

送货无人机

摘 要

本方案旨在完成无人机对目标位置的送货和穿越呼啦圈的功能。系统底部搭载 T265 双目摄像头确定无人机的实时位置,利用单片机和飞行器 IMU 所生成的 PWM 数据,结合 PID 算法实现无人机运动的控制。

对于两种不同的目标位置坐标输入方式,我们分别使用矩阵键盘和 Intel 二代神经计算棒识别进行输入。对于目标位置的确定先由 T265 确定大致位置,而后利用视觉进行校准。待抵达目标位置后,定点降低巡航高度进行送货。在送货过程中,通过 PWM 驱动舵机完成吊舱(砝码)的升降。完成一次送货后,恢复原始巡航高度,寻找下一个目标位置。待全部位置均完成送货后,返回起飞点。对于穿越呼啦圈的任务,则由无人机的侧面摄像头搭配 PID 进行控制。

关键词: T265 双目摄像头 STM32F4 单片机 集成 IMU PID 控制
视觉神经网络

一、系统方案

1.1 技术路线与系统结构

实现该系统主要有两方面，一是飞行控制模块，二是视觉识别模块。本系统采用的飞行控制模块主要由凌霄 IMU 和 STM32F407 实现，视觉识别模块主要依靠华硕 Tinkerboard 2s 和双目摄像头 T265 实现。

无人机底部搭载的双目立体视觉系统与光流传感器通过 STM32F407 串口驱动，配合 IMU 中陀螺仪，加速度计等多种传感器进行数据采集，同时配合 PID 算法，可以得到飞行器的高度，加速度，速度等姿态信息以及飞机的补偿方向与角度。IMU 将这些数据转换为 PWM 数据发送至 STM32F407 中，而 STM32F407 则通过获得的 PWM 信息进行转换，并将得到的 PWM 波输出给电调实现飞行器的运动控制，从而完成定高飞行、定点飞行、悬停、巡点降落等要求。

T265 摄像头通过其内含的宽视场立体鱼镜头捕获空间内容，使用视觉惯性里程计计算其在 3D 空间中的方向和位置及六自由度。T265 则通过实时捕获图像信息，确定飞机的实时位置。同时，华硕 Tinkerboard 2s 开发板则通过 T265 获得的位置信息与 STM32F407 得到的信息数据进行不同线程之间的调度与调整从而完成飞行器的稳定飞行与任务执行。

1.2 系统方案的论证、比较和选择

1.2.1 飞行控制模块

方案一：采用 Pcdino 板为控制核心。其内存大而且板子体积较小，重量较轻，符合飞行控制的实时性并有效降低飞行器自重。但 Pcdino 板对电源的要求较高，而且 I/O 口较少，不能很好的扩展飞行器其他功能。

方案二：采用 STM32F407：STM32F4 是基于 ARM Cortex-M4F 内核的微型处理器，主要瞄准低成本预算的应用，但如果仅仅使用 STM32 完成包括图像处理、飞行器控制、飞行器定高、姿态检验等功能相对来说十分困难，但搭配无人机 IMU 可较好完成对飞行器的控制。

综合对比两个方案，我们选择方案二的芯片，其存在的弊端对整个装置影响较小，更能满足我们的需求。

1.2.2 视觉模块

方案一：OpenMV：其具有低功耗，低成本的特点，方便实现机器视觉相关的应用。其主频为 480HZ，处理器采用 STM32H473VI，数据传输接口有 I2C、

UART、SPI。

方案二：Intel-T265 实感追踪摄像头：采用直接在设备上运行的高度优化的专有 V-SLAM 算法，运行功率仅为 1.5W，具有低功耗的特点。同时，经过性能测试和验证，在预期使用条件下，闭环偏移小于 1%。除此之外，该设备尺寸为 $108 \times 25 \times 13$ 毫米，重量仅为 55 克，因此不会过度增加无人机的负担。

方案三：Intel 二代神经计算棒：具有多功能硬件处理能力，缩短了对神经网络进行原型设计或调整的时间。采用 16 核外加神经计算引擎、专用的深度神经网络加速器。利用深度神经网络推理实现性能提升高达 8 倍，加快了深度神经网络的应用速度。

综合对比三个方案，我们最终选择方案二和三并用，这样的方案功耗小，处理能力强，搭配 Tinkerboard 2s 计算板使用，作为最终的视觉处理模块。

二、设计与计算

2.1 控制方法描述及参数计算

2.1.1 飞行姿态控制

飞行姿态的量化可以采用四元数的方式表示，四元数的基本表达形式包含 1 个实部和 3 个虚部，即 $q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k$ 。四元数可以表示为一种旋转，例如某一向量绕单位向量 $\mathbf{n} = [n_x, n_y, n_z]$ 表示的转轴进行逆时针旋转 θ 角度，可以用四元数表示为 $p = \left(\cos \frac{\theta}{2}, \mathbf{n} \sin \frac{\theta}{2} \right)$ 。

对于 i、j 和 k 本身的几何意义可以理解作为一种旋转，其中 i 旋转代表 Z 轴与 Y 轴相交平面中 Z 轴正向向 Y 轴正向的旋转，j 旋转代表 X 轴与 Z 轴相交平面中 X 轴正向向 Z 轴正向的旋转，k 旋转代表 Y 轴与 X 轴相交平面中 Y 轴正向向 X 轴正向的旋转，-i、-j、-k 分别代表 i、j、k 旋转的反向旋转。

我们所采用的坐标系变换，是按 ZYX 顺序从地理坐标系转到机体坐标系的，其旋转矩阵可以表示为式，从机体坐标系到地理坐标系即为式的转置。

$$C_w^a = \begin{bmatrix} \cos \theta \cos \psi & \sin \psi \cos \theta & -\sin \theta \\ \sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi & \sin \varphi \sin \theta \sin \psi + \cos \varphi \cos \psi & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta \cos \psi + \sin \varphi \sin \psi & \cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \cos \theta \end{bmatrix} \quad (1)$$

其中， φ, θ, ψ 分别表示绕 X、Y、Z 轴旋转的角度，其可以按式进行计算。

飞行器姿态的改变，可以对应到旋转矩阵的改变，进一步对应到四元数的改变。实时姿态计算，实际上也就是实时更新四元数。我们可以构建四元数关于时

间的微分方程，来研究四元数关于时间的变化规律，求解该微分方程，即可解出四元数。设有微分方程 $dy/dx = f(x,y)$ ，求解 y 的迭代公式如式所示。

$$\begin{aligned}\varphi &= \arctan \frac{2(q_2 q_3 + q_0 q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \\ \theta &= -\arcsin 2(q_1 q_3 - q_0 q_2) \\ \psi &= \arctan \frac{2(q_1 q_2 + q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\end{aligned}\quad (2)$$

$$y(\lambda + \Delta\lambda) = y(\lambda) + \Delta\lambda \cdot f(x(\lambda), y(\lambda)) \quad (3)$$

对应到四元数微分方程为：

$$Q(t - \Delta t) = Q(t) + \Delta t \cdot \Phi(t) \cdot Q(t) \quad (4)$$

整理得式，其中 ω 为三轴角速度，可以由陀螺仪直接测得。

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t+\Delta t} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_t + \frac{1}{2} \cdot \Delta t \cdot \begin{bmatrix} -w_x \cdot q_1 - w_y \cdot q_2 - w_z \cdot q_3 \\ w_x \cdot q_0 - w_y \cdot q_3 + w_z \cdot q_2 \\ w_x \cdot q_3 + w_y \cdot q_0 - w_z \cdot q_1 \\ -w_x \cdot q_2 + w_y \cdot q_1 + w_z \cdot q_0 \end{bmatrix} \quad (5)$$

2.1.2 PID 控制

相比传统方法在单纯的比例控制的基础上增加了积分项与微分项的修正控制，我们还引入了前馈修正。积分算法通过对前面若干次误差分量的积分项模拟实际系统中存在的惯性问题，从而逐渐消除了比例算法带来的稳态误差；微分控制则为了反映偏差信号的趋势，在偏差信号过大之前引入一个早期修正信号，减小了控制过程中可能发生的震荡现象。除此之外，前馈调节属于开环控制，可以以目标值为核心基准进行调控，进一步提高系统响应速度，减小反馈控制压力。PID 控制的数学表达式见式。

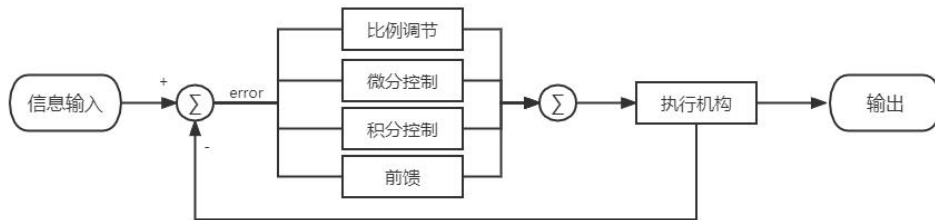


图 1: PID 控制流程图

$$u = k_p \cdot e(t) + k_i \cdot \int_0^t e(\tau) d\tau + k_d \cdot \frac{de(t)}{dt} + k_f \cdot obj \quad (6)$$

其中， $e(t)$ 为误差函数，其值等于目标值 obj 与当前值之差， k_p, k_i, k_d 和 k_f 分别表示比例系数、积分项系数、微分项系数和前馈系数。

三、电路与程序设计

3.1 系统组成与原理框图

首先，针对要求 1 和要求 2，分别通过矩阵键盘和视觉识别确定目标位置的位置信息。矩阵键盘向单片机发送信息而后传到飞控板上，后续通过飞控与视觉结合到达目标点，完成送货任务。

控制系统硬件主要包含 MCU、IMU 及多个传感器。惯性测量单元 IMU 内含陀螺仪、加速度计等多个传感器可以较为轻松完成控制算法、姿态解算、传感器融合等多项任务。而此时只需要一个辅助单片机进行数据传输，将控制数据和由视觉模块根据识别结果给出的指令通过串口发送给 IMU，然后接收 IMU 输出的 PWM 数据转化为 PWM 波输出给电调，完成“语音播报”、“升降吊舱”等一系列送货任务。

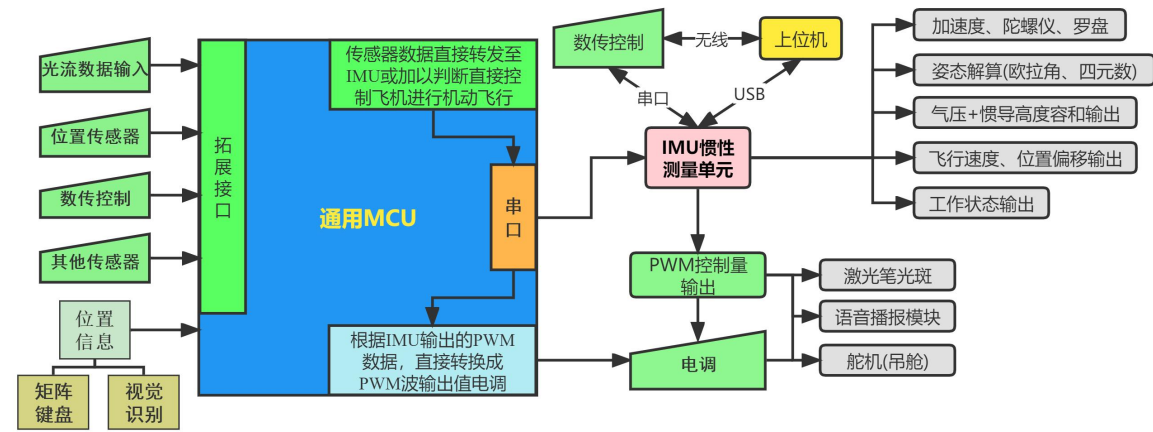


图 2：系统原理框图

3.2 各部分电路图

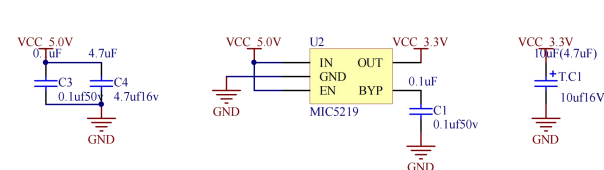


图 3：稳压电路图

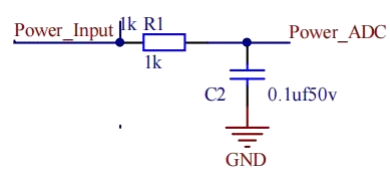


图 4：电压测量电路图

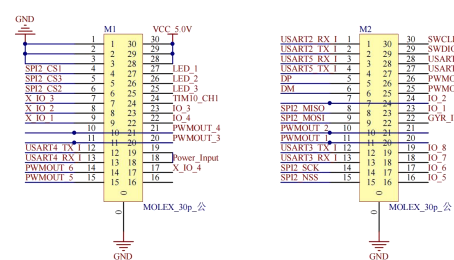


图 5：MCU CORE 接口电路图

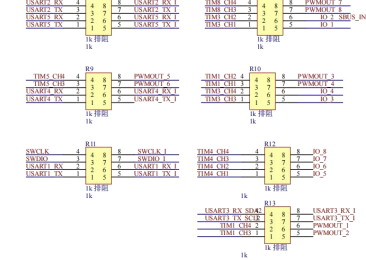


图 6：排阻电路图

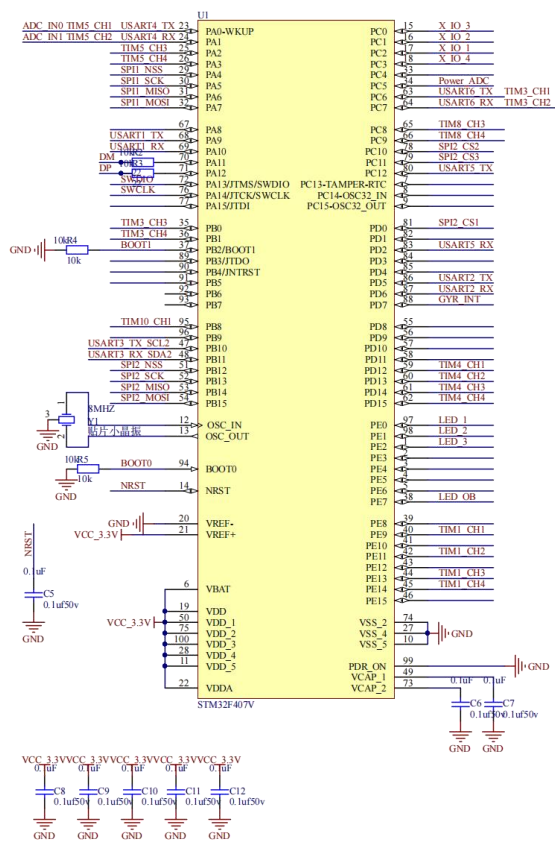


图 7: STM32 电路图

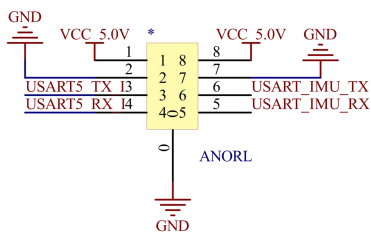


图 13: 数传封装电路图

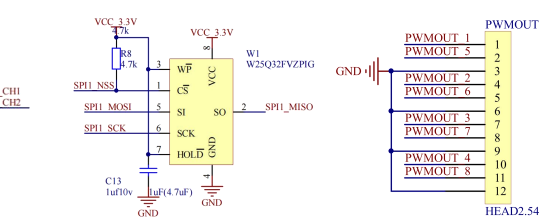


图 8: SPI Flash 电路

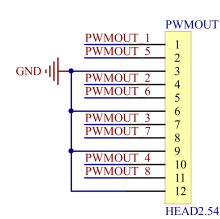


图 9: 电调电路

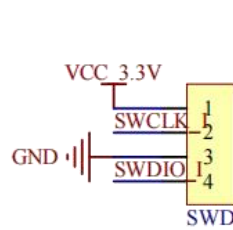


图 10: SWD 接口

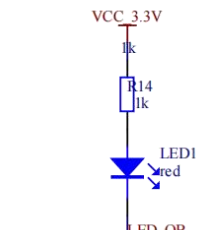


图 11: 可控 LED

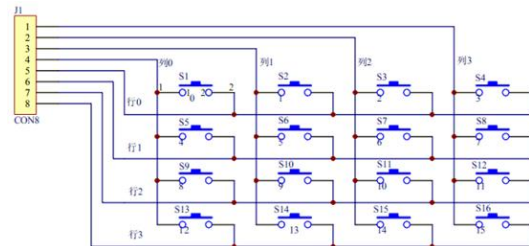


图 12: 矩阵键盘扫描电路

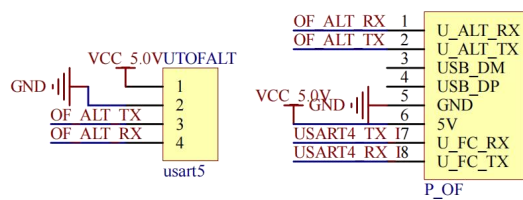


图 14: 光流封装电路图

3.3 系统软件设计与流程图

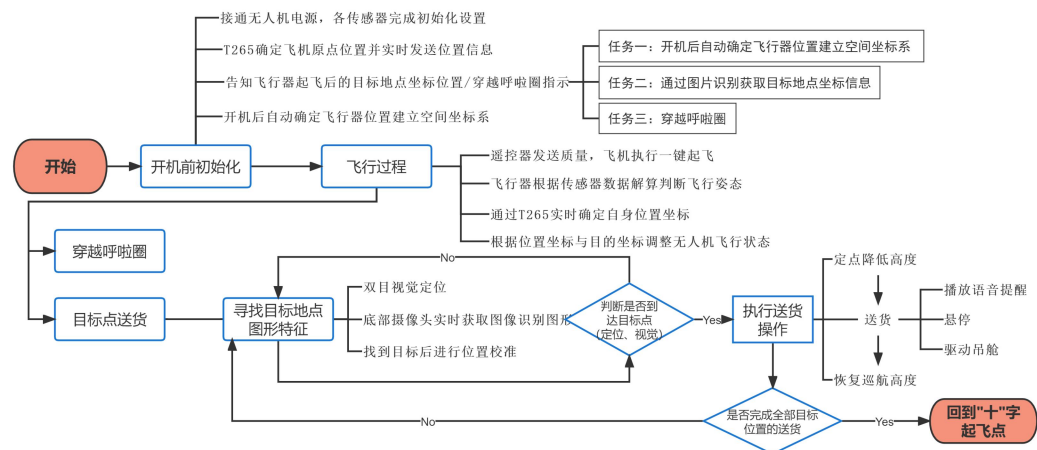


图 15: 程序设计流程图

⑥送货完成即恢复巡航高度；

⑦作业完成后飞行到起降点稳定准确降落，无人机几何中心点与起降点中心距离偏差不大于 $\pm 10\text{cm}$ 。

⑧对于矩阵键盘键入坐标的信息输入方法（要求 1），送货过程必须要在 180 秒内完成，用时越少越好。对于视觉识别目标位置特征的信息输入方法（要求 2），送货过程用时越少越好，需在 270 秒内完成。

测试现场避免阳光直射，但不排除顶部照明灯及窗外环境光照射。

4.2 测试结果

表 1：“无人机送货”测试结果

目标地点 坐标输入方式	第一个 目标位置坐标	第二个 目标位置坐标	第一个 送货坐标	第二个 送货坐标
键盘输入	(125,50)(7 号)	(275,200)(3 号)	(122.4,47.6)	(272.1,196.3)
键盘输入	(50,125)(9 号)	(275,50)(6 号)	(48.1,123.2)	(272.4,47.1)
特征识别：●	(125,50)(7 号)	(125,200)(8 号)	(123,48.8)	(123.4,197.1)

表 2：“穿越呼啦圈”测试结果

选用的 A 点坐标	朝向	是否成功穿越
(200, 200)	平行 x 轴	是
(125, 125)	平行 y 轴	是
(200, 50)	平行 x 轴	是

4.3 测试结果分析

对于“无人机送货”任务，两种目标位置信息获取方式下，无人机均可以以较高的精度完成定位而后进行送货，这是因为仅由双目摄像头就可以较为精准地得到无人机当前坐标信息，再辅以视觉进行校准，定点会更准确。

对于“穿越呼啦圈”任务，先令无人机沿靠近边界处与呼啦圈圆心等高处飞行，在飞行过程中使用无人机侧面的摄像头拍摄画面，倘若拍摄到红色圆位于正中心，则向前飞行穿越呼啦圈。