ISSN: 2661-3719(Print); 2661-3727(Online)

微型飞行器对地侦察与打击

张　铎
太原理工大学　信息与计算机学院　山西太原　030000

摘　要：本文设计了一种基于视觉传感器和IMU的飞行控制器，开发了一种飞行器自主识别与追踪的系统。着重阐述了对固定翼飞行器、飞行器的发射装置和飞行器侦察与打击功能的研究。其中侦察功能通过视觉和IMU传感器对目标物进行识别，获得位置信息，根据位置信息来控制飞行器飞行姿态，从而实现打击的目的。

关键词：飞行器；固定翼；视觉；IMU

1.微型飞行器对地侦察与打击设计简要分析

本设计主要的研究内容包括：

（1）飞行器对地侦察与打击的总体设计，围绕智能微型无人飞行器的发展，从而确定了飞行器机械结构、硬件设计和软件设计方案。

（2）飞行器机械结构方面的设计要求包括：飞行器翼面设计、飞行器转向设计和飞行器发射架设计。

（3）飞行器硬件方面的设计要求包括：K210芯片电路设计和MPU-6050芯片电路设计。

（4）飞行器软件方面的设计要求包括：飞行器发射架软件调试、K210目标物识别算法设计、K210目标物阈值调试、K210目标物坐标提取算法设计和飞行器姿态解算算法设计。

2.微型飞行器对地侦察与打击的设计概述和工作原理

2.1飞行器对地侦察与打击设计概述

该课题设计的目标是自主研发一款微型飞行器，在其基础上实现侦察与打击的功能。其中侦察功能要求该微型飞行器具有识别目标物（如色块、数字和具有特征的物体等）并处理其位置信息的能力。打击功能则要求该微型飞行器对所获得的位置信息和自身搭载的IMU中数据信息进行处理，通过对舵机打角的控制来调整飞行器翼面，从而对飞行器飞行姿态进行矫正，实现精准打击的目的。

2.2微型飞行器对地侦察与打击工作原理

微型飞行器安装在发射架上，通过发射架的自动装填结构将微型飞行器送入发射轨道，利用发射架的动能装置为飞行器提供动力使其发射出去。微型飞行器在空中主要先经过升力大于重力的上升过程，后进行升力小于重力的滑翔过程。在上升过程中，微型飞行器主要通过IMU的姿态解算进行飞行姿态的调节与校准，防止微型飞行器在上升过程中出现翻转现象。在滑翔过程中微型飞行器通过其搭载的摄像头识别捕捉目标物得到其位置信息，完成侦察功能。根据对所获得的位置信息和自身搭载的IMU中数据信息进行处理，计算出飞行器飞行轨迹的偏移量，通过舵机打角的方式来调整飞行器翼面，从而对飞行器飞行姿态进行矫正，从而使得飞行器落点在目标物误差允许的范围内，完成打击功能。

3.微型飞行器的机械、硬件和软件技术概述

3.1微型飞行器相关机械技术

（1）微型飞行器机翼设计。微型飞行器需要具有一定载荷的能力，以便于携带传感器。而打击功能则要求微型飞行器具有高速飞行和机动性高的特性。结合课题需求和微型飞行器制作的难易程度，决定设计有尾常规式固定翼微型飞行器。

（2）微型飞行器发射架设计。根据课题要求，发射装置决定采用弹射式发射，发射姿态为倾斜发射。

（3）发射架装填设计。为了使本课题中微型飞行器的发射装置更加智能化，消除人为装填带来的误差影响，为微型飞行器的发射装置设计自动装填的功能。

3.2微型飞行器相关硬件技术

（1）电源选型。由于微型飞行器受到尺寸和重量的限制，在保证续航的同时应选用重量较轻的供能装置。经过对重量、容量、效率和寿命等的综合考量后，决定使用聚合物锂电池作为电源为其供能。

（2）K210卷积神经网络开发板设计。传感器作为微型飞行器的“感官”，是微型飞行器智能化过程中必不可少的要素。根据无人飞行器领域内科技公司研究的先例，采用摄像头作为微型飞行器的传感器是比较成熟的应用。而嘉楠科技自主研发的K210机器视觉（卷积神经网络加速处理器KPU）系统级芯片（SOC）可用来处理摄像头采集到的目标物信息。

作者简介：张铎，男（1998.10-），汉族，山西临汾人，太原理工大学信息与计算机学院本科生在读，研究方向：电子信息工程。

（3）IMU-MPU6050电路设计。微型飞行器离开发射装置到打击到目标物这个过程中，需要不断地对自身飞行姿态进行调整，以避免发生翻滚，使得飞行器平稳地实现其功能。MPU-6050是6轴的IMU传感器，可以用来确定微型飞行器的倾角方向和大小，从而根据所得值对微型飞行器的姿态进行调整。

3.3微型飞行器相关软件技术

（1）微型飞行器发射装置软件调试。通过设计程序控制单片机发送指令，使微型飞行器发射装置上电机带动摩擦轮转动，实现微型飞行器自动装填和自动发射的功能。

（2）微型飞行器侦察功能调试。微型飞行器侦察功能是使用K210芯片的机器视觉功能实现的。通过Micropython移植到K210芯片（在K210上运行Micropython的解析器）的功能，我们可以直接运用Micropython编程来控制K210芯片。

（3）微型飞行器打击功能调试。利用IMU的姿态解算控制舵机打角实现微型飞行器的稳定飞行，使得其在飞行过程中可以有效识别到目标物。再通过K210的视觉功能，计算出目标物位置坐标，计算偏差，使舵机打角从而实现对目标物的打击。

4.微型飞行器课题总体设计

4.1微型飞行器机械设计

对于飞行器对地侦察与打击课题的任务要求来说，它的侦察功能需要微型飞行器具有载荷的能力，以便于携带传感器。而打击功能则要求微型飞行器具有高速飞行、机动性高和能量损耗小的特性。结合课题需求和微型飞行器制作的难易程度，决定设计有尾常规式固定翼微型飞行器来完成课题。结合实际情况，经过多种材料的一系列对比，决定分别采用pp（聚丙烯）板和PLA（聚乳酸）作为机翼和机身的设计材料。固定翼飞行器设计斜视图与实物图分别如图1、图2所示。

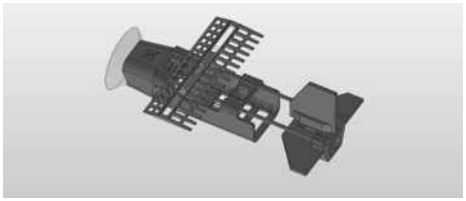

图1　固定翼飞行器设计总览斜视图

图2　固定翼飞行器设计实物图

4.2微型飞行器发射装置机械设计

本课题发射装置所采用的弹射发射方式有三种不同的实现方案，分别使用橡皮筋、气动装置和电机作为动力源。在试验过程中发现通过电机带动摩擦轮对进入轨道的微型飞行器进行挤压弹射出去，这种发射方式可通过控制电机转速来控制微型飞行器发射的初始动力。发射过程中产生的误差较小。为了使本课题中微型飞行器的发射装置更加智能化，消除人为装填飞行器带来的误差影响，决定为微型飞行器的发射装置设计自动装填的功能。通过电机带动圆形齿轮组，从而将齿条上安装的微型飞行器送入发射结构的摩擦轮组。发射装置的发射结构设计图、自动装填功能设计图与发射装置整体侧视图分别如图3、4、5所示。

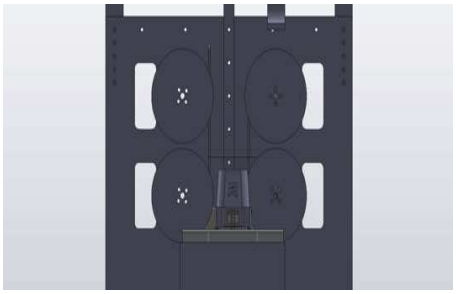

图3　发射结构设计图

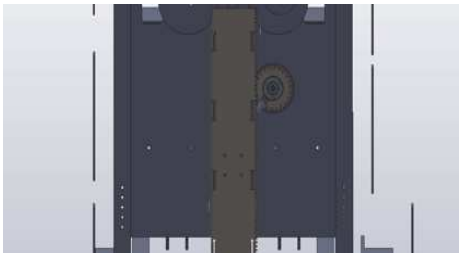

图4　自动装填功能设计图

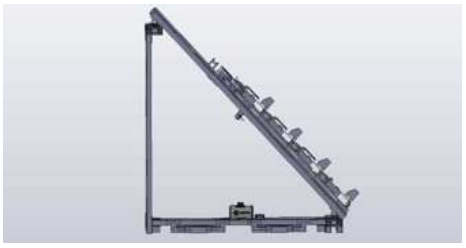

图5　发射装置整体侧视图

4.3微型飞行器硬件设计

Kendryte K210是集成机器视觉与机器听觉能力的系统级芯片（SoC），该芯片力求零门槛开发，可在最短时效部署与用户的产品中，赋予产品人工智能。

根据SIPEED公司的开源资料，我对Maix Bit开发板进行了优化设计。删减了麦克风阵列、LCD显示和MicroSD卡槽等冗余功能，集成了MPU-6050模块。使用Altium Designer软件设计的电路原理图与3D图如下：

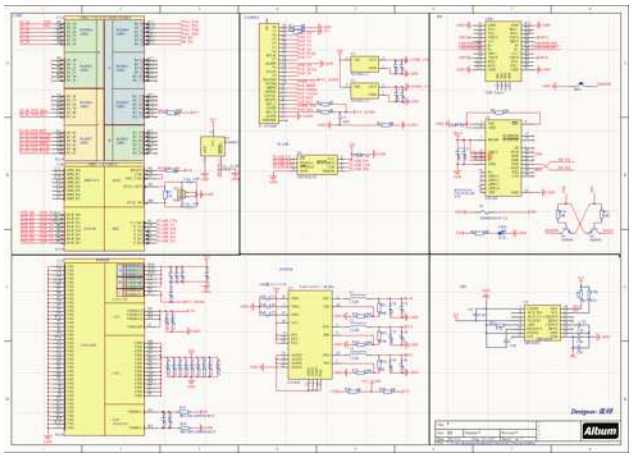

图6　硬件电路设计原理图

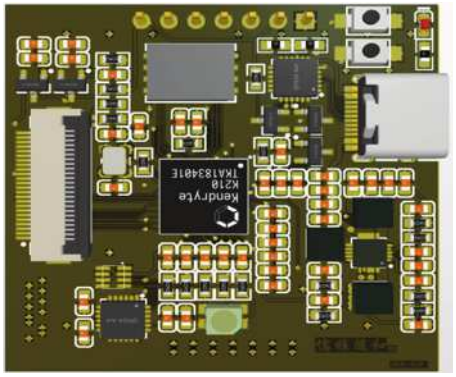

图7　硬件电路设计PCB3D图

4.4 微型飞行器软件设计

发射部分

微型飞行器发射装置采用的是电机带动摩擦轮的方案，包含两对摩擦轮组，共四个电机，采用增量式PID对电机进行程序设计。

以下是对电机控制的部分程序：

```
float     Ki[2] = {35,0};
float     Kp= 35, Ki= 0.1, Kd= 0;
int16_t   p[5]={0,0,0,0,0},i[5]={0,0,0,0,0},d[5]={0,0,0,0,0};
int16_t   P_diff[5] = {0,0,0,0,0},P_diff_b[5] = {0,0,0,0,0};

int16_t PID(int16_t Expect,int16_t Current,int ID)
{

  P_diff[ID-0x201] = (Expect - Current);

  p[ID-0x201] = P_diff[ID-0x201] * Kp;
  i[ID-0x201] += P_diff[ID-0x201] * Ki;
  d[ID-0x201] = (P_diff[ID-0x201]-P_diff_b[ID-0x201]) * Kd;

//i值抗饱和处理
//  if(Abs(diff) < 500)
//    Ki = 1;
  P_diff_b[ID-0x201] = P_diff[ID-0x201];//差值备份


  return ((p[ID-0x201]+i[ID-0x201]+d[ID-0x201])>32768?32768:(p[ID-0x201]+i[ID-0x201]+d[ID-0x201]));
}
```

图8　电机控制程序（部分）

```
switch(rx_header.StdId)
  {
    case 0x201:
      Speed_Current[0] = (int16_t)(rx_data[2]<<8|rx_data[3]);//速度解算
      //Angle_Current[0] =(int16_t)(rx_data[0]<<8|rx_data[1]);//角度解算
      break;
    case 0x202:
      Speed_Current[1] = (int16_t)(rx_data[2]<<8|rx_data[3]);//速度解算
      //Angle_Current[1] = (int16_t)(rx_data[0]<<8|rx_data[1]);//角度解算
      break;
    case 0x203:
      Speed_Current[2] = (int16_t)(rx_data[2]<<8|rx_data[3]);//速度解算
      //Angle_Current[2] = (int16_t)(rx_data[0]<<8|rx_data[1]);//角度解算
      break;
    case 0x204:
      Speed_Current[3] = (int16_t)(rx_data[2]<<8|rx_data[3]);//速度解算
      //Angle_Current[3] = (int16_t)(rx_data[0]<<8|rx_data[1]);//角度解算
      break;
    case 0x205:
      Speed_Current[4] = (int16_t)(rx_data[2]<<8|rx_data[3]);//速度解算
      Angle_Current[4] = (int16_t)(rx_data[0]<<8|rx_data[1]);//角度解算
```

图9　电机速度与角度解算程序

如图9对发射装置发射部分电机进行了速度解算和角度解算，使它们转速一致，确保微型飞行器在加速过程中不会有过多能能量损耗。

```
//发射
if(dart_client_cmd.dart_attack_target == 0)
  launch_deinit = 0;
 if(dart_client_cmd.dart_attack_target == 1
  && dart_client_cmd.dart_launch_opening_status == 0
 && launch_deinit == 0
 && game_status.game_progress == 4)
 {
  launch = 1;
  launch_deinit++;
  position = Angle_Count_Current[4];
 }
 if(launch)
 {
  //设定期望值
  Speed_Expect[4] = -8000;//推动
 }
 else
 {
  //设定期望值
  Speed_Expect[4] = 0;//推动
 }
 //打出一发距离后停止
 if(position - Angle_Count_Current[4] > 350000) launch = 0;
 //限位
 if(Angle_Count_Current[4] < -1200000)
  Speed_Expect[4] = 0;//推动
 timed = dart_client_cmd.operate_launch_cmd_time;
 CAN1_Send();
```

图10　电机速度与角度解算程序

如图10在接收到发射指令后，发射装置自动装填部分电机开始转动，带动齿条，将微型飞行器送入发射部分。在发射完一架微型飞行器后，开始判断是否满足发射条件，若满足条件则继续发射。反之，则停止发射，等待发射指令。

侦察功能部分

为了使微型飞行器更加智能化，本课题对微型飞行器提出了侦察的要求。要使微型飞行器在飞行过程中，可以通过摄像头图识别到具有鲜明特征的目标，并对其进行追踪。在侦察过程中，对其位置坐标进行计算，所得数据信息用来实现微型飞行器的打击功能。选取具有明显特征的日光灯作为目标物，对其进行阈值调试。

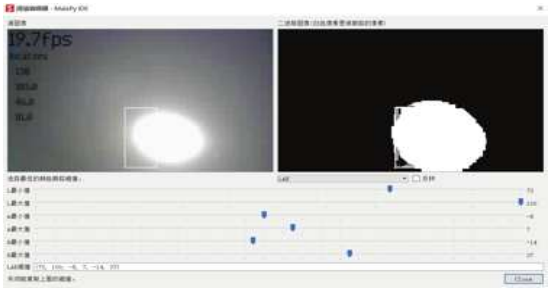

图11　阈值编辑器

将阈值作为目标物的特征值，对目标物进行识别。将识别到的目标物进行框选捕捉，并在框选中心画上十字，以显示寻找到的目标。找到目标物后对目标物位置坐标进行运算，显示在屏幕上。

```
while(True):
        clock.tick()
        img = sensor.snapshot()
        fps =clock.fps()
        blobs = img.find_blobs([white_threshold_01],
                               roi = roi2,
                               area_threshold=1000)#在图像中寻找目标颜色区域
        last_blobs = blobs
    else:
        blobs = img.find_blobs([white_threshold_01],
                               area_threshold=1000)#在未框选的区域寻找目标颜色
        last_blobs = blobs
    if last_blobs:
        #如果找到了目标颜色
        #  print(blobs)
        for b in last_blobs:#迭代找到的目标颜色区域
            img.draw_rectangle(b[0:4])#将找到的目标物框选
            img.draw_cross(b[5], b[6])#在目标物框选中心画十字
```

图12　目标物识别算法

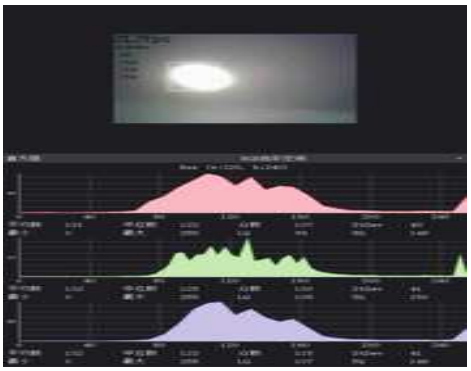

图13　提取目标物坐标

打击功能部分

打击功能的实现，首先要对飞行器飞行姿态进行自稳调节，使得飞行器能够在预定轨迹范围内飞行，以免偏差过大，导致摄像头无法捕捉到目标物特征。在识别到目标物后，要根据处理后的目标物位置信息再次调整飞行轨迹，向目标物方向飞去，最终对目标物进行打击。在上电自校准完成后，使用由威廉·曼哈顿提出的四元数法进行陀螺仪解算，得到pitch（俯仰角）、yaw（偏航角）、roll（翻滚角）的偏差值。在飞行器刚离开发射装置上升的过程中，摄像头无法捕捉到地面的目标物，这个过程需要由IMU进行姿态解算，保持飞行器稳定的飞行状态。当飞行器出现俯角并且进入摄像头识别范围时，就需要用到侦察功能的到的目标物位置信息。这时，只使用IMU姿态解算并不能使飞行器打击到目标物，需进行IMU与机器视觉的姿态融合解算。在程序中加入判断，计算出融合后的pitch轴、yaw轴、roll轴偏差，进而控制舵机打角，实现打击功能。IMU姿态解算算法程序设计图与姿态融合解算算法设计图如图14、15所示。

图14　IMU姿态解算算法

```
Get_EulerAngle(mpuData.q,&rocketData.yaw,&rocketData.pitch,&rocketData.roll);
//rocketData.state=2;
switch (rocketData.state)
{
  case 0:
  {
    if(mpuData.ay<-5&&rocketData.flyTick==0)
    {
      rocketData.flyTick=1;
      rocketData.adjTick=0;
      yawPPid.aimData = mpuData.yaw-20;
      rollPPid.aimData = rocketData.roll;
      rocketData.aimYaw =yawPPid.aimData;

    }
    if(rocketData.flyTick>0)
    {
      if(rocketData.flyTick++==100)
      {

        rocketData.state=2;          //Enter 姿态调整
        rocketData.flyTick=0;

      }
    }
```

图15　姿态融合解算算法设计

但是，这样的姿态融合算法仍然不能使飞行器实现精准打击的目的。飞行器在空中飞行时的姿态不仅受到舵机打角的影响，空气阻力、发射装置摩擦轮两边的差速、摄像头传输图像的延时和舵机响应的滞后性等因素的存在，使飞行器的姿态调节算法效果的不确定性显著增大。为了尽可能使飞行器对目标物打击的准确度提高，需要不断进行参数调试，在这个过程中，对算法的优化尤为重要。

参考文献：

[1]夏鑫宇，肖君如.智能无人飞行器的发展现状及趋势[J].石河子科技，2021（01）：17-18.

[2]昂海松.“微型飞行器的现状、难题和发展趋势”.2014（第五届）中国无人机大会论文集.Ed..航空工业出版社，2014，664-669.

[3]姜倩，张泽卿，闫晓坤.微型飞行器发展现状与关键技术[J].企业技术开发，2015，34（08）：1-2.

[4]潘宇，唐万洪，刘斌，陈俊成.基于OpenMV开发的数字图像处理技术[J].电子技术与软件工程，2021（09）：130-131.

[5]孙雪丽，张亚周，金堃.OpenMV视觉模块的滚球目标识别与追踪系统研究[J].单片机与嵌入式系统应用，2021，21（02）：8-10.