时内部自带 60K FLASHROM，而且 STC 系列单片机支持串口程序烧写，对开发设备的要求很低，开发时间也大大缩短，对于有一定编程基础的人较容易使用。

方案二：STM32 单片机。STM32F103 系列单片机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，32 位字长，时钟频率达到 72MHz，同时功耗较低。内部自带 12 位 ADC 单元，6 通道 DMA，多达 8 个定时器，以及强大灵活的特定控制接口。控制和计算功能十分强大，是同类产品中性价比较高的产品，但系统开发需要有较强的编程能力和微机硬件基础。

方案三：Arduino Uno 控制器。Arduino 是一款便捷灵活、方便上手的开源电子平台，包含硬件（各种型号的 Arduino 板）和软件（Arduino IDE）。硬件部分是可以用来做电路连接的 Arduino 电路板；软件是 Arduino IDE，即在计算机中的程序开发环境。只要在 IDE 中通过简单的语言编写程序代码，将程序上传到 Arduino 电路板后，电路板便执行相关的操作。该控制器几乎不需要任何编程基础，适用于简单的机电控制系统。

综上所述，为了达到更好的控制效果，选用 STM32F103 单片机作为控制器，通过软件编程实现控制算法，达到姿态角控制的目的。

3) 角度测量传感器

陀螺仪作为惯性器件的一种，常用来测量角速度和角度，而 MEMS 陀螺由于成本较低，在对角度精度要求不高的工业和民用产品中最常使用。按照题目要求选用 MPU6050 作为角度测量的传感器。

MPU6050 是一种利用 MEMS 技术集成三轴陀螺仪，三轴加速度计的运动处理组件，通过 IIC 总线通信，输出测量的各个轴向的角速度和加速度的 ADC 值，通过对其输出值的解算，可以得到其相对于惯性空间的姿态角度。MPU6050 及其同类产品以其超小体积和封装类型，在智能型手机、

手持型游戏产品、3D 遥控器、可便携式导航设备等应用广泛。由于 MPU6050 的解算涉及到较复杂的导航解算算法，不在本题目设计要求范围内，因此选用已解算好角度，通过串口通信输出数据的 MPU6050 模块，直接通过串口读入角度信息实现本系统的角度测量反馈。

4) 执行机构

执行机构通常包括电动机和相关的机械结构，其作用于被控对象，实现控制作用。执行机构选取时常考虑负载的大小，实现的难易程度，成本等因素，有以下几种方案：

方案一：电动舵机。电动舵机使用 PWM 信号驱动，内部通过电位器反馈，实现信号脉宽和旋转角度的一一对应，操作简单，响应迅速，常用于对角度的控制。

方案二：带编码器的直流电机。直流电机反应迅速，负载能力强，通过与电机同轴连接的编码器可以实现对电机位置的控制，进而实现对角度的绝对控制，控制精度取决于编码器的精度。该种角度控制方案精度高，控制效果好，但对电机的位置控制本身就是一个闭环反馈系统，若再用其构成该系统，则整个系统属于串级控制，较复杂。

综上所述，选取方案一，使用两个电动舵机，通过一定的机械连接实现两个方向的旋转控制。

5) 系统原理总体框图

综合上面对系统各个部分实现方案的选择，设计系统总体原理框图如图 2.1 所示。

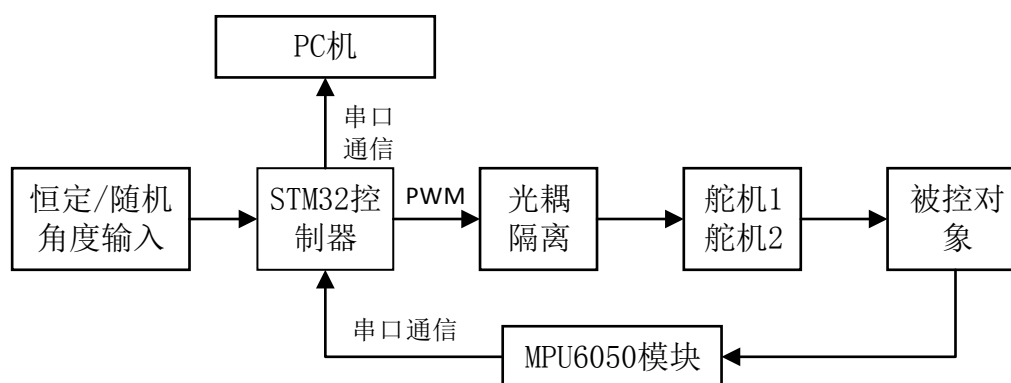


图 2.1 系统总体原理框图

如上框图所示，STM32 控制器根据输入的姿态角设定值和 MPU6050 模块反馈的被控对象姿态角的当前值之间的偏差进行控制，控制量以 PWM 波的形式输出，经光耦隔离后，驱动舵机转动以调整被控对象当前的姿态角，直到达到设定的姿态角。同时 STM32 控制器通过串口通信方式向 PC 机发送被控对象当前的姿态角信息，在 PC 机串口助手的上位机软件中可实时查看并保存姿态角数据，用于分析。

2、软件设计

计算机控制系统软件一般包括采样、滤波、标度变换、控制输出、中断、计时、参数修改及曲线显示等功能。因此实时控制软件设计主要包括数据采集及数据处理程序、控制算法程序、控制量输出程序、中断处理程序、数据管理程序和数据通信程序等程序模块的设计。结合本系统的特点，软件的基本结构采用主程序加中断服务程序的形式，并将程序分为主函数，姿态角数据采集与处理，角度控制算法等部分，其中姿态角数据采集与处理程序在 DMA 中断中进行，角度控制程序在定时器中断中进行。程序总框架图 2.2.1 所示。

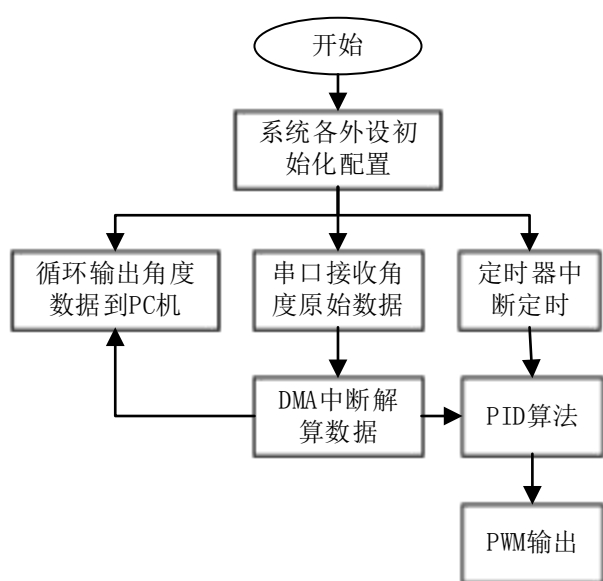


图 2.2.1 程序总体框架图

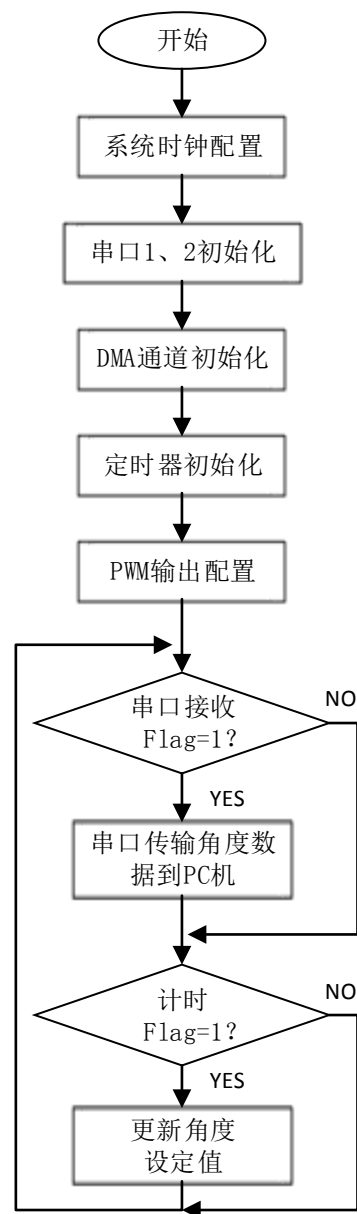


图 2.2.2 主函数程序流程图

1) 主函数

主函数主要完成全部硬件资源的初始化配置和向上位机输出角度信息的功能。主函数程序流程图如图 2.2.2 所示。

2) 姿态角数据采集与处理

姿态角数据是通过串口接收原始数据，在串口接收中断里对数据进行辨识，找到数据帧头后开启 DMA 传输，直接将数据保存到内存，一组数据包传输完成后触发中断，在中断中对数据进行校验，若数据正确无误则对其进行解算，最终得到角度信息。姿态角数据采集与处理程序的流程图如图 2.2.3 所示。

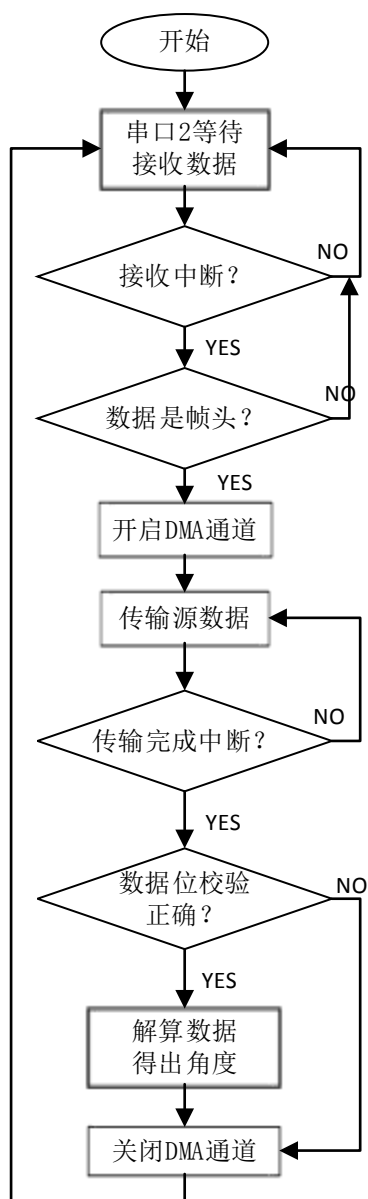


图 2.2.3 姿态角数据采集与处理程序流程图

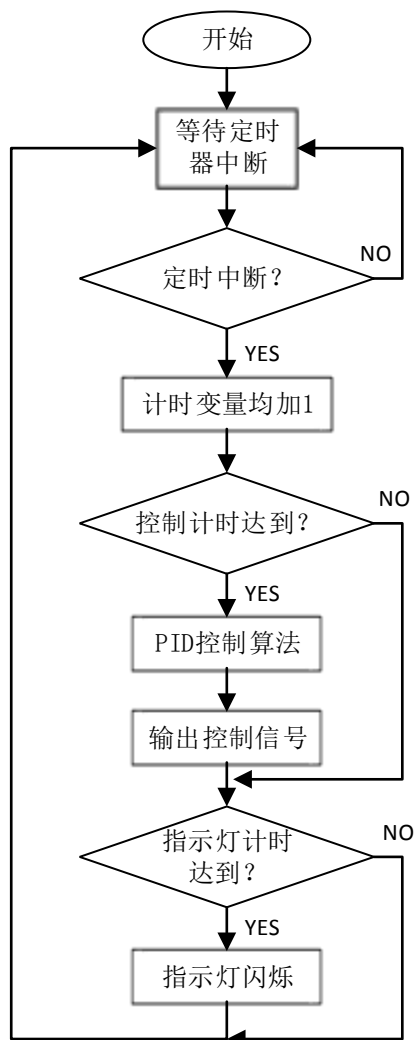


图 2.2.4 角度控制程序流程图

3) 角度控制算法

在定时器中断中执行角度的 PID 控制，并输出控制量，改变 PWM 信号的脉宽，达到控制舵机的目的。角度控制程序流程图如图 2.2.4 所示。

三、设计内容（包括具体的原理和电路，相关的硬件设计和软件程序等，可以结合图、表等形式说明，分析）

1、硬件实现

根据硬件系统的选型方案，对各个模块分别进行具体的设计，主要包括 STM32 控制器最小系统的设计、电源系统设计、光电隔离模块设计等内容。

1)STM32 最小系统

为了简化系统设计，使用 STM32F103C8T6 最小系统模块和配套下载仿真器 J-link-OB 模块。其中最小系统模块使用 3.3V 直流电压供电，外接 8MHz 晶振和硬件复位电路，并通过排针引出所有 I/O 口。根据 STM32F103C8T6 的芯片手册并结合系统硬件资源的需求，分配片上硬件资源情况如下表 3.1 所示。

外设	名称	引脚	功能	配置情况
串口	USART1	PA9	向上位机发送数据	波特率 115200, 8 位数据位, 1 位停止位, 无奇偶校验
	USART2	PA3	接收 MPU6050 模块的数据	波特率 115200, 8 位数据位, 1 位停止位, 无奇偶校验, DMA 使能
定时器	TIM3	——	定时中断, 执行控制	定时周期 2ms
	TIM4	PB6, PB7	输出 PWM 信号	频率 50Hz, 脉宽 0.5ms-2.5ms
DMA	DMA1_Channel6	内部总线	直接传输串口 2 的数据	普通模式, 8 位数据宽度
I/O 口	I/O	PC13	驱动 LED 指示灯	推挽输出

表 3.1 STM32 片上硬件资源分配

2) 电源系统设计

系统总电源使用 220V 交流输入，通过开关稳压电源分别转化为 12V/1A 直流电源作为控制电源输入；5V/3A 直流电源作为舵机动力电源输入。由于控制器和传感器模块分别使用 5V 和 3.3V 供电，因此，通过 DC-DC 降压模块 LM2596，将 12V 电压转化为 5V 电压，进一步使用 DC-DC 降压芯片 MIC5219 将 5V 电压转化为 3.3V 电压。整个电源系统的结构图如下图 3.1.1 所示。

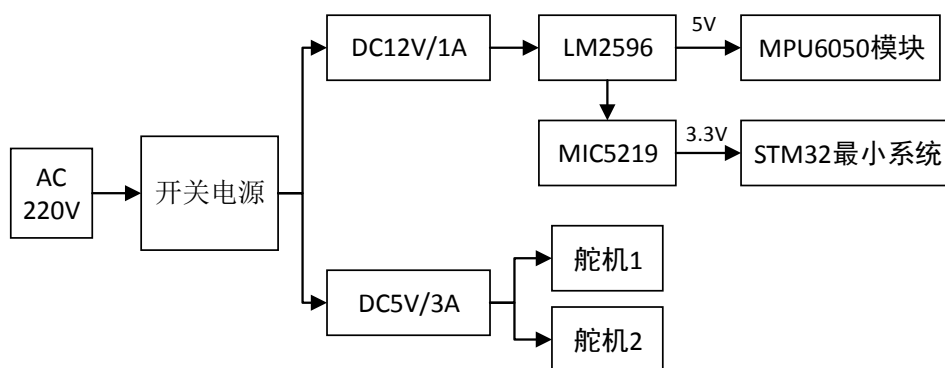


图 3.1.1 电源系统结构图

3) 光电隔离

光电隔离通常将电子信号转换为光信号，在另一边再将光信号转换回电子信号。如此两边电路互相隔离开来，从而防止驱动电路的高电压、脉动电流对微控电路的影响。由于要通过信号的频率为 50Hz，因此选用低速光耦 TLP521。光电隔离单元电路图如图 3.1.2 所示。

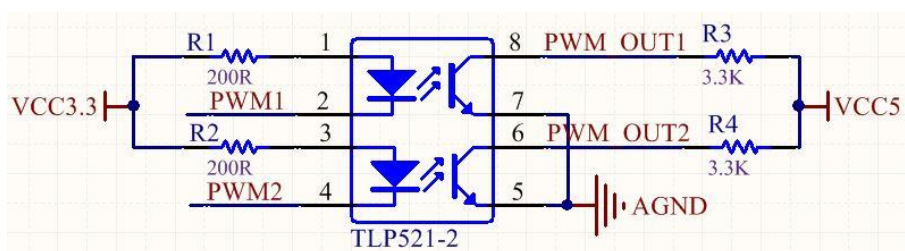


图 3.1.2 光电隔离电路

4) 系统电气连接图

综合以上各部分硬件设计，绘出硬件系统简要电气连接图如下图 3.1.3 所示。

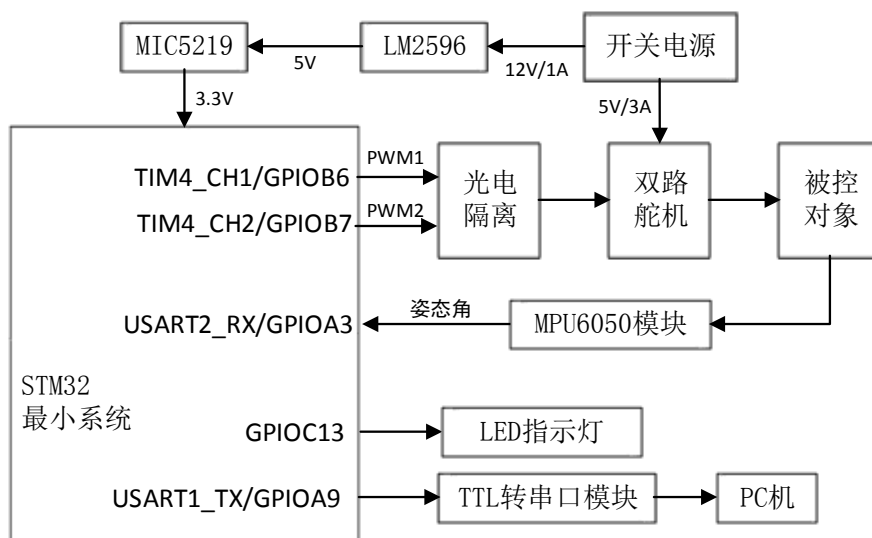


图 3.1.3 系统简要电气连接图

2、软件实现

根据软件设计思路，在 MDK4.29 编程环境下，对各部分进行编程实现。给出系统中较为关键的 PWM 信号输出配置程序，角度数据解算程序，数字 PID 控制程序，中断服务程序等功能函数。

1) PWM 信号输出配置程序

PWM 是 Pulse Width Modulation 的简写，也就是脉宽调制，把恒定的直流电压调制成频率一定、脉冲宽度（占空比）可变的电压信号。不同占空比的 PWM 信号如图 3.2.1 所示。

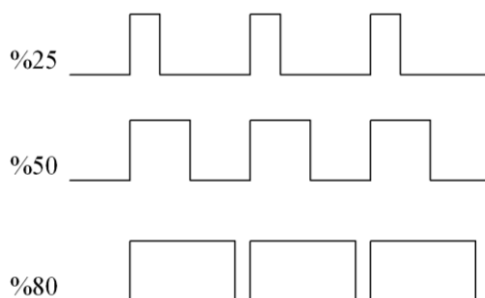


图 3.2.1 PWM 波

PWM 产生方法可以利用定时中断，基本思路是用一个定时器做基本单元定时，每次基本定时到时，则在中断程序中将高电平计数减 1，直到高低平计数完，开始低电平计数，周而复始。而 STM32 上自带 PWM 信号发生单元，基本原理是利用高速时钟的周期 T_0 为基本时间单元，通过设定周期寄存器和比较寄存器输出 PWM。假设周期寄存器设定的值为 N ，则 PWM 波的周期 $T = N \cdot T_0$ 。当启动 PWM 单元工作后，对应定时器工作，当定时器的计数值与比较寄存器相等时对应输出管脚的电平发生翻转，当计数结束又从新开始，周而复始，生成 PWM 波。用中断模拟的方式要占用 CPU 的资源，因此使用 STM32 自带的 PWM 发生单元产生 PWM 波信号。相关程序如下：

```
void TIM4_PWM_Init(u16 arr,u16 psc)//arr: 自动重装值; psc: 时钟预分频数
{
    RCC->APB1ENR|=1<<2;    //TIM4时钟使能
    RCC->APB2ENR|=1<<3;    //IO端口B的时钟使能
    GPIOB->CRL&=0X00FFFFFF; //PB6,PB7
    GPIOB->CRL|=0XBB000000;  //复用功能输出
    GPIOB->CRH&=0XFFFFFF00; //PB8,PB9
    GPIOB->CRH|=0X000000BB;  //复用功能输出

    TIM4->ARR=arr;           //设定计数器自动重装值
    TIM4->PSC=psc;           //预分频器设置
    TIM4->CCMR1|=0X0D<<3;   //CH1输出,PWM1模式,预装载使能
    TIM4->CCMR1|=0X0D<<11;  //CH2输出,PWM1模式,预装载使能
    TIM4->CCER|=0X1111;     //OC1,2,3,4 输出使能,输出极性为高电平有效
    TIM4->CR1=0x0080;       //向上计数,自动重载使能
    TIM4->CR1|=0x01;        //使能定时器4
    TIM4->CCR1=4500;        //初始装载值
    TIM4->CCR2=4500;        //初始装载值
    //PWM输出模式1下, 占空比的计算方法为CCR1/ARR*100%
    //PWM输出模式2下, 占空比的计算方法为(1-CCR1/ARR)*100%
}
```

由于舵机的控制信号是周期 20ms，脉宽 0.5ms~2.5ms 的 PWM 波信号，因此程序中配置的 PWM 波输出频率是 50Hz，比较装载值设为 60000 时，对应的脉宽等效值范围为 1500~7500。对于量程是 180 度的舵机，若 1.5ms 脉宽的 PWM 信号对应机械零位，则通过调节 PWM 信号的脉宽，可以控制舵机在-90 度~+90 度绝对角度范围内的旋转，从而控制载体的姿态角。考虑机械安装的误差，为了使载体上电初始位置是水平的，应先调节舵机角度，找到对应载体水平时的脉宽，作为中值零位脉宽。另外，由于陀螺仪大角度下各轴角度耦合变化，因此，将双轴舵机的旋转角度均要使用软件限位。

2) 角度数据包的解算算法

MPU6050 模块通过串口输出载体相对于惯性空间的角速度，角度，加速度等信息，因此只需要配置好串口通信程序，接收数据即可得到该系统需要的姿态角。为了保证传输过程中数据的准确性，数据经过特殊通信协议的处理加工，因此得到的数据需要经过相反的解码过程才能够得到想要的角度信息。串口通信的协议分为三个数据包，其中角度数据包协议内容如下图 3.2.2 所示。

数据编号	数据内容	含义
0	0x55	包头
1	0x53	标识这个包是角度包
2	RollL	X 轴角度低字节
3	RollH	X 轴角度高字节
4	PitchL	y 轴角度低字节
5	PitchH	y 轴角度高字节
6	YawL	z 轴角度低字节
7	YawH	z 轴角度高字节
8	TL	温度低字节
9	TH	温度高字节
10	Sum	校验和

图 3.2.2 角度数据包协议

根据协议规定，编写角度解算程序如下：

```
void Update_Angle()//串口读取6050数据包并解算////标志位sign_usr2
{
    Angle_Now.Angle_X= -(((short)(D_6050[24]<<8| D_6050[23]))/32768.0*180)-Angle_Init[0];
    //X轴滚转角(x轴)除去初始偏置
    Angle_Now.Angle_Y= -(((short)(D_6050[26]<<8| D_6050[25]))/32768.0*180)-Angle_Init[1];
    //Y轴俯仰角(y轴)
    Angle_Now.Angle_Z= -(((short)(D_6050[28]<<8| D_6050[27]))/32768.0*180)-Angle_Init[2];
    //Z轴俯仰角(z轴)
}
```

为了避免系统上电时载体抖动或者放置不平等因素带来的角度误差，三个轴向的角度分别采集 100 次，做算术均值滤波后作为载体启动时的角度初始值，并在以后的角度解算中统一减去，从而保证三个轴向的姿态角均是以 0 为初值。

另外，由于三轴是耦合的，只有在小角度是才会表现出独立性，在大角度时姿态角度会耦合变化，比如当 X 轴接近 90 度时，即使姿态只绕 X 轴转动，Y 轴的角度也会跟着有较大的变化，这是欧拉角表示姿态的固有问题。因此为了减小该现象的影响，将两个轴向的姿态角均限制在大约-60 度~+60 度之间。

初值计算具体程序如下：

```
if(sign_usr2)
{
    if(Num>100)
    {
        GYRO_Flag=0;
        Num=100;
        Angle_Init[0]=(float)(Angle_Sum[0]/100);
        Angle_Init[1]=(float)(Angle_Sum[1]/100);
        Angle_Init[2]=(float)(Angle_Sum[2]/100);
        printf("%f %f %f \r\n",Angle_Init[0],Angle_Init[1],Angle_Init[2]);
        printf("ok\r\n");

        TIM3->CR1|=1<<0; //使能计数器,开始PID控制
    }
    if(GYRO_Flag)
    {
        Num++;
        Angle_Sum[0]+=Angle_Now.Angle_X;
        Angle_Sum[1]+=Angle_Now.Angle_Y;
        Angle_Sum[2]+=Angle_Now.Angle_Z;
    }
    else
        printf("%f %f \r\n",Angle_Now.Angle_Y,Angle_Now.Angle_Z);
    sign_usr2=0;
}
```

3) 数字 PID 控制算法

离散的位置式 PID 算法：

$$u(k) = K_p \{e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} (e(k) - e(k-1))\} \quad \text{式 3.1}$$

其中：\$e(k)\$ 是第 \$k\$ 次采样时的偏差；\$e(k-1)\$ 是第 \$k-1\$ 次采样时的偏差。

增量式为：

$$\Delta u(k) = K_p \{[e(k) - e(k-1)] + \frac{T}{T_i} e(k) + \frac{T_d}{T} [e(k) - 2e(k-1) + e(k-2)]\} \quad \text{式 3.2}$$

增量式可以写为位置式：

$$u(k) = u(k-1) + \Delta u(k) \quad \text{式 3.3}$$

其中 \$u(k-1)\$ 是第 \$k-1\$ 个控制周期里输出的控制量。

根据式 3.1，3.2，3.3 编写 PID 算法函数如下。


```

//增量式PID算法
void Increment_PID(PID_IncrementType* PID)
{
    PID->d_CtrOut = PID->Kp*(1+T/PID->Ti+PID->Td/T)* PID->ErrNow
                - PID->Kp*(1+2*PID->Td/T) * PID->ErrLast
                + PID->Kp*(PID->Td/T) * PID->ErrPrev;//增量式PID算法

    PID->ErrPrev = PID->ErrLast; //存储误差，用于下次计算
    PID->ErrLast = PID->ErrNow;  //

    PID->CtrOut += PID->d_CtrOut;
}

```

舵机的电压输出与角度输出近似成一级惯性环节，因此采用PI控制规律。因此PID调试参数中Td设为0，只需反复调节Kp和Ti，直到达到很好的控制性能。给出两通道角度PI控制及输出函数如下：

```

//Pitch绕Y轴上层电机，YAw绕Z轴，下层电机
extern volatile Angle_XYZ Angle_Exp;///位置环输出的期望角度
PID_IncrementType PID_Pitch,PID_Yaw; //外环：角度PID环
s16 Motor_Value[2]={0,0};//均赋舵机中值，范围1500-7500

void PID_Angle(Angle_XYZ Now_angle, Angle_XYZ Exp_angle)
{
    PID_Pitch.ErrNow=Exp_angle.Angle_Y -Now_angle.Angle_Y;
    PID_Yaw.ErrNow= Exp_angle.Angle_Z - Now_angle.Angle_Z;

    PID_Pitch.Kp=19;//19,0.11
    PID_Pitch.Ti=0.12;//////tiaohao
    PID_Pitch.Td=0;

    PID_Yaw.Kp=19;
    PID_Yaw.Ti=0.18;
    PID_Yaw.Td=0;

    Increment_PID(&PID_Pitch);//执行绝对式PID算法
    Increment_PID(&PID_Yaw);//执行绝对式PID算法

    Motor_Value[1]=(s16)PID_Pitch.CtrOut;
    Motor_Value[0]=(s16)PID_Yaw.CtrOut;
    Motor_Drive(Motor_Value);
}

void Motor_Drive(s16 To_Motor[2])
{
    TIM4->CCR1-=To_Motor[0];//偏航yaw
    TIM4->CCR2-=To_Motor[1];//俯仰pitch

    if((TIM4->CCR1)>6000)//输出量限幅，抗积分饱和
        TIM4->CCR1=6000;
    else if((TIM4->CCR1)<3000)
        TIM4->CCR1=3000;

    if((TIM4->CCR2)>6000)//输出量限幅，抗积分饱和
        TIM4->CCR2=6000;
    else if((TIM4->CCR2)<3000)
        TIM4->CCR2=3000;
}

```

4) 中断服务程序

整个程序采用主程序加中断的结构，在中断中完成至关重要的角度信息解算和控制量输出的任务。其中角度解算是在串口 DMA 传输完成中断中执行，控制量输出是在定时器中断中执行。

串口数据 DMA 传输完成中断在一组完整的数据包传输完成时产生。MPU6050 发送数据的频率为 100Hz，因此 DMA 中断的频率大约为 100Hz。当中断发生时，先对数据的正确性进行校验，若数据正确，则直接执行角度解算函数。中断服务函数如下所示：

```
void DMA1_Channel6_IRQHandler() //dma1 中断
{
    if(DMA1->ISR&(1<<21)) //传输完成
    {
        DMA1->IFCR|=1<<21; //清除标志位
        if((0X51!=D_6050[0]) || (0X52!=D_6050[11]) || (0X53!=D_6050[22])) //校验数据正确性
        {
            DMA1_Channel6->CCR&=~(1<<0); //通道开启
            Dma1_C6_Close=1;
        }
        else
        {
            Update_Acce();
            Update_WAng();
            Update_Angle(); //只更新角度
            sign_usrt2=1; //标志位置位
        }
    }
}
```

定时器中断频率是 500Hz，即基本定时时间是 2ms。通过中断程序里定义计时变量累加实现更长时间的定时周期。由于舵机的时间常数一般在几十毫秒左右，因此控制周期设置为 4ms，即控制周期 250Hz；同时设定 LED 指示灯的闪烁频率为 2Hz。具体程序内容如下：

```
void TIM3_IRQHandler(void) //计数器中断
{
    if(TIM3->SR&0x0001)
    {
        Con_num++;
        if(Con_num == 2) //控制频率250Hz
        {
            Con_num = 0;
            PID_Angle(Angle_Now, Angle_Exp); //PID控制
        }

        Led_num++;
        if(Led_num>250) //2HZ
        {
            LED0=~LED0;
            Led_num=0;
            Tim_flag=1;
        }
    }
    TIM3->SR&=~(1<<0); //清除中断标志位
}
```

四、调试结果分析及结论

在硬件与软件完成的基础上，通过对 PID 参数的整定，反复调节使系统有良好的性能。由于不能准确建立系统的数学模型，所以只能采用试凑法来整定系统控制器的参数。一般先调整控制器 K_p 的参数，适当增加 K_p ，使系统的响应速度提高，偏差减小，但不可过大增加 K_p ，否则系统会产生严重超调甚至不稳定。当动态特性较好时，但偏差不在允许范围内，这时适当加入积分控制器，以消除静差。联合调试 K_p 和 K_i 两参数直到系统的性能指标达到要求。

1) 程控姿态控制系统

由于二自由度云台两个轴向的运动姿态是相互独立的，为了使调试更加简单，先固定一个轴向的角度不予控制，只调节另一个轴向的角度控制参数，然后用同样的方法调试第二个轴向的参数。最后将两个轴向的控制量同时加上，再根据系统性能微调参数。

调试过程中，通过串口将需要的角度信息发送到电脑，在电脑上使用串口助手软件接收显示数据。因为数据发送是在 DMA 中断里数据解算完成后即发送的，因此发送的频率略小于系统采样频率 100Hz。将接受的数据保存到 Text 文件后，导入到 MATLAB 软件中进行分析，并绘制出系统在不同输入值下的阶跃响应曲线。通过上述方法得到调试过程中的部分系统响应曲线如下：

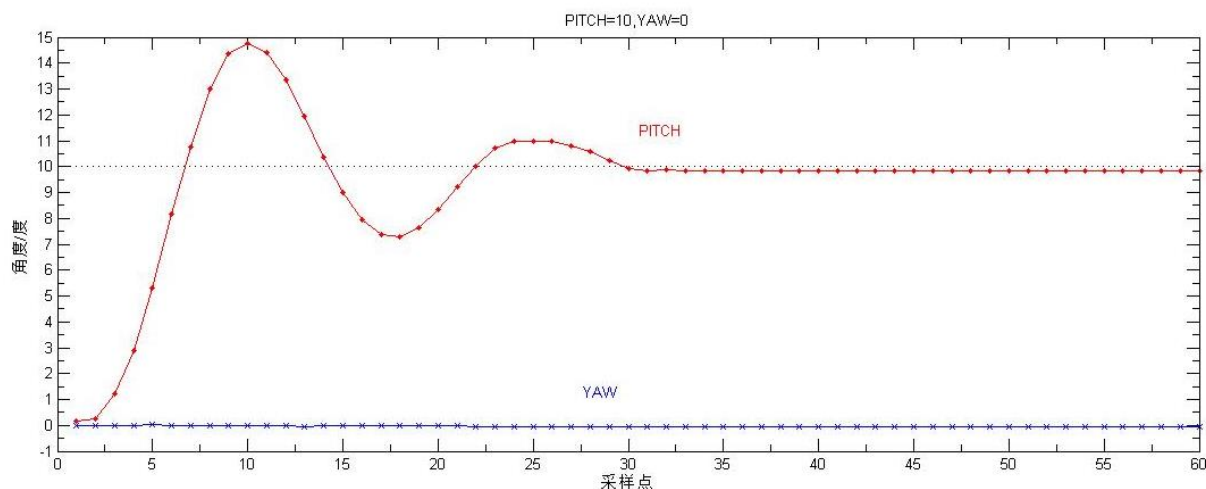


图 4.1.1 Pitch=10, Yaw=0 时的系统阶跃响应曲线

如图 4.1.1 为系统 Pitch=10, Yaw=0 时的阶跃响应曲线，可以看出系统 Pitch 角在第 30 个采样点后稳定在略小于 10 度的位置，即调节时间大约为 0.3s，稳态误差在 2% 左右，说明系统的稳态性能很好。同时看到曲线在第 10 个采样点，即 0.1s 时达到第一个峰值 14.5 度，随后经过一次波动达到稳定状态，可知系统的响应速度很快，但超调量有些大。对于姿态控制系统来说，要求系统的响应快，精确度高，超调量在可接受范围内即可，因此该参数对应下的系统是可用的，还可以进一步调节，达到更好的性能指标。

同样的方法调节 Yaw 的 PI 参数，分别得到不同参数对应下的 Pitch=10 度，Yaw=10 度和 Pitch=30 度，Yaw=30 度两组曲线如图 4.1.2 和图 4.1.3 所示。

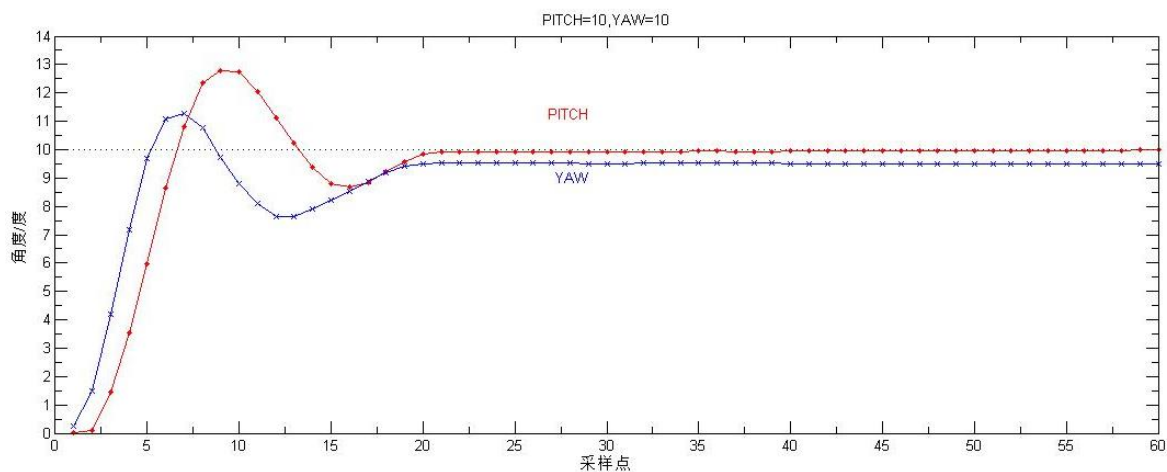


图 4.1.2 Pitch=10,Yaw=10 时的系统阶跃响应

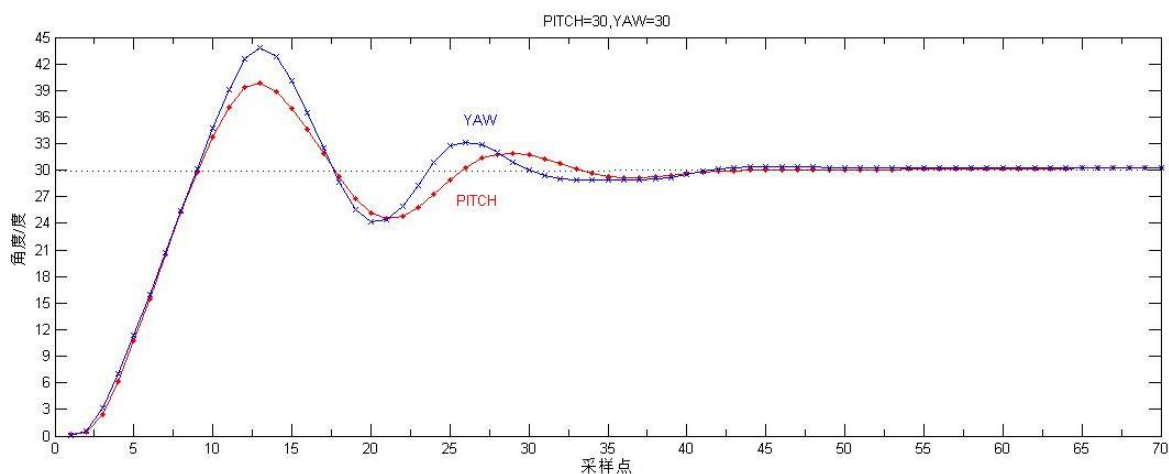


图 4.1.3 Pitch=30,Yaw=30 时的系统阶跃响应

对比两组数据可知，Yaw=10 组的超调小，响应速度快，但稳态误差大，Yaw=30 组的则正好相反，这也说明了动态性能和稳态性能之间的矛盾关系，此时可以加入 D 控制，增大系统阻尼的同时也能提高响应速度，同时达到更好的动态和稳态性能。Pitch 轴的两组数据在动态和稳态性能上基本相同，稳态误差均几乎为 0，动态性能上 Pitch=30 组有两个波峰，表现出更好的调节性能。

进一步随机改变两个轴向的角度设定值，按照一定时间改变设定值 (Pitch, Yaw) 的变化，变化情况为 (0, 0)；(0, 10)；(0, 20)；(0, 30)；(0, 0)；(10, 0)；(20, 0)；(30, 0)；(0, 0)；(10, 10)；(20, 20)；(30, 30)；(0, 0)；(10, -10)；(0, 0)；(-30, -30)；(0, 0)，得到载体角度的实时变化情况如下图 4.1.4 所示。

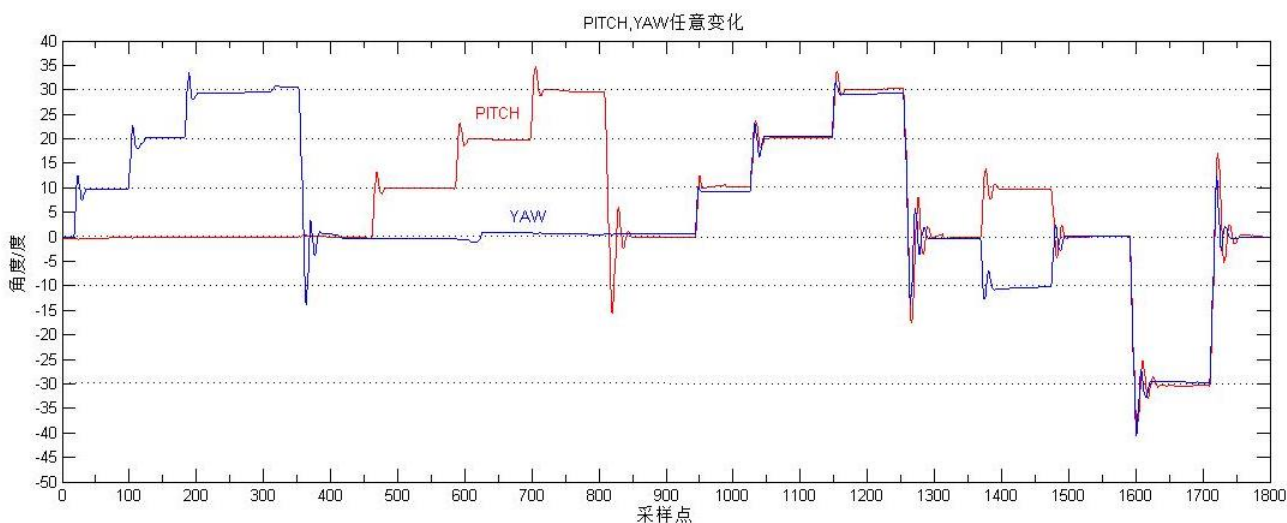


图 4.1.4 Pitch, Yaw 角度任意变化时的系统响应曲线

从上图可以看出，除了在输入角度变化量较大的情况下，会有较大超调外，其他情况下，系统的动态响应迅速和稳态误差很小，性能指标均符合设计要求。因此该程控双轴云台姿态角系统设计任务完成。

2) 恒值姿态控制系统

恒值姿态控制系统即是姿态角输入量始终为初始零值而有外界不断的随机强扰动作用下的角度控制系统，这要求系统的抗干扰性强。为了使所设计的系统有更强的实用性，将一个摄像头固定在两个舵机组成的双轴云台上，通过对系统施加干扰作用时，摄像头采集到的图像画面的抖动和画面的偏移情况，评定系统的性能好坏。

在程控姿态角控制系统设计的基础上，仍使用 PI 控制算法，得到在外界不同程度的干扰下的姿态角的变化情况，并绘制曲线分析。

首先，施加单个轴向的随机扰动信号，信号强度不定，分别得到 pitch 和 roll 轴向的角度响应曲线如图 4.2.1 所示。从曲线中可以看出在随机扰动的持续作用下，单个轴向的角度调节偏差基本能够控制在 $-5\sim+5$ 度左右，扰动强度越大，调节偏差也相应增大；当扰动消失，系统能够很快将角度调整到 0 度附近。对比两个轴向的调节效果，可以看出 Pitch 方向参数的调节效果好于 Roll 方向的参数调节效果，因此，Roll 方向的 PI 参数需要进一步调节。

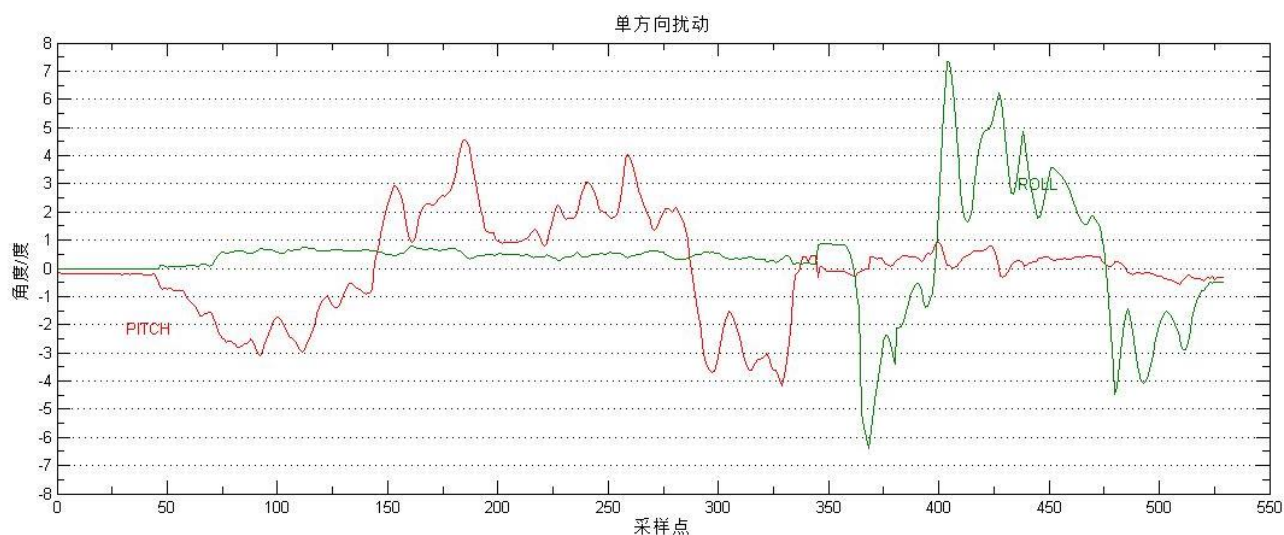


图 4.2.1 pitch 和 roll 轴向在随机扰动下的角度响应曲线

当施加任意方向的随机强扰动时，得到系统相应的响应，绘制响应曲线如图 4.2.2 所示。

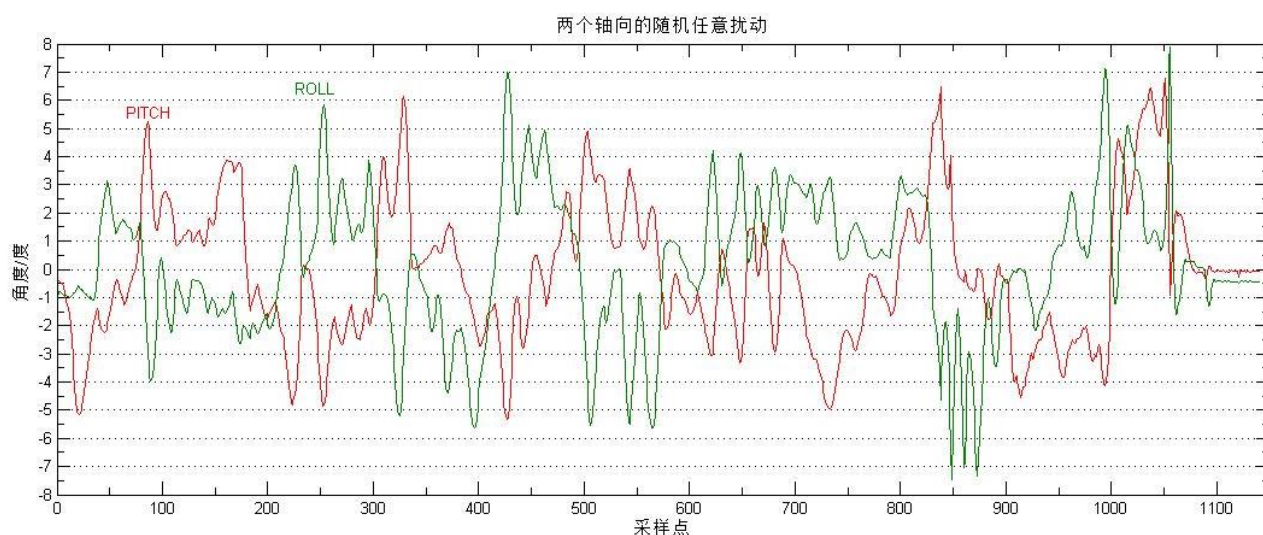


图 4.2.2 任意方向的随机强扰动下系统的响应曲线

分析曲线可知，系统对干扰的抵抗能力是有限的，扰动较强时，控制偏差超出 5 度的范围，其他情况下，控制精度在可接受范围内。虽然系统在扰动结束后能够很快恢复到初始姿态角，但在持续的强扰动中的控制效果不是很理想，需要加入 D 控制，或使用串级控制等复杂的控制规律进行控制。

当放置一个目标物体在摄像头的画面中央，系统受到强烈扰动时，观察到目标物体在画面中并没有明显的抖动或者移动，表明该系统基本可用，但若想要有更好的性能，还需要进一步调试和改进。

3) 作品实物图

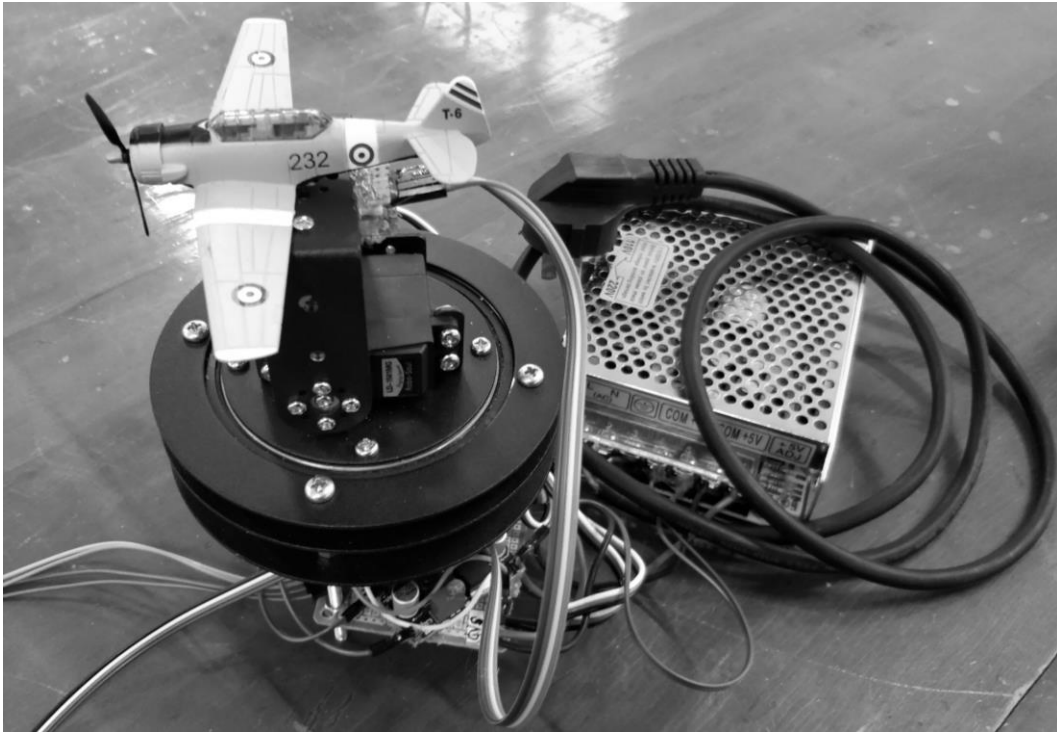


图 4.3.1 程控姿态控制系统实物图

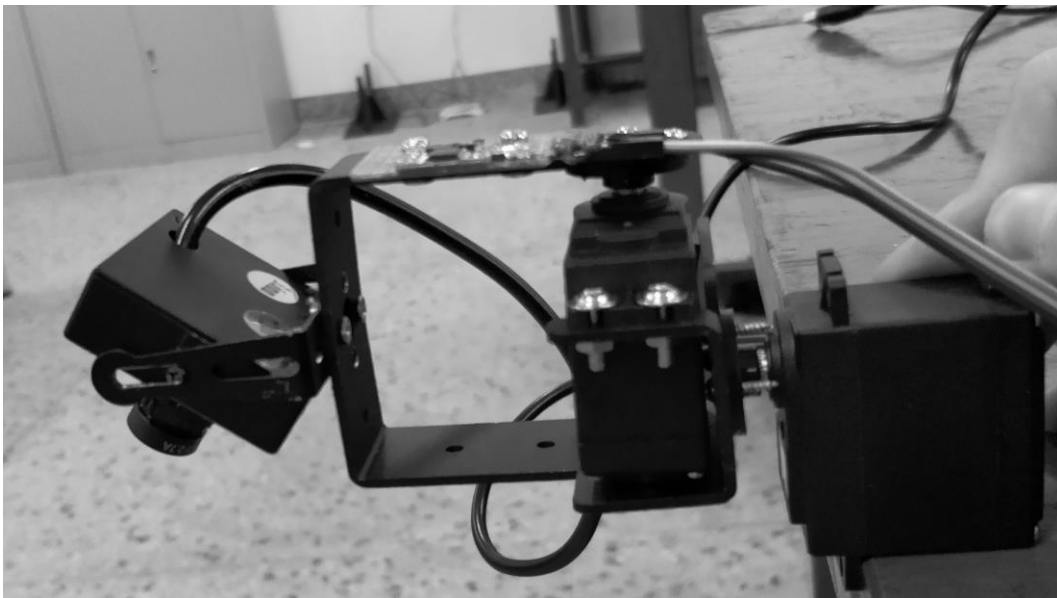


图 4.3.2 恒值姿态控制系统实物图

五、指导教师评语及学生成绩

指导教师评语：		
年 月 日		
成绩		指导教师(签字)：
