

# 一种基于无人机与无人车协同的精准投递智能物流工具的设计

董豪泽 张有成 刘涵宇 周昊天 陈泊帆

## 一、前言

### 1.1 物流行业现状

随着互联网时代的到来，电子商务的崛起为传统的物流行业带来了井喷机遇。而近年来，以美团为首的外卖行业，更是将同城物流的巨大市场潜力展现在世人面前。然而，掩盖在物流市场容量迅猛增速之下的，是其依旧“劳动密集”的行业本质。

20 世纪初，福特率先在汽车工业领域开创了流水线，拉开了制造业无人化的序幕。时至今日，工业生产早已成为了机器的舞台。反观物流行业：如果城际物流的仓储、中转模式还能勉强称之为处于流水线时代的话；那城市内成千上万骑手小哥骑着电动车在商家和客户间来回穿梭的场景，恐怕只能称为福特之前的“原始时代”了。

### 1.2 无人物流的黎明

2021 年 8 月 26 日，交通运输部和科学技术部联合发布了《交通运输部科学技术部关于科技创新驱动加快建设交通强国的意见》，意见明确指出，要推动无人机（车）物流递送发展；要围绕人工智能、自动驾驶、无人机等领域研究推动相关立法。

事实上，各大物流巨头早已开始了无人物流的探索：

- 2019 年，顺丰就已经取得了城际支线无人机运输的许可。同样在探索城际无人机运输的还有电商物流一体化的京东。在他们构想的模式里，大型无人机将承担起跨城市、非主干线路的长途运输任务。这无疑是迈向物流无人化的一小步。只不过，这种长途的点对点运输，对人力的节省还是较为有限。在物流中，真正消耗大量人力的地方，还是城市内的配送。
- 2021 年，美团在深圳开始了城市内无人车及小型无人机的试点。美团采用“配送站到配送站”的模式。骑手从商户拿到外卖后，送至配送站，由无人车或无人机携带外卖，送至目的地配送站；再由目的地的骑手送货上门。这是首次在城市内物流的无人化尝试，不过它仍依赖骑手将物品送到用户手中——这便是困扰当前无人物流的、至关重要的“最后一公里难题”
- 放眼全球，美国的高科技巨头们同样在进行无人物流的尝试。亚马逊和 Alphabet 旗下的 Wing 是其中的先行者。它们将目标市场定为郊区城镇，并真正实现了无人化

的“送货上门”。美国的郊区域镇，都是带着大院子的别墅平房。无人机从商家起飞后，直接飞到用户的院子里投递物品，首次实现了商家与用户之间的“End to End”配送。不过这种模式也同样存在缺陷：无人机的载重较小，一次能运输的物品十分有限，一旦超过无人机运输能力，就不得不在商家与用户间多次往返。

### 1.3 深入分析无人物流

下文将采用问答形式，对无人物流进行深入探讨：

- 我们需要什么样的无人物流？

理想中的无人物流，是在不降低用户体验的前提下，尽可能节省人力，提高效率。在这里，“不降低用户体验”是重要的前提：如果原本可以由快递员送货上门，改用无人物流以后反而需要用户去配送点自提，这样的无人物流显然是难以推广的。而如果让快递员从无人配送点取货，再送货上门，这样的无人物流从“节省人力”的角度看效果堪忧。

- 无人物流采用何种运输工具？

考虑到运输能力与技术难度，第一位候选者自然是无人车。然而，无人车并不能实现“送货上门”，用户可能需要下楼甚至是走到小区门口才能取货。因此从目前来看，想解决无人物流的“最后一公里难题”，无人机是不可或缺的。

- 无人物流有没有技术上的困难？

布局无人物流的巨头有很多，之所以还未能推广开，技术上的困难是最主要的。当前，低速条件下的无人车技术已经基本成熟，主要难题还是无人机。

可既然美团已经能够实现了“配送站 to 配送站”的无人机运输，为什么不能进一步送到客户手上呢？事实上，在配送站之间运输，与“送货上门”难度天差地别，只需要给无人机画好固定的线路，它便能在 GPS 的导航下完成飞行。而若是要送货上门，则需要面对各种复杂的环境，这不仅需要无人机进行各种路径规划与避障操作，还对导航与定位的精度提出了极高的要求。

不过，为什么亚马逊和 Wing 的无人机能做到“送货上门”呢？这得益于他们选择的目标场景：美国（或澳洲）的城镇郊区。在那里，低矮且稀疏的别墅平房，不会像城市的高楼那样对 GPS 产生干扰。同时，每家每户都有的大院子，也大大降低了投递的精度要求——有点偏差没关系，只要投在院子里就行了。其实，Wing 的 CEO 也坦言过，他们的无人机还无法在大城市中投入使用。

此外，无人机在硬件层面也存在着困难：受电池等技术影响，无人机的承载与续航能力较为有限。如果单靠无人机运送，效率比较低下。

基于对无人物流的深入分析，我们团队设计并开发了一种基于无人机与无人车协同的精准投递智能物流工具，旨在解决无人物流当下的痛点，造福社会。

## 二、项目总览

### 2.1 系统概述

我们的无人物流配送系统由两个部分组成：

- 负责搭载无人机并批量运输物品的无人车
- 负责将物品送上客户阳台的无人机

二者的协作如同“航空母舰”：无人车搭载着无人机，完成前段运输任务；无人车抵达目的地后，无人机再从无人车上起飞，完成“送货上门”的任务。这样的分工既体现了无人车的优势：大载重，可批量运输物品；又发挥了无人机的特长：精准投放到客户手中，完成“最后一公里”任务。同时，无人车还可以为无人机充电，解决了无人机的续航问题。

### 2.2 运行场景示例

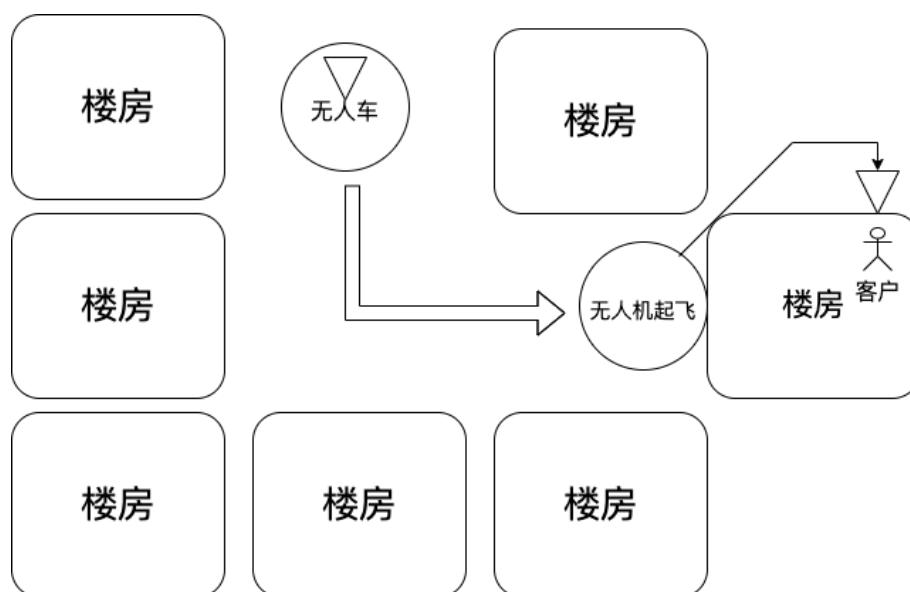


图 2.1：运行场景简图

我们取北大校园南部的宿舍区作为运行场景示例：

- 无人车搭载无人机，携带着大量该区域的物件，从南门驶入北大校园
- 根据客户的位置，无人车行驶至指定的宿舍楼下
- 无人车上的机械装置将物件推送到无人机上，无人机起飞

- 无人机飞到客户的宿舍阳台，降落并卸下物件
- 无人机飞回无人车上，并重复上述过程，为该区域内下一客户投送

## 2.3 技术路线

本节将概述项目的技术路线，提到的技术都将在第三章中详细介绍。

### 2.3.1 无人车

无人车需要具备的功能主要有两项，下文将逐一介绍实现方式：

- 自动行驶至目标楼下：

要完成该过程，首先要解决的是无人车的操作控制。为了缩短开发进程，我们使用了现成的小车底盘，它提供了现成的程序包，可直接调用接口控制其移动。同时，小车还搭载了电脑，我们在电脑上安装了机器人操作系统 ROS，用以沟通传感器、小车底盘以及控制程序。

接下来要解决的是无人车的定位与导航。我们使用了差分 GPS，为小车提供亚米级精度的定位。在知道自身定位，以及客户的坐标后，我们可以使用导航算法引导小车来到客户楼下。不过由于项目时间紧张，我们暂时采用了循迹算法作为导航的替代，即：预先手动操作小车走出一条轨迹，记录下沿路轨迹点；使用时再沿着采好的轨迹点自动行驶。此外，我们还在小车上使用了激光雷达，利用 Cartographer 算法进行建图与避障。

- 将车上的货物装载到无人机上：

该部分由简单的推杆结构实现，将货物从小车的货仓直接推入飞机货仓，再由飞机货仓上的舵机完成固定。

### 2.3.2 无人机

无人机部分是项目的主要难点，下文将逐一介绍技术路线：

- 无人机的控制：

我们在飞机上安置了机载电脑，上面同样运行着 ROS。同时，我们使用了 PixHawk 飞控板，其自带的控制系统可以接受机载电脑的指令，控制电机的转速并实现稳定飞行。

- 无人机的定位与自主飞行：

无人机的定位是整个项目最大的挑战。要想实现“精准降落在指定的阳台”，我们不仅需要知道客户阳台的坐标，还需要实时得知飞机自身的坐标，以便精准调控飞行轨迹。与小车不同：无人机飞进阳台，需要极高的精度；同时由于建筑物的遮挡，

使得 GPS 在靠近楼宇时会受到很多干扰。因此，我们必须采用 GPS 以外的方式得到飞机自身的定位。我们选择了 SLAM（实时建图与定位）技术。SLAM 算法主要有两类：一类使用激光雷达，一类使用视觉传感器。其中，激光雷达精度极高，但是价格昂贵；视觉传感器比较便宜，但是精度无法与激光雷达相提并论，且容易受到各种环境干扰。出于成本原因，我们使用了基于双目视觉+IMU（无人机上惯性传感单元）的 ORB-Slam 3 算法。只需使用无人机搭载相机，预先对整个场景建图，便可以在后续使用中，对无人机进行重定位，实时获取无人机在场景中的位置。由 SLAM 算法给出的定位，将与激光定高等工具得到的信息，通过 EKF（扩展卡尔曼滤波）进行融合，得到无人机的准确位置。知道自身位置以后，便可以调用现成的导航/循迹算法，控制无人机飞到阳台。不过，我们在尝试 SLAM 算法的过程中遇到了许多技术困难，目前还在努力攻克中。

- 无人机的降落：

虽然 SLAM 算法能帮助无人机飞到阳台，但进入阳台后，没有了预先建好的场景图，无人机的降落需要另外的辅助。我们使用了二维码辅助降落的方式：这需要客户的阳台贴有一块二维码，当无人机飞入阳台后，便不再使用 SLAM 提供的坐标系，转而采用地面上二维码的坐标系。飞机下方的摄像头将对准二维码降落在其正上方，并打开货仓，放下物件。

## 三、系统各部分详细介绍

### 无人车部分：

#### 3.1 无人车机械结构

##### 3.1.1 底盘结构

我们采用市面上较为成熟的 AutolaborPro1 小车产品作为我们无人车的底盘。它配备有较为成熟的车控程序，并安装了一个 Jetson nano 作为车载电脑，同时在车体前后安装有两个激光雷达，能实现四轮差速转向，负载能力达到 50kg，为我们后续的进一步开发和改造奠定了基础。



图 3.1: AutolaborPro1 外观图

### 3.1.2 交互结构

无人车与无人机货物之间的交互结构设计如下图所示。我们在无人车上设计三层平台：最底层为置物层，放置推杆与车载电脑、惯导、GPS 等必要器件；中间层为载货层，放置货物，可以通过履带传输，然后通过升降平台将货物推至最上层；最上层为停机层，无人机降落于此，通过无人机机架上舵机的开合取走货物（无人机机架设计见 3.3）。

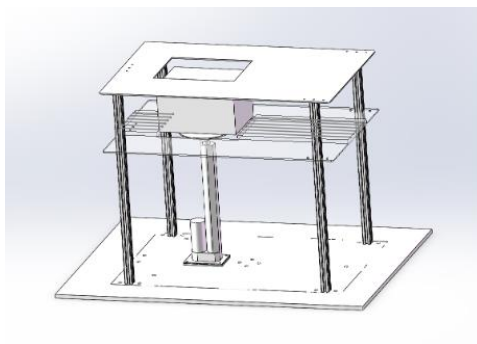


图 3.2: 小车供货平台设计图

支杆与平台的连接参考铝型材与角码标准件的连接方式（见图 3.3 和 3.4），铝板通过激光切割加工，并使用极客实验室的 3D 打印机打印载物平台（见图 3.5），采购并应用电动推杆作为供货核心动力（见图 3.6）。



图 3.3 & 3.4：铝型材角码与平台连接

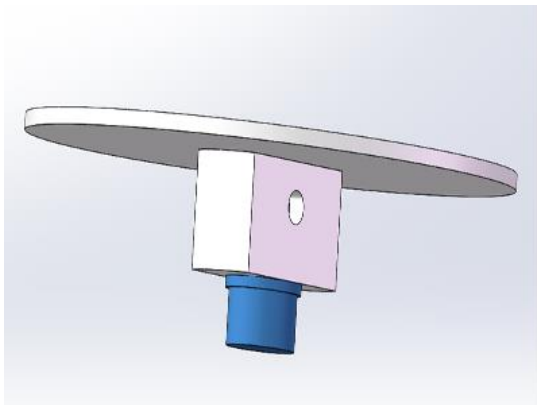


图 3.5 3D 打印载物平台

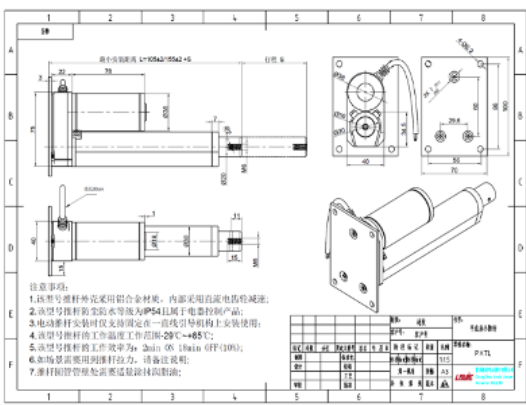


图 3.6 电动推杆（图源商家）

在具体实现过程中，为利用小车原有的支杆并节约经费，我们将最底层设计为单杆支撑。这导致了在运行过程中车体出现较明显的振动，我们将在未来重新设计成多杆支撑。同时，为避免改造小车底板，我们将推杆与货物放在了一层，这增加了小车的高度，未来若仍采用朴素的推杆设计，需要重新设计推杆与底盘的连接。下图为我们设计载货无人车实物：



图 3.7：无人车实物结构

## 3.2 无人车算法与代码实现

我们定义无人车算法部分的任务目标为从一个初始目标点（比如北大南门），载着无人机和货物，到达指定目标点（如宿舍楼楼下的停车位置）。我们认为此任务的难点主要有以下两点：

- 如何在室外宿舍楼底下达到一个较高的定位精度
- 无人车如何规划线路避开障碍物

### 3.2.1 利用差分 GPS 解决定位精度问题

对于第一个难点，我们采用了可以达到厘米级精度的差分 GPS 的数据信息来进行定位，当我们的无人车以一个较高的精度停靠在宿舍楼楼下时，可以有助于无人机基于原来存储地图进行重定位，并防止因为停放位置有较大误差而影响到交通。

在探索如何解决精度问题的过程中，我们进行了大量的尝试：首先，我们在室外测试了传统的 SLAM 导航建图方法，发现定位的效果一般。我们认为可能的原因是户外道路特征不明显，并且室外的障碍物布局时常变化与室内墙体固定布局的不同。接着，我们在室外场景尝试采用 GPS 进行定位。但是由于我们的终点设在寝室楼楼下，楼面的反射和遮挡会导致 GPS 的信号产生较大的偏差，很难实现高精度的定位。最后，我们选择了差分 GPS，通过使用 MD-649 挂载千寻账号为无人车提供亚米级甚至厘米级精度的定位数据。

差分 GPS 不同于普通的 GPS，它在移动接收站的测量得到的位姿信息上增加了一个修正值。这个修正值由一个已知精确坐标的地面基准站得到。地面基准站通过比对接收到的 GPS 经纬度信息和已知的正确经纬度，得到当前时刻此区域内的 GPS 测量偏差。接着将这个修正值作用于移动的接收站，进而得到更加精确的位姿信息。

差分 GPS 主要为我们提供以下数据：Heading（偏航角，与正北方向顺时针转过的角度，取值为 0-360）、Latitude（纬度，精度达到小数点后 7 位）、Longitude（经度）、Ve（东向速度，当静止时也会有漂动）、Vn（北向速度）、Baseline（基线长度，这里就是两根天线间距离）、NSV1（前天线收到卫星数）、NSV2（后天线收到卫星数）、Cs（校验位，采用的是异或校验）。我们可以很轻松的通过串口接收这些小车的位姿信息。经过我们的试验，在未外部基站（千寻账号）时，差分 GPS 精度达到米级（静止时经纬度小数点后 6 位飘动），设置千寻账号后，精度达到分米甚至厘米级。

### 3.2.2 利用循迹算法解决线路问题

对于第二个难点，我们决定采用如下方法：首先，我们提前由人工根据现场障碍物情况规划好无人车到每一个寝室楼的路径；然后，无人车根据所需投递包裹上的信息



选择一条通往目标宿舍楼的路径；最后，无人车根据已有的路径和自身收到的差分 GPS 实时位置信息进行循迹，最终停靠到目标宿舍楼的指定位置。因此，针对这个应用场景和传感器数据，我们设计了一个无人车循迹算法，最终实现了亚米级精度的效果。算法的主要思路如下：

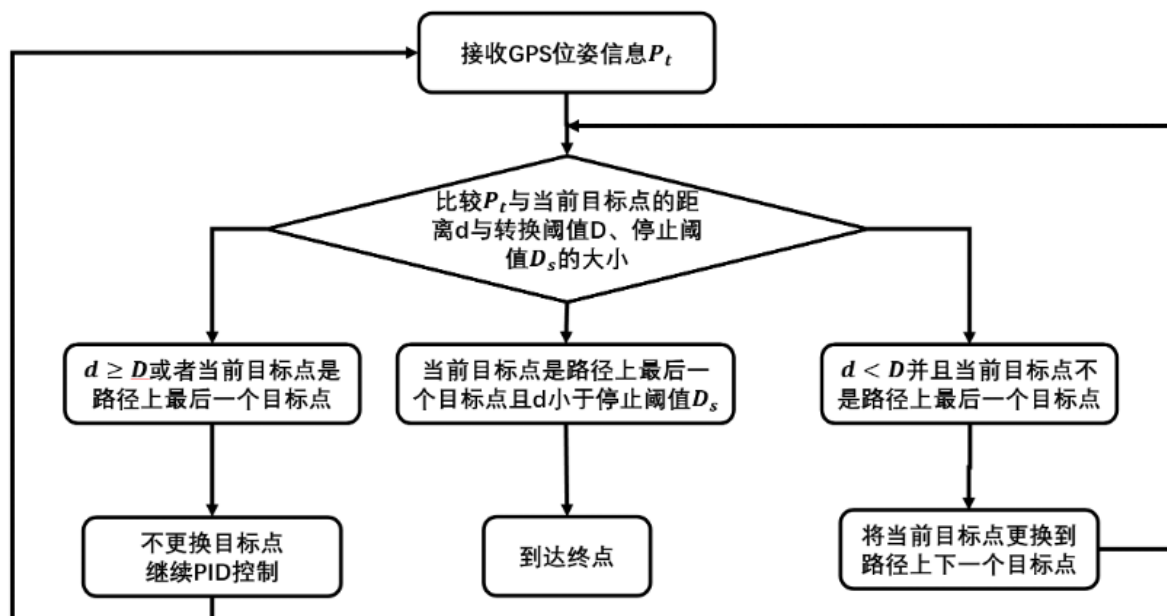


图 3.8：算法思路图

#### a. 路径信息的记录

我们首先通过手动遥控的方法确定一条无人车的路径，在遥控的过程中，我们每隔 1 秒记录一下差分 GPS 传输的无人车位姿数据。因此最终我们路径的数据形式是无人车经纬度坐标、速度、方向等状态信息构成的时间序列数据。

#### b. 路径跟随控制方式

为了完成无人车对我们记录的路径的循迹功能，我们决定使用一种不断更换目标点控制方式。具体而言，我们首先根据差分 GPS 的实时位姿信息和路径中首个时间点的位姿信息控制小车朝路径的起点移动。在这一步，我们主要使用了 P 控制的方法，保证它的位姿可以收敛在目标点。但是，这显然不能达成跟随路径移动的效果。因此在小车即将到达目标点时（我们会设定一个阈值），我们将目标点调整到路径时间序列中的下一个位姿。就这样，小车会随着我们事先设定的路径上点的序列，经过路径上一一个个关键点最终到达我们的目标位置。

#### c. 算法参数调试的目标

显然，基于上述思路实现的代码是需要经过进一步的调试的。调试的第一个参数是目标点更换的临界值。当这个临界值比较小时，也就意味着无人车将在非常靠近目

标点时才更换到下一个目标点，而且下一个目标点的位置依然非常近。这必然会导致无人车的路径跟随出现弯弯曲曲，速度也不平稳的情况。当这个临界值比较大时，无人车在离目标点较远的位置就会更换到下一个目标点，会造成路径跟随的相似程度较低。特别是在转弯时，较大的阈值会增大无人车转弯的曲率半径，可能会撞到拐角处的障碍物。因此，无人车路径跟随时目标点的临界值需要权衡两者的利弊进行考量。

ROS 是机器人设计中常用的操作系统。我们基于 ROS 写出了算法的实现代码，它的具体结构如下：

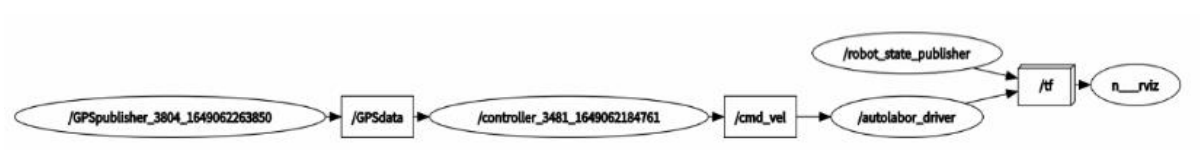


图 3.9: rqt 节点图（循迹）

- a. 由 GPSpublisher 节点接收差分 GPS 信号，发布此数据的话题。此节点的作用是将接收到的位姿数据经过修正后分配给预先设置好的存储结构，然后发布话题 GPSdata，并把接收到的数据显示在终端上。（如下图）

```

[INFO] [1649062485.716183]: GPSWeek: 0.0
GPSTime: 3222.0
Heading: 0.0
Pitch: 0.02
Roll: -0.78
Latitude: 4.5066867099e-06
Longitude: 1.53005088563e-06
Altitude: 0.0
Ve: 0.0
Vn: 0.0
Vu: 0.0
Baseline: 0.0
NSV1: 0.0
NSV2: 0.0
Status: "00*56\r\n"
  
```

图 3.10: 终端显示的 GPS 接收数据

（此图为差分 GPS 非正常使用状态的演示图片，因此数据基本上是 0）

- b. 由 controller 节点订阅 GPSdata 话题，并发布 cmd\_vel 控制话题。此节点是控制算法的主要节点，它的作用是根据 GPSdata 话题下的无人车位姿数据和存储在电脑上的路径信息，计算出无人车当前瞄准的目标点。然后根据 P 控制，计算出无人车的线速度和角速度，发布到 cmd\_vel 话题下。最后把无人车的控制速度、目标点编

号、此时的位姿、目标位姿显示在终端，以便于调试。（如下图）

```
[INFO] [1649062393.793688]: linear:
  x: 0.4
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -0.355426287121
point:462153.0
(12947195.306217419, 4436537.984235683)
(0.0, 1.0)
```

图 3.11：终端显示的小车控制数据

（此图为差分 GPS 非正常使用状态的演示图片，因此数据有较大偏差）

- c. 由 autolabor\_driver 节点接收无人车 cmd\_vel 话题的控制数据，并实现对车体的控制。Autolabor\_driver 为小车自带的车控节点，其主要功能是接收我们发送的控制信息，并完成对底盘的操控。

除了控制算法之外，我们还编写了实时记录小车路径的节点（routecollect），我们可以在初始规定小车路径的时候运行它，将小车路径按照时间顺序存储到指定的 csv 表格中。

### 3.2.3 室外实现效果和反思

根据上述代码，我们最终成功地完成了小车在室外的寻迹，其精度达到了亚米级。可以说，我们成功地初步实现了预期物流系统中无人车完成的功能。试验效果可以参考附件中的“试验视频展示”文件。

根据试验结果，我们依然存在比较大的改进空间。简而言之，主要有以下几点：

- a. 无人车机械结构有待改进。为了进一步提高差分 GPS 的数据精度和准确度，我们在未来需要适当增加差分 GPS 两个天线之间的距离，最好可以达到 2 米以上（目前 1 米）。同时，我们可以适当抬高 GPS 的天线位置，以防止地面信号反射的影响。
- b. 算法参数设置有待改进。视频中的循迹依然存在较为明显的弯折，我们可以适当提高目标点转换的阈值，让无人车行驶的路径更加平滑。
- c. 对于路径终点控制的处理有待改进。视频中我们发现，在小车非常接近终点后，存在较长时间的调整。我们分析这可能和 GPS 接收的数据存在波动有关。由于这些波动，让小车认为自己始终没有到达终点，从而继续直行或者旋转，这反而有可能加大了与终点之间的偏差。因此，我们在未来可能可以尝试适当放宽精度要求或者针

对小车到达终点的情况专门设计一个算法进行优化，以此提高这个循迹算法在终点的处理速度。

同时，我们也必须承认，我们现有的技术依然存在较大的局限性，比如无人车在户外无法完成对行人的自动避障、差分 GPS 价格较贵、无人车路径设置固定且单一等，我们将在未来努力探索更加先进的技术来满足更智能更快捷的物流需求。

### 3.2.4 室内演示的 SLAM 功能

出于室外演示的安全性考虑，我们在室内也为无人车设计了完成自动导航功能的模式。由于室内接收不到 GPS 信号，我们主要采用无人车前后激光雷达的相关数据，利用传统的 SLAM 方法进行室内的建图导航。SLAM 建图导航的 rqt 节点图如下两图所示：

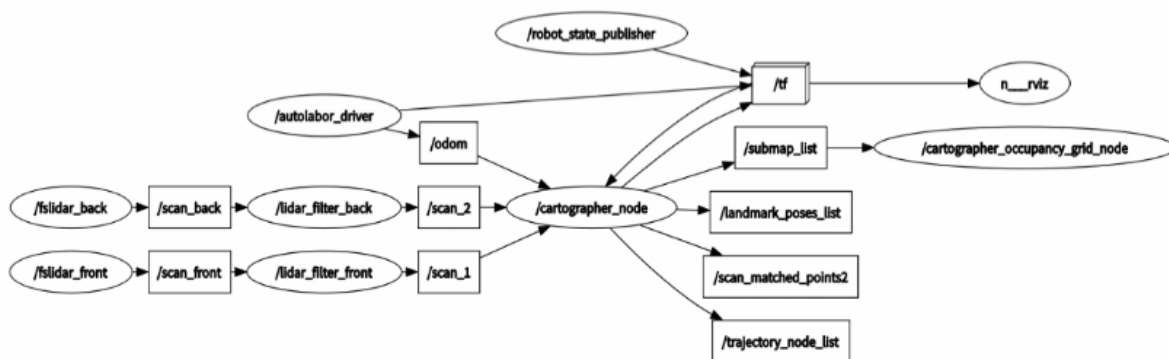


图 3.12: rqt 节点图（建图）

- 建图：由 cartographer\_node 节点接收无人车前后激光雷达的数据和车控自带的里程计数据，输出无人车周围环境的地图信息。

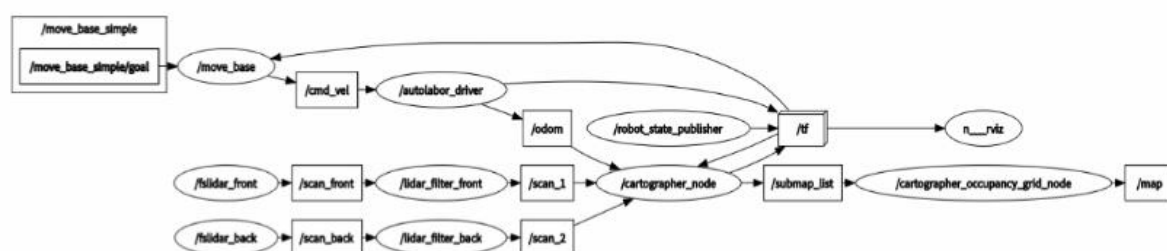


图 3.13: rqt 节点图（导航）

- 导航：根据存储的地图信息和现在接收到的地图信息，由 move\_base 节点对数据进行对比，从而确定小车的位置，进而根据我们设置的终点进行路径规划并导航。

根据 SLAM 建图导航的相关集成好的 ROS 节点，我们可以在事先建立好室内地图的情况下，使小车自主规划路线到达指定的停靠位置，完成运送无人机和货物的功能。试验效果可以参考附件中的“试验视频展示”文件。

## 无人机部分：

无人机是我们这个物流投递系统的重要组成部分之一，它承担着物流“最后一公里”中用户端投递的最后一个环节的任务，即从无人车上搭载货物，然后执行从无人车向用户接收货物处所（例如阳台等）的最后输运任务。为了完成我们预想的执行效果，我们基于自行设计组装的无人机机架和开源的 Ardupilot 固件设计了这一物流投递系统的无人机部分。本节将分两部分详细介绍我们的无人机从硬件到软件的设计与相应功能的实现。

### 3.3 无人机机架结构设计

在第一部分中，我们将主要聚焦于阐述我们的物料投递系统中无人机机架结构的设计。在前期的考察调研之中，我们发现现有市面上能够满足我们这一物流投递系统功能要求的无人机屈指可数，且不具备较好的开源性、可改造性，不利于二次开发实现我们个性化的功能和需求。因此，我们选择自行设计加工无人机机架，以此更好地适应我们对于开发一个物流投递系统的需求。

#### 3.3.1 整体结构与部件组成

我们在整体机架结构设计上始终秉承降低制造难度、增加可扩展性的思路，这主要体现在以下两个方面：首先，我们的无人机主体部分采用以“榫卯”结构为主的设计（见图 3.14），通过平面板件之间的拼插和限制达到由二维零部件组装出三维结构的效果，这在一定程度上使得加工方式只用切割即可完成，不需要额外的模具设计，制造的难度大大降低。其次，我们在设计时将无人机进行了“分层”设计，可以根据需求增加额外的平台容纳无人机所需的新传感器等零部件，加强了无人机的可扩展性。

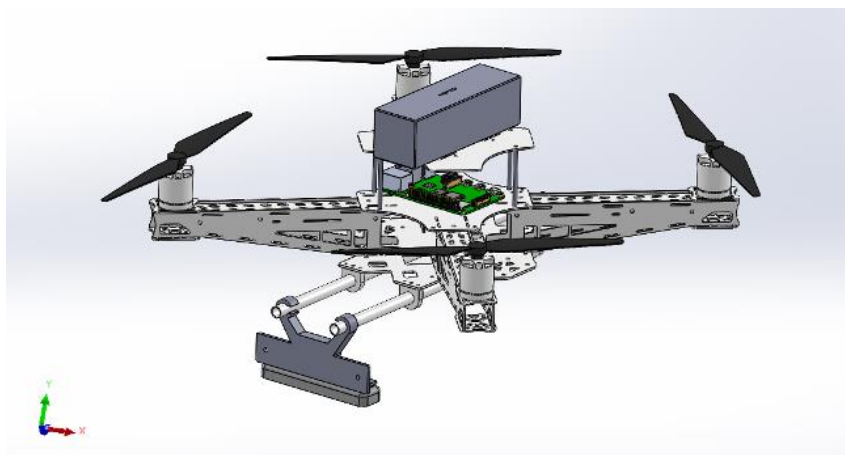


图 3.14：无人机机架 SOLIDWORKS 预装配体（不含脚架）

整体机架结构（见图 3.15）由上至下可以大致分为 5 个区域，分别是电池动力区



(安装电池)、机载电脑区 (安装机载电脑)、主体“榫卯”结构 (安装飞行控制器、分电路板、电机、电调等)、拓展平台 (悬挂视觉传感器) 和料仓脚架 (安装舵机)。

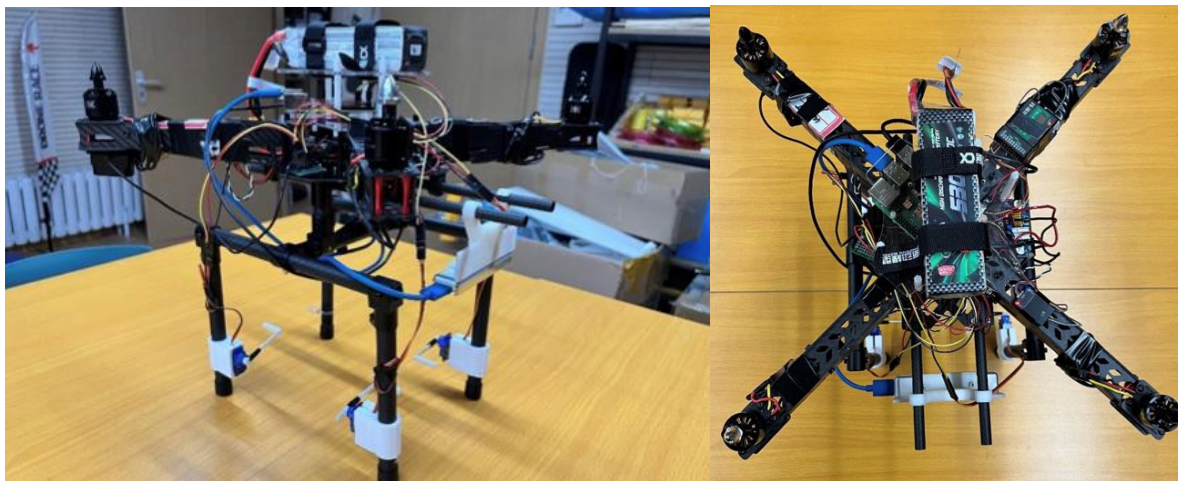


图 3.15：无人机实物图

为了实现整个货物投送的功能，我们对无人机各个零部件进行了优化，在我们设计的无人机上集成了众多性能优良的组件，包括树莓派 4B（用作机载电脑）、PixHawk4 飞行控制器、英特尔 RealSense T265（用作二维码识别与无人机定位）、TFmini plus 单点激光雷达（用作高度测定）、格氏 4000mAh 4S 电池、YH 2212 自锁电机、好盈乐天 20A 电调、SG-90 舵机（用作物料投递）等。得益于自行设计的无人机机架，我们可以在上面搭配各种功能的组件，实现我们预期的效果。

此外，由于无人机的飞行对于自身重量要求颇高，我们在选料时以碳纤维为主要材料，辅以亚克力和成分为聚乳酸（PLA）的 3D 打印件。

### 3.3.2 各部分功能实现

我们的无人机机架共分 5 个区域，下面分别介绍每一个区域的结构设计以及相应的功能实现：

- 电池动力区位于最顶层，其上用绑带固定电池，主要负责整个无人机的供电。在设计电池动力区时采用的就是“分层”的设计原则，电池层可以随时从无人机上拆下，便于充放电。
- 机载电脑区位于第二层，用于放置树莓派 4B，它位于无人机中心部位，可以通过数据传输线、杜邦线等连接飞行控制器、RealSense T265 和舵机等部件。
- 主体“榫卯”结构中安装有飞行控制器、分电路板、电机、电调、单点激光雷达、遥控器接收机，是飞机的主体部分，其中飞行控制器、分电路板在图中不可见，位于无人机内部。该部分主要实现的是无人机的动力控制。
- 拓展平台为管状设计，通过“挂载”的方式可以在其上安装传感器。通过自制 3D 打

印件，我们在其上配置了一个朝向地面的视觉传感器 RealSense T265，它可以实现对于地面二维码停降区的识别以及对于无人机自身位姿的估计。同时管状的设计便于视觉传感器的延申，在无人机下方配有物料时也不容易被遮挡，便于功能的实现。

- e. 料仓脚架是一个集无人机脚架与料仓于一体的设计，为了尽可能地减少无人机在结构上的重量负担，我们大胆地尝试将脚架与料仓融为一体，通过碳纤维管连接的方式预留出货物的盛放空间，依靠四个脚架部位的舵机实现对于物料的固定，同时又借此实现对无人机的支撑作用（详细的演示可以参见试验视频展示）。

### 3.4 自主飞行技术

在无人机配送的过程中，最重要的一环就是如何让无人机识别自身位置，获得路线信息，并安全的降落在既定位置。为了实现这一目标，我们设计了激光定高+视觉 SLAM 定位帮助无人机实现稳定飞行，并采取 AprilTag 二维码标记的方式达到精准降落的效果。

#### 3.4.1 激光测距定高+EKF 数据融合

对于所有无人机来说，确认高度都是不可缺少的一环，在本次无人机配送的任务中，这一点尤其重要。从 GPS 到气压计到光流设备，市面上的无人机使用了各种各样的设备来辅助高度的测定，但是这些设备即使较为精准也难以达到分米级精度，这对于配送到阳台的无人机来说显然是远远不够的。为了解决这个问题，我们采用了精度达到 1cm 的 TFmini plus 单点激光雷达用于测定无人机与地面的高度，给予无人机最精准的高度信息。

为了保证系统稳定性，我们的无人机系统还同时采用了气压计与惯性测量单元 IMU 获取的高度数据，采用扩展卡尔曼滤波（EKF）算法将多种数据来源的高度信息进行融合，保证数据的稳定性。下图是飞行过程中一段实测数据，以便说明 EKF 融合的结果。



图 3.16: TFmini plus 单点激光雷达数据融合图

本段数据分析对应附件中试验视频展示中无人机在试验场地内的起飞与识别二维码起落片段，其中 RFND（绿色线）是单点激光雷达的测量数据，而 CTUN（红色线）是 EKF 融合的高度数据。二者趋势基本一致，且相差不大，说明单点激光雷达经过调试，其数据已经被利用起来，用作无人机飞行的辅助。上图横轴为时间，上面标注出了各种飞行模式的切换时间点。

### 3.4.2 基于 RealSense T265 鱼眼相机的 ORB-Slam 3 建图与定位

对于任何机器人来说，导航通过陌生的空间，同时避免障碍物都是极其重要的。由于本任务中无人机配送经常处于高楼之间或是半室内地区，GPS 信号所能达到的精度和可靠程度都远远不足，因此需要一个新的更加可靠的方法来实现为配送无人机的空间坐标定位。而依靠视觉惯性里程计（VIO）的实时定位建图是一种能够提供准确、可靠的无 GPS 定位和导航的技术，它将来自摄像机和惯性传感器的信息结合起来，告诉无人机自身相对于环境的位置。

SLAM 所做的事情可以描述为：机器人在未知环境中从一个未知位置开始移动，在移动过程中根据位置和地图进行自身定位，同时在自身定位的基础上建造增量式地图，实现机器人的自主定位和导航。通过 RealSense T265 鱼眼相机和惯性测量单元 IMU 的配合，我们的 SLAM 得以使用环境中的视觉特征较为准确地跟踪它在未知空间周围的路径。

根据视觉信息和 IMU 进行 SLAM 建图定位所得出的位置坐标，在加上 EKF 融合激光雷达数据所得到的高度信息，我们得以获得精准的空间坐标信息，允许无人机根据此坐标信息在无 GPS 的条件下也能达到定点飞行的效果。



不过，在尝试 SLAM 算法的过程中，我们遇到了许多技术上的困难，目前还在努力解决中。

### 3.4.3 AprilTag 二维码识别定位

无人机载物降落在目标点的过程需要无人机根据目标点位置实时调控自身位置，以便成功精准降落。由于 VIO-SLAM 在长距离飞行过程中容易累计误差，仅仅依靠 SLAM 所计算的坐标难以完成精准的降落，因此在降落过程中需要更加高精度的定位方式 AprilTag。

AprilTag 作为一种视觉定位标志符，类似于二维码或者条形码。在视觉导航的位姿测定中，我们选取地面上的 Tag 标识作为世界坐标原点（即其坐标已知且固定），而 RealSense 摄像头所测得的图像经过机载计算机处理可以得到一个以摄像头为标准的图像坐标数据，这个图像坐标系实际上正是由于无人机把地面的 Tag 进行了一系列姿态旋转和位置平移后所形成的，并且图像坐标和世界坐标都是已知的，因此只要我们找到联系这两个坐标系的变换矩阵，自然就确定了无人机旋转和平移的信息。

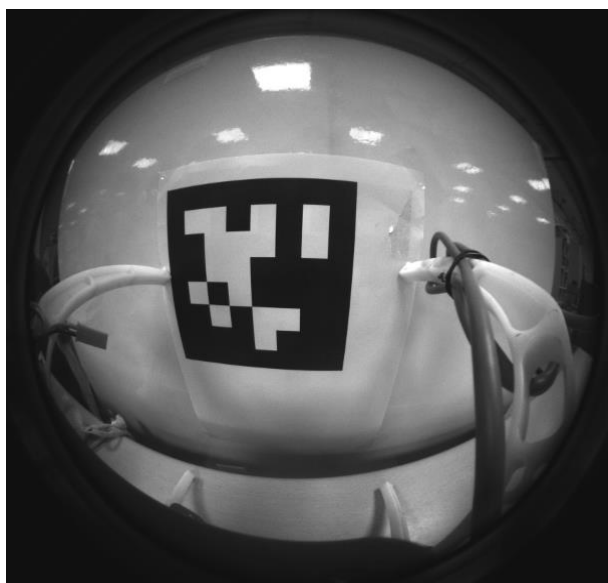


图 3.17：T265 鱼眼图像中的二维码

首先，由于 RealSense T265 是广角镜头，它所获得的图像是鱼眼图像，如上图所示，无法直接进行二维码识别，我们需要使用 OpenCV 对视频流进行处理，以便消除图像的失真，获得原始的图像数据，如下所示：



图 3.18：处理后的二维码图像

随后，我们在未失真的图像上运行 AprilTag 检测算法，如下图所示，以获得标签相对于摄像头的相对位姿，经过坐标系的变换矩阵即可获得降落点相对无人机坐标系的相对位置，从而得知无人机下一步的飞行路线，完成精准降落投料。

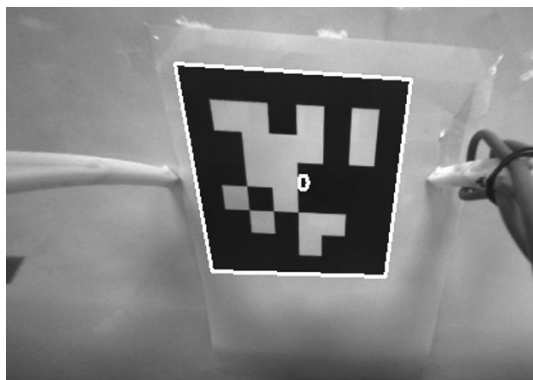
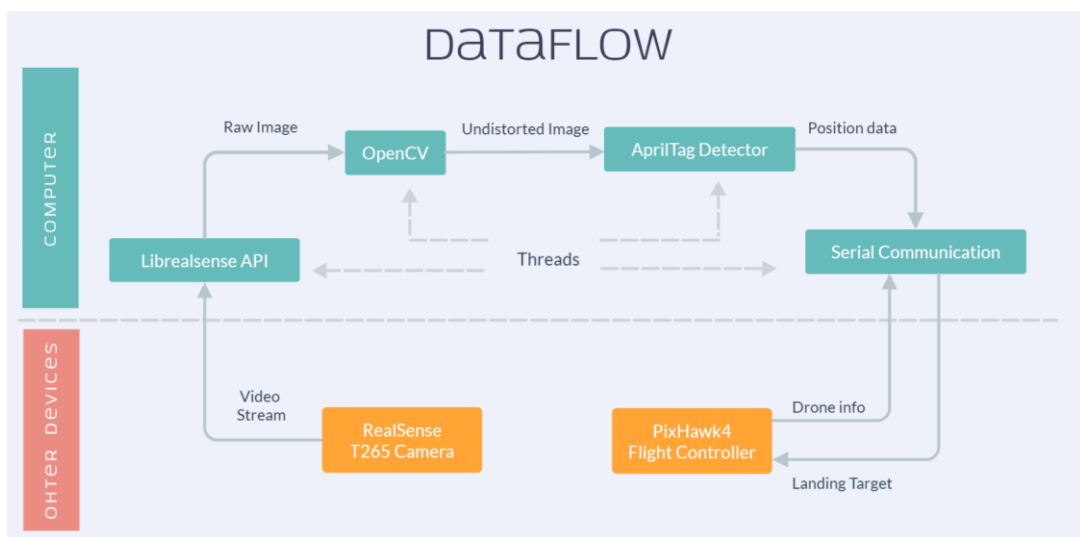


图 3.19：AprilTag 检测算法检测到的二维码位置

(图中白色方框框定了二维码范围，白色圆圈指定了二维码中心)

### 3.4.3 机载电脑数据处理及数据发送



图

3.20：数据流示意图

机载电脑同时要 RealSense 数据接收，鱼眼图像还原，AprilTag 检测等多项任务，为了最大化利用机载电脑四核处理器性能，并尽可能提高数据发送的频率，我们选择利用线程级并行构建图像处理流水线，分别设置了 4 个独立线程负责通过 librealsense 提取相机原始视频数据；还原鱼眼图像；进行 AprilTag 检测；发送数据并监测飞控信息，并使用了 mutex 互斥锁来保证线程间共享数据 buffer 的访问是互斥的。最终在飞控的结果显示 LANDING\_TARGET 数据接受的频率达到了较为稳定的 15Hz，足以引导无人机进行稳定飞行以及降落的操作。虽然相比相机的 30Hz 输出频率一定差距，但是对比单线程运行结果，已经有接近 1 倍的性能提升。

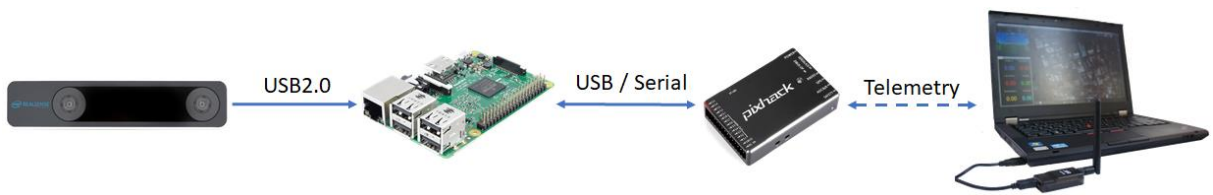


图 3.21：无人机部件间通信方式

RealSense 视频流处理和 AprilTag 识别均在机载电脑上完成，而无人机的飞行姿态与轨迹是由 PixHawk 飞行控制器控制的，因此我们需要将机载电脑上所处理得到的位置信息与目标信息实时传输到 PixHawk 飞控上。为了实现这个目标，我们采用了波特率 921600 的串口通信，并使用了 MavLink 协议来编码发送数据内容，将所获得的数据转换为对应的 uint 及 float 格式编码在对应位置上，即可获得 LANDING\_TARGET 以及 VISION\_POSITION\_ESTIMATE 可发送的数据包，并将其发给飞控模块。

LANDING\_TARGET ( #149 )

[Message] The location of a landing target. See: [https://mavlink.io/en/services/landing\\_target.html](https://mavlink.io/en/services/landing_target.html)

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
target_num	uint8_t			The ID of the target if multiple targets are present
frame	uint8_t		MAV_FRAME	Coordinate frame used for following fields.
angle_x	float	rad		X-axis angular offset of the target from the center of the image
angle_y	float	rad		Y-axis angular offset of the target from the center of the image
distance	float	m		Distance to the target from the vehicle
size_x	float	rad		Size of target along x-axis
size_y	float	rad		Size of target along y-axis
x **	float	m		X Position of the landing target in MAV_FRAME
y **	float	m		Y Position of the landing target in MAV_FRAME
z **	float	m		Z Position of the landing target in MAV_FRAME
q **	float[4]			Quaternion of landing target orientation (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
type **	uint8_t		LANDING_TARGET_TYPE	Type of landing target
position_valid **	uint8_t			Boolean indicating whether the position fields (x, y, z, q, type) contain valid target position information (valid: 1, invalid: 0). Default is 0 (Invalid).

图 3.22：MavLink 数据包格式

## 四、项目下阶段展望

经过不懈的努力，我们的项目已经取得了不少阶段性进展，但这离我们想要实现的目标依然存在距离。尽管挑战杯已经接近尾声，我们仍计划在未来继续攻坚克难，完善这个项目。

下一阶段，我们将把主要目光放在无人机的定位与自主飞行上，重点解决“从小车上起飞并飞到阳台”这一过程。当前我们在这一阶段碰到的困难主要是定位的稳定性。这导致了飞行的不稳定，尽管发生频率较少，但考虑到安全问题，我们仍不敢在室外进行试验。

解决了这个难题后，我们便可以将实验场地移至室外，更好地调试我们的整个系统。希望在未来，我们的无人物流系统能真正走出实验室，造福千家万户！

## 五、致谢

本课题是一项十分偏重工程实践的项目，这对于我们这个由本科生组成的团队来说，充满了挑战。在过去半年多的课题准备与实践的过程中，有很多老师和同学给予了我们帮助和指导，让我们团队得以顺利地推进项目的实施。在此我们团队全体成员向他们表示由衷的感谢：感谢工学院王雪峰老师在项目全程中对于我们从选题到实施的全方位指导；感谢信息科学技术学院查红彬老师和其实验室的师兄们给予我们关于视觉 SLAM 方面的指导，并感谢查老师借给我们 AutolaborPro1 无人小车平台进行试验；感谢信息科学技术学院赵卉菁老师给予我们在选题上的启发以及对于无人车自动驾驶技术方面的指导。除此以外，还要感谢工学院对本项目提供的慷慨的资金支持，感谢北京大学极客实验室提供的场地支持。