

面向多无人机协同飞行控制的云系统架构

史殿习^{1),(2),(3)} 洪 臣⁴⁾ 康 颖^{2),(3),(5)} 金松昌^{2),(3)} 张拥军²⁾ 杨 东⁵⁾

¹⁾(国防科技大学计算机学院 长沙 410073)

²⁾(军事科学院国防科技创新研究院 北京 100166)

³⁾(天津(滨海)人工智能创新中心 天津 300457)

⁴⁾(中国人民解放军 61932 部队 北京 100071)

⁵⁾(中国人民解放军 31401 部队 长春 130022)

摘 要 有限的载荷使得无人机难以承载过多的仪器设备,其极大地限制了无人机上资源的交互、共享以及无人机遂行任务多样性的能力.多无人机智能协同成为 AI 领域发展的主流方向,但已有的多无人机系统存在通信距离受限、数据资源分离等问题,且因网络连接的不稳定性及数据传输的延迟性,多无人机间难以实现有效的协同控制.本文提出了一种面向多无人机协同飞行控制的云系统架构 UAV3CA. UAV3CA 在云端为多无人机构建了一个“远程大脑”,通过 PaaS 抽象封装无人机及其上资源,可实现多无人机间信息数据的实时共享,并基于云端强大的存储和计算能力完成多无人机的统一规划及协同飞行控制.实验从单无人机制导和多无人机协同两个方面,对 UAV3CA 在网络通信的稳定性、数据传输的实时性和协同控制的有效性等方面的性能给予了验证,最终以完美的编队队形呈现出 UAV3CA 协同控制方式的灵活多样性.

关键词 多无人机;云系统架构;协同飞行控制;实时共享

中图法分类号 TP393 **DOI 号** 10.11897/SJ.1016.2020.02352

Cloud-Based Control System Architecture for Multi-UAVs Cooperative Flight

SHI Dian-Xi^{1),(2),(3)} HONG Chen⁴⁾ KANG Ying^{2),(3),(5)} JIN Song-Chang^{2),(3)}

ZHANG Yong-Jun²⁾ YANG Dong⁵⁾

¹⁾(School of Computer Science, National University of Defense Technology, Changsha 410073)

²⁾(National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100166)

³⁾(Tianjin Artificial Intelligence Innovation Center, Tianjin 300457)

⁴⁾(People's Liberation Army of China No. 61932, Beijing 100071)

⁵⁾(People's Liberation Army of China No. 31401, Changchun 130022)

Abstract Unmanned aerial vehicle (UAV), as a prior branch of intelligent unmanned system, has attracted a wide range of interest from different kinds of application domains, including real-time monitoring, providing wireless coverage, remote sensing, search and rescue, delivery of goods, security and surveillance, precision agriculture, and civil infrastructure inspection etc. However, there are several new challenges arising from more and more complex application environments and scenes, which setting a higher demand for the extensible capacity of UAV, especially for some properties of UAV such as small size, light weight, enduring energy, processing capability, signal transmission range and so on. The limited processing power and payload of UAV are the most challenging issues, which restrain the on-board requirements like sensor weight, size and required power. As a result, it is difficult for UAV to carry on plenty of instruments and

收稿日期:2019-06-17;在线发布日期:2020-01-18. 本课题得到国家重点研发计划(2017YFB1001901)资助. 史殿习,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为人工智能、分布式计算、云计算及大数据处理等. E-mail: dxshi@nudt.edu.cn. 洪 臣,硕士,主要研究方向为人工智能、分布式计算及云计算等. 康 颖(通信作者),博士,助理研究员,主要研究方向为人工智能、数据挖掘、社区发现等. E-mail: kangying_841218@sina.com. 金松昌,博士,助理研究员,主要研究方向为人工智能、数据挖掘、社区发现等. 张拥军,博士,研究员,主要研究领域为高性能计算、分布式应用系统等. 杨 东,学士,主要研究方向为有线通信、信息安全防护等.

equipments and hardware processors, which imposes dramatical restrictions on the exchange and share of resources between UAV, and on the capability of UAV to execute diverse tasks like tracking multiple objects, exploration of complex terrain and so on. To address the problems mentioned above, many solutions have been promoted, of which one mainstream on AI development is to make multi-UAVs flying in some formation and completing jobs in the form of intelligent cooperation, while breaking through the intrinsic limitations of individual UAV in payload, enduring energy, computing and memory power. However, the existed systems of multi-UAVs lack of collaboration have been faced up with such problems as the confined distance of wireless communication and separation of kinds of data sources. In addition, the cooperative control on multi-UAVs is hard to realize because of the instability of network connection as well as the latency of data transmission. In light of the deficiency of synergy of multi-UAVs, this paper proposes a cloud-based control system architecture for multi-UAVs cooperative flight (short for UAV3CA). The main idea of UAV3CA is to configurate a “remote brain” for multi-UAVs based on the construction of cloud, equally meaning that by virtue of the abstraction and encapsulation of the resources from UAVs or on board of UAVs based on PaaS (platform as a service), the information and data among multi-UAVs are able to share in real time. For another thing, the powerful storage and computing ability of the cloud put it into reality that enabling the multi-UAVs to plan in a unified manner and controlling the cooperative flight of multi-UAVs. Abundant experimental results from two differ designed aspects, of that one for singleton UAV guidance and the other for multi-UAVs cooperation, demonstrate the excellent performances of the presented UAV3CA, which are concretely shown as the network communication becoming stable without interruption, the data transmission reaching one another among multi-UAVs system in time, and achieving the valid cooperative control on multi-UAVs by means of the resources shared on cloud. In the end, the perfect display of formation evolution of five unmanned aerial vehicles in the virtual environment has accounted for the flexibility and multiformity of cooperative control patterns of multi-UAVs based on the proposed UAV3CA.

Keywords multi-UAVs; cloud-based system architecture; cooperative flight control; real-time sharing

1 引言

智能无人系统是人工智能(AI)领域的重要应用之一,其按照适用场景的不同分为无人车、无人机、空间机器人、海洋机器人、无人工厂机器人等不同类别^[1],其中以无人机(Unmanned Aerial Vehicle, UAV)的应用最为典型且广泛.作为一种新兴产业和最具发展前景的 AI 技术,无人机不仅潜移默化的改变着人们的生活模式,还将在未来的产业或商业乃至民用、军事领域引发一场革命性变革.无人机因其具有成本低、全天候、零伤亡等优势,不仅能够取代人完成简单的重复性劳动,如边境巡逻、精准农业、货物运送、道路交通实时监管等,还能够在复杂环境或危险场景中替代人完成特殊难度的任务,如遥测遥感、建筑与公共设施巡检、事故灾害现场勘测

及救援等^[2].但随着应用需求的不断扩展,实时多变、纷繁复杂的任务场景对无人机提出了更多更高的要求和挑战.面对单无人机有效载荷低、数据信息整合能力有限等情形,多无人机集群化、协同化、智能化成为无人机技术发展的一种必然趋势.

体积小、重量轻是无人机的优势,但同时制约着无人机自身有效的载荷能力,使之难以承载过多的仪器设备和硬件处理器.目前,已有的无人机操作系统主要是以无线局域网或 WiFi 作为传输媒介,通过遥控器或者软件 APP 建立端到端的连接并施加控制,其最大的局限性是通信距离受限、可操作性不强,且不支持多架无人机同时接入同一个操控端局.因此,如何突破上述瓶颈成为亟待解决的问题.扩展无人机的体积以增加其有效载荷及存储计算能力或许是一种方法.但如果将多架无人机组成一个集群,使之协同飞行并执行任务,其将具备单无人机无

可比拟的优势.正如 Humboldt 等人^[3]所言,利用有限的资源创造无限的价值(make infinite use of finite means),大量低成本的无人机通过有机的组合,会产生群体智能效应,这将极大地推进无人机潜在的应用能力.多无人机协同技术的研究未来可用于多个领域,其控制系统架构具有广阔的应用前景,并将在国民经济中产生巨大的经济效益.

无线网络的飞速发展和物联网(Internet of Things, IoT^[4])的广泛推崇,使得研究人员将云计算的概念引入无人机领域,形成“云+无人机”的概念.追溯到 20 世纪 90 年代,东京大学 Inaba^[5]提出了远程大脑的概念,即提议将机器人与计算机通过万维网连接——发展“网络机器人”.这一概念实现的典型案例是 Goldberg 等人^[6]通过 Internet 浏览器远程控制网络中的机器人;随后, Kim 等人^[7]采用 UDP 协议连接无线传感器和群体机器人实现基于网络连接的控制.网络机器人仅仅是通过网络连接实现多机器人的简单相加,并没有从根本上解决机器人的物理瓶颈问题.在 2010 年的 Humanoids 大会上,卡耐基梅隆大学的 Kuffner 博士^[8]首次提出“云机器人”的概念,即机器人可利用互联网上的云服务器进行分布式并行计算和大数据存储、分析与处理,机器人本身无需能力扩展,若想获取信息,连接服务器即可.基于这一概念,很多研究尝试将繁复的计算卸载到云端来解决机器人自身存储计算能力受限的问题. Arumugam 等人^[9]在 Hadoop^[10]集群上采用 Map/Reduce^[11] 计算框架来增强机器人算法执行的有效性,但该方法没有给出可靠的通信机制. Hunziker 等人^[12]设计了一个开源的平台即服务机器人应用框架 Rapyuta,提供了一种安全的可定制化的计算环境以帮助机器人将繁重的计算任务转移到云端.上述两种方法仅专注于解决机器人计算层级的问题,协同控制的讨论没有深入. Kuriki 等人^[13]基于 c 共识协议提出了一种具有避碰能力的协同编队控制结构,其核心关注点在于如何解决多无人机间的避碰问题.

随着无人机角色在机器人领域的不断增强,人们开始将无人机部署到云端. Gharibi 等人^[14]基于 IoT(Internet of Things^[4])提出了 IoD(Internet of Drones)的概念模型并规划了一个设计框架. Koubaa 等人^[15]提出了一个软件架构 Drone Map Planner(DMP),通过具体的实现和实验证明了 IoD,即基于互联网的无人机云处理架构,但该方法一次仅支持访问控制一架无人机. Lin 等人^[16]提出将无人机连接到 Google Earth 云服务,并将数据存储于 MySQL 数据库中,用户通过 Web 浏览器可访

问数据库中的无人机信息,该方法没有阐明多无人机控制问题,且通过数据库监控无人机将影响系统的实时响应能力. Sanchez-Lopez 等人^[17]提出了一种在复杂环境中构建自主无人机系统的控制架构 Aerostack, Aerostack 具有管理大量无人机的能力,但没能实现多无人机间的全局调控. Sampedro 等人^[18]设计了一个全局协调器 Global Mission Planner(GMP),弥补了 Aerostack 无法协调多无人机执行任务的缺陷,其可动态地将任务分配给每架无人机,同时监控无人机集群行为,但与负责上层任务分配的本地协调器 Agent Mission Planner(AMP)的异步通信却制约着整个系统的实时性. Dong 等人^[19]提出了一种基于局部邻域信息的多无人机编队控制协议,研究多旋翼无人机系统的编队-围堵控制理论,但控制平台是基于局部拓扑结构实现的,缺乏多无人机系统的全局信息. Zhao 等人^[20]探讨了无人机通信与控制一体化的通道跟踪、协同通信、自定位、无缝覆盖、轨迹设计、资源配置等技术问题,仅从通信的视角分析了多无人机控制所面临的挑战和机遇. Pham 等人^[21]通过一个基于分布式控制的多无人机集群,可在野火蔓延的情况下协同跟踪动态环境,但系统需要预规划编辑,灵活性一般. Zou 等人^[22]针对无人机编队队形变换问题设计的协同控制方法,其网络通信拓扑是弱连接的,数据交互会出现延时现象. Yu 等人^[23]研究了多无人机在执行器故障和输入饱和条件下的分布式容错控制问题,采用图论和滑模观测器 Sliding Mode Observer(SMO)相结合的方法增强了系统的可靠性,但忽略了无人机间信息共享的实时性等问题.一些无人机的开源控制平台如 QGroundControl^①和 DJI FlightHub^②等,虽然都支持基本的飞行控制功能,但系统启用的飞行模式单一,且不具备通用性. CoUAV 是由 Wu 等人^[24]面向多无人机协作设计的一个通用控制监控平台, CoUAV 基于 Linux 核作为中间件实现与无人机的交互,虽然屏蔽了机组硬件上的差异,但无云架构支撑,通用性不强且服务应用范围有限.综上所述,基于云架构实现多无人机的数据共享和协同控制是扩展无人机应用场景、提升动态环境任务需求的有效方法,但系统集成对网络通信的实时性和稳定性以及控制协同的有效性均提出了新的挑战及要求.

多无人机联合旨在“1+1>2”,实现多无人机协同飞行控制的目标拟解决以下几个关键问题:(1)多无人机协同控制的有效性;(2)多源异构数据的高

① QGroundControl. <http://qgroundcontrol.com/>

② DJI FlightHub. <https://www.dji.com/flighthub>

效存储、处理及信息共享；(3) 通信链路的稳定性及数据响应的实时性。针对上述问题和已有研究存在的不足如数据实时处理能力差、计算资源匮乏、无人机间数据分离、难以实施全局协同调度等，本文提出了面向多无人机协同飞行控制的云系统架构(Multi-UAV Oriented Cooperative Control System Architecture Based on Cloud, UAV3CA)，其原理是通过云服务器连接无人机，并在平台即服务(Platform-as-a-Service, PaaS^[25])层将无人机资源封装并提供公开接口；云控制端根据预执行的任务对无人机进行分组并做有机组合，通过远程过程调用(Remote Procedure Call, RPC)获取无人机及其传感器上的数据，并将数据卸载到云端做高效存储，实现多无人机间数据的交互及共享；云计算节点作为无人机集群的远程 CPU，运用其强大的计算能力对无人机回传的数据进行处理，生成控制命令并将其发送到无人机上完成远程控制，实现对多无人机的联合部署、飞行调度以及通力协作。

2 面向多无人机协同飞行控制的云系统架构

多无人机协同飞行是未来机器人在人工智能领

域发展的必然趋势。通过将多架低成本的无人机组合并部署到云端，可弥补单无人机因体积小、续航时间短等自身条件限制所带来的能力不足及缺陷，使得无人机集群具备单无人机硬件无法超越的计算、存储及处理能力。传统的无人机集群技术难以满足日趋复杂的任务场景需求，如(1)稳定的通信链路以支持数据实时传输，(2)状态数据和感知信息强大的存储、计算能力，(3)全局最优解的决策生成及命令部署，(4)系统控制的协同性、智能性、灵活性等。鉴于此，本文提出一种面向多无人机协同飞行控制的云系统架构 UAV3CA，为无人机协作应用创建一个开发原型系统。

UAV3CA 采用互联网将无人机与云及云控制端连接，实现多无人机之间的超视距连接。基于基础通信链路，无人机将承载的传感器数据和自身的状态信息同步到云端，而云控制端根据目标任务间接驱动无人机协同飞行。作为无人机与控制端的中间桥梁，云端一方面对无人机卸载下来的感知数据进行处理，并公开且共享无人机上的状态信息；另一方面按照控制端的要求进行数据分析，将生成的控制命令经封装后传递给无人机的飞控系统。图 1 所示为 UAV3CA 的系统架构图，下面将详细阐述系统

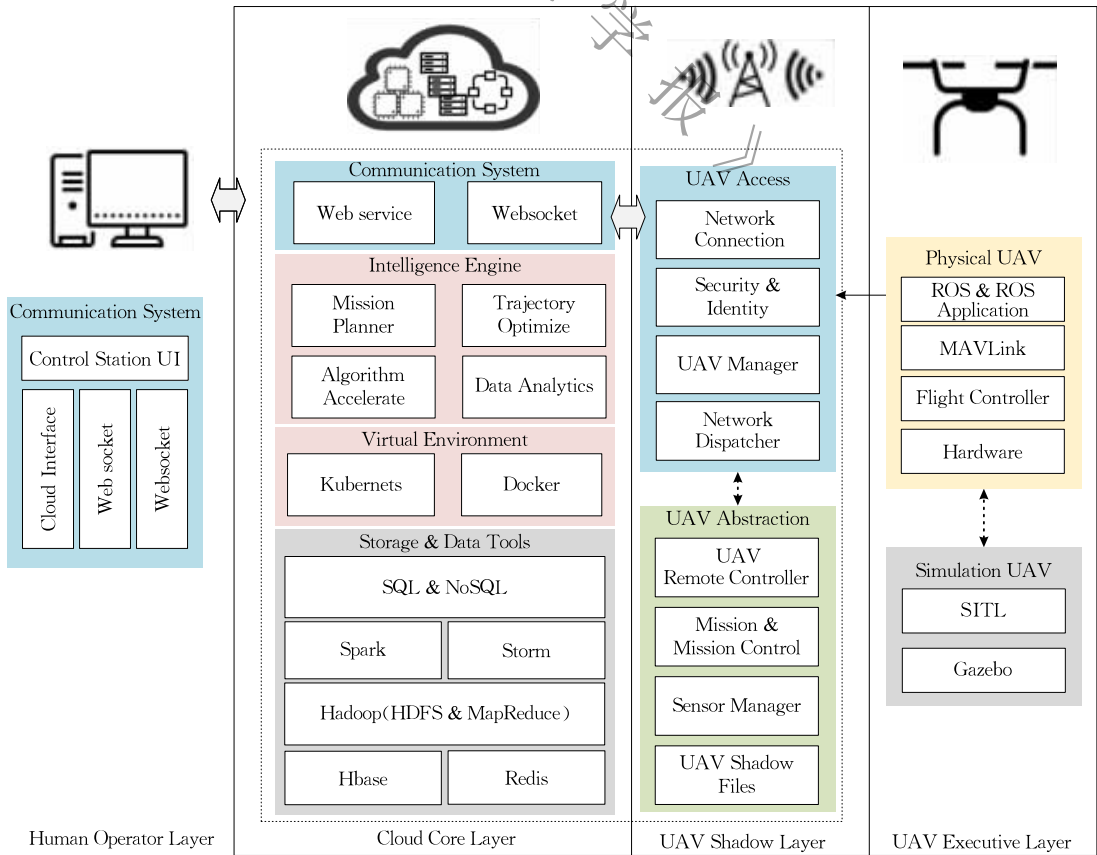


图 1 UAV3CA 的系统架构图

架构的组成部分及软件功能的具体实现。

UAV3CA 系统架构主要由无人机执行层、云端层和操控层组成,针对不同的层级和功能需求,本文采用模块化松散耦合的方式,设计了三个软件功能逻辑节点——无人机节点、云控制节点和协同控制节点。下面进行详细介绍。

2.1 无人机执行层

无人机是系统最终的交互目标,其代表一组资源,可向外部应用提供服务。硬件层上,本文通过开源机器人操作系统 ROS^[26]的一些标准服务如硬件抽象、进程间消息传递等,将无人机资源对外服务化;通过另一种资源抽象方式——微型飞行器轻量级消息编组库 MAVLink^①的不同传输方式如 TCP、UDP、USB 等,使无人机与地面控制站交换预定义消息。为减少因重复性操作造成的资源浪费,降低系统开发的经济成本,缩短开发周期,本文基于 Gazebo 运行模拟环境场景,并通过 PC 机上 SITL^②模拟器模拟仿真无人机。仿真无人机上载有飞行控制器程序 ArduPilot^③,与飞行控制相关的状态信息和环境感知数据可从仿真无人机的飞行动力学模型中模拟收集,未来用于系统框架模型的仿真实验测试。

软件层上,无人机被抽象为一个逻辑节点。首先通过 ROS 的发布/订阅机制和软件包 MAVROS^[26]的解析封装功能,向无人机发送 MAVLink 消息,为开发人员提供高级接口。其次基于 ROSBridge^[27]软件包,开发人员可在非 ROS 系统和 ROS 系统之间进行交互式通信,本文在操控层(非 ROS 系统,参见 2.3 节内容)上采用 Websocket^④协议格式发送数据,并由 ROSBridge 进行转换发送给相关的 ROS 主题;反之,从 ROS 主题获取的无人机上数据经 ROSBridge 转换回传到操控层。另外,无人机节点还包含飞行控制模块和数据感知采集模块两个部分。飞行控制模块用于控制无人机的基本动作,通过解析 ROS 发来的 MAVLink 指令消息包,将生成的预期飞行动作与无人机当前的姿态和位置进行比较,并经飞行控制算法计算出无人机的运动偏移量,制导无人机飞行。数据感知采集模块,用于收集无人机上的数据,其中包括无人机的硬件信息、客户端验证数据信息、无人机状态同步信息以及传感器采集信息等。基于 ROS 的消息发布/订阅机制和 ROSBridge 工具的代理转发,数据感知采集模块可与云控制节点(参见 2.2 节内容)的数据感知接收模块进行数据交互。

2.2 云端层

云端层是 UAV3CA 系统架构设计的重点,是操控层与无人机执行层间连接控制的桥梁,具有无人机远程“CPU”之美誉,是多无人机协同飞行的核心运行层。为了实现各组件间的松散耦合且功能需求可扩展,本设计将云端层细化为两个部分,即无人机影射层和云核心层,另外在软件层上抽象一个逻辑节点——云控制节点。

2.2.1 无人机影射层

无人机影射层旨在将无人机抽象为云端资源,为开发者提供完全独立于无人机具体特征的操作方法。无人机影射层在物理无人机和云之间起桥接作用,一方面将无人机状态及感知数据转移到云端进行处理,另一方面将云端计算、分析的结果发送给无人机作任务调度。无人机影射层包含以下两个基本组件:

(1) 无人机接入组件,亦称为底层通信模块。无人机可选择多种网络协议与控制端交互数据,但为了避免云端部署 ROS 时发生中心节点 Master 过载的宕机现象,本文在云端(非 ROS 端)采用 ROSBridge 与无人机(ROS 端)发送消息,对应于无人机执行层建立 Websocket 协议通信。当无人机申请接入到云服务器时,首先需要经过身份验证。身份一旦验证成功,云服务器将通过线程池与无人机保持连接,且为有效处理不同格式的数据和指令信息,通信模块被设计为多线程服务模式。

UAV3CA 设计是面向多无人机和多控制端服务的,访问请求必然会存在冲突。为避免此类现象发生,本文将在控制端与无人机之间设置一定的匹配权限,即先对无人机进行分组,然后在控制端设置不同的组权限来分层管理无人机信息,实现对无人机资源的有效管理。

假设存在一种情形,即某一时刻有多个用户同时访问一架无人机,由于当前请求频繁且即时并发数据量过大,网络空间通信链路会因带宽资源有限而发生超载。为此,系统设计嵌入了一个网络分发器,将系统多源请求数据进行存储转发,并采取网络代理的方式把 MAVLink 消息和 ROSBridge 消息广播到其它端口。

(2) 无人机抽象组件,是对无人机及其上资源,

① MAVLink. <http://qgroundcontrol.org/mavlink/start>

② SITL simulator (software in the loop)

③ ArduPilot autopilot suite. <http://ardupilot.com/>

④ The Websocket Protocol. <https://tools.ietf.org/html/rfc6455>

包含与无人机相关的信息以及可执行的操作,进行抽象并封装,具体包含:① 远程控制器,封装为可在无人机上执行的所有 MAVLink 命令消息以及和与 ROS 主题相关的无人机动作指令,如起飞、降落、悬停、前进、拍照等;② 任务控制器,以协助无人机自主完成任务,其可以运行单个任务,也可依时间轴运行一系列连续的任务;③ 传感器管理器,用统一的方式集中管理传感器信息,以解决因无人机异质性、传感器多样性而产生的数据不一致性,以及 ROS 中基于不同主题发布消息而产生的交互性冲突;④ 无人机影射文件,通过一个 JSON 格式的文件反映物理世界中无人机的形态。

2.2.2 云核心层

云核心层是 UAV3CA 系统框架的核心,具体包含以下组件:

(1) 数据存储工具组件,为来自无人机的各种数据提供存储服务。无人机系统中需要存储、检索和访问不同类型的数据,且根据数据类型、结构及应用需求的不同采用不同的数据库。如 SQL 数据库存储位置坐标、三维速度、姿态欧拉角、方向、环境温度等结构化数据;NoSQL 数据库存储无人机设备信息、认证信息、图像及视频等非结构化数据;对于及时响应的指令数据,系统采用近实时处理的 Storm 流,而对于实时性要求不高的大容量数据,系统在 Hadoop^[10] 框架上使用 HDFS^[28] 文件系统读取数据,并采用分布式编程模型 Map/Reduce^[11] 对数据进行并行处理。

(2) 虚拟环境组件。虚拟技术已经成为构建服务器集群的重要手段。本文选择 Docker^[29] 和 Kubernetes^[29] 进行集中式服务器的管理和调度,其中 Docker 负责提供虚拟容器运行时环境,Kubernetes 负责管理、编排和调度由 Docker 引擎构建的容器,对服务器集群进行资源整合和调度。

(3) 智能引擎组件。无人机在任务执行的过程中需要智能算法的支持,如任务匹配、路径规划、动态调度等。智能引擎组件基于 Hadoop 集群、采用分布式计算框架 Map/Reduce 实现对大规模数据的并行计算和分析。这种远程“大脑”的控制模式,可支撑多无人机系统智能协同运行。

2.2.3 云控制节点

云控制节点上装有 Ubuntu 操作系统,但不部署 ROS,以规避 Master 的中心节点效应。为实现对多无人机的协同控制,云控制节点设计了多个功能服务模块。

(1) 数据感知接收模块。使用消息发布/订阅机制从数据感知采集模块中订阅与主题相关的数据信息,并依类型的不同对数据作分发处理,如将入网验证信息转发至认证管理模块,状态数据转发至状态同步模块,传感器数据转发至误差检测模块。

(2) 认证管理模块。接收来自数据感知接收模块的无人机 IP 地址以及用于认证的密钥 Key,通过和系统内置约定的安全验证码进行比对匹配来确定请求入网的无人机是否合法。

(3) 状态同步模块。为无人机协同控制提供实时且稳定的无人机状态数据。通过 Timing 机制定期从数据感知接收模块中获取无人机的状态数据,同时对无人机的特定状态进行实时监听,以保持无人机影射文件内容与物理无人机真实状态的同步性。

(4) 误差检测模块。将从数据感知接收模块中获取的传感器数据与预期目标指令进行对比,鉴定无人机是否成功接收到命令、是否成功执行命令且达到预期目标。MAVLink 协议本身没有反馈机制,因此,对于控制端的即时命令,系统每发出一次 DO 命令,误差检测模块就检测一次无人机状态,通过反复发送、反复检测,直至确认无人机执行该 DO 命令为止。而对于过程执行命令,系统不作强精度要求,仅预设误差容限范围,即通过反复计算和调整无人机当前状态和预期目标的差值,使其降至误差容限范围内来表明命令执行成功。

(5) 基于影射文件的命令控制模块。影射文件在认证管理模块认证成功且分组编成后生成,是存储在云服务器上物理无人机的数字孪生,反映物理世界中无人机实时状态(如飞行模式、当前位置等)的 JSON 格式文件,可视为虚拟无人机。影射文件将发送给无人机的命令执行序列按时间戳顺序先缓存再发送。物理无人机和虚拟无人机通过状态同步模块推送同步状态信息,即无人机将实时状态上传至影射文件,控制端通过影射文件将期望的状态指令发送给无人机。基于影射文件的命令控制模块不仅可以克服因网络不稳定性而导致的无人机频繁上线、掉线现象以及控制命令的时间顺序误差,而且其内置的存储转发机制可以减轻由多个控制节点向同一架无人机重复请求访问而引发的网络负载过压问题。

(6) 坐标转换模块。是将一个参考系下的坐标点转换为另一个参考系下的坐标点,以统一系统内部的控制调用接口。

(7) 服务提供模块。是对无人机及其上资源封

装并向外部提供统一风格接口的服务. 其通过面向资源的接口设计提供三种类型的服务: 无人机自身状态数据 UAVInfo, 如方向、速度等; 可提供资源服务 UAVService, 如传感器数据、雷达数据等; 执行动作参数 UAVAction, 如运动轨迹改变参数等.

2.3 操控层

操控层建立在云端层之上, 云核心层给予操控层强大的数据计算与分析能力支撑; 当有任务下达, 操控层生成决策部署, 并通过云端层间接控制多无人机协同飞行. 操控层为无人机飞控程序提供了两种类型的接口: Websocket 和 Webservice, 通过这两种接口, 操控层可以同时访问多个无人机, 或基于云端层提供的服务为无人机设置参数并执行协同飞行算法. Websocket 是一种全双工通信协议, 一方面允许无人机主动向客户端同步信息, 另一方面允许操控层通过云端向无人机推送指令. ROSBridge 包中的 ROSJS 库同样支持 Websocket. 当无人机执行层 MAVLink 消息包进入云端层时, 将首先通过套接字被接收, 然后由网络分发器经 Websocket 发送到操控层. Webservice 接口用于开发实时控制、任务控制、无人机管理以及其他非连续数据流等应用.

协同控制节点是建立在操控层上的可公开调用无人机资源的节点, 基于云控制节点中的功能模块, 即通过统一接口封装屏蔽不同无人机的异质性, 降低传感器数据的异构性, 无人机之间可实现数据的交互与共享. 基于操控层上的 Web^[30] 协议, 协同控制节点通过 HTTP 接口向外部提供无人机的资源和服务, 如通过 HTTP GET 方式获取无人机上的资源; 通过 HTTP POST 方式请求服务; 通过 HTTP PUT 方式更新无人机的 ID 信息; 通过 HTTP DELETE 方式中止无人机执行当前任务或移除任务. 基于资源公开和数据共享, 协同控制节点可面向具体的任务需求, 调用一定的智能算法对在线的多架无人机进行匹配、分组和调度, 使之协同飞行并完成期望目标任务.

综上所述, UAV3CA 设计完成了一个面向多无人机协同飞行控制的云系统框架, 其从通信链路建设开始, 到基础设施和功能组件服务模块的中间件设计, 从底层抽象到面向应用, 最终在控制端实现了对多无人机终端的远程协同飞行控制.

3 系统框架设计的关键技术机制

多无人机协同控制是建立在无人机到端端的可

靠连接之上, 基于连接, 先将无人机上的数据卸载到云端, 通过云核心层对多源异构数据进行处理、存储、计算和分析, 并将生成的指令回传至无人机实施控制. 而设计的核心控制层将来自不同无人机上的数据信息进行交互共享, 参考任务作统一规划、调度, 控制多无人机协同飞行, 共同完成目标. 因此, 面向多无人机协同飞行控制的云系统架构 UAV3CA 提出了三种嵌入式关键技术机制: (1) 基于 ROS 发布订阅的无人机连接机制; (2) 基于 MAVLink 协议的无人机控制机制; (3) 基于 RESTful^[31] 架构的无人机数据共享机制. 下面详细阐述这三种关键机制.

3.1 基于 ROS 发布订阅的无人机连接机制

无人机上装载 ROS^[26], 一方面可重用开源的项目库, 另一方面还可为硬件资源提供软件抽象. 目前, ROS 已广泛应用于机器人的程序开发, 但却缺乏通过互联网来远程控制和监控机器人的原生支持. 因此, 本文在 ROS 发布/订阅的机制上通过引入 ROSBridge 和代理服务器的方法, 提出一种新的无人机连接机制, 以支持 UAV3CA 框架能够快速创建无人机实例, 安全可靠的连接无人机与云端, 对无人机进行硬件级别的认证, 并基于策略的授权, 使控制端对特定主题拥有读写权限, 以保证安全.

3.1.1 基于 ROSBridge 的去中心化方法

ROS 是一个分布式系统, 其将机上每一个应用程序表征为一个节点(Node), 节点之间采用基于主题(Topic)的发布(Publish)和订阅(Subscribe)机制来实现消息传递, 不同的 ROS 程序间通过 RPC (Remote Procedure Call) 方式建立通信并传输数据. 为了简单, 云端可选择部署 ROS, 但分布式 ROS 系统管理是一种基于 Master 主节点的“集中式”管理模式, 如果将云端设置为 Master 主节点, 且运行在 Internet 上具有公共 IP 地址的服务器上, 进而通过基于发布/订阅主题和服务的方式实现无人机和云端以及无人机之间的通信访问来维护网络中所有其他节点的信息, 必将导致整个网络强烈依赖于中心节点 Master, 极大地影响系统的实时性和稳定性, 具体表现为: (1) 当多个无人机同时申请接入网络时, 主节点 Master 会因负载过高而导致崩溃, 影响整个系统的运行并且难以恢复; (2) 当网络系统中存在大量主题和数据时, 本地数据传输会产生延迟, 远程数据传输会因网络带宽和处理能力而受限. 因此, 为保证无人机远程控制的稳定性, 本文设计采用一种基于 ROSBridge 的去中心化方法来实现无

人机连接。

ROSBridge^[27]是扩展功能包,其可在尽量减少对现有开源软件改变的基础上建立 ROS 无人机和非 ROS 云端之间的通信连接. ROSBridge 包含 ROSBridge-Protocol 和 ROSBridge-Implementation 两部分. ROSBridge-Protocol 提供基于 JSON 格式的通信协议. ROSBridge-Implementation 由 ROSAPI、ROSBridge-library、ROSBridge-server 等包组成. ROSBridge-library 是 ROSBridge 的核心功能包,提供的 JSON-ROS 转换,即任何可发送 JSON 的语言或传输方式均可与 ROSBridge-Protocol 协议进行通信并与 ROS 进行交互,且 JSON 采用的“Key-Value”字符存储和表示方式,易于理解,冗余量少,方便转换. 数据处理包 ROSBridge-server 提

供 TCP、UDP、Websocket 三种通信连接方式,本文采用 Websocket 方式与客户端建立连接,Websocket 可实现单个 TCP 连接上的全双工通信协议,使云端服务器既能被动接收无人机状态数据,也能主动推送指令给无人机,比 Ajax 轮询机制节约带宽资源.

3. 1. 2 基于代理服务器的通信模型

虽然 ROSBridge-serve 提供的 Websocket 通信是建立在 TCP 连接上的全双工通信协议,但信息频繁的变化会对系统的实时性要求高;且当数据交互时,网络中的每架无人机都要求具有可被访问的公共 IP 地址,条件限制苛刻. 因此,本文将不采用 ROSBridge-serve 通信接口,而是在具有公共 IP 的无人机上设置一个代理服务器来实现无人机和云端的交互,其通信模型示意图如图 2 所示.

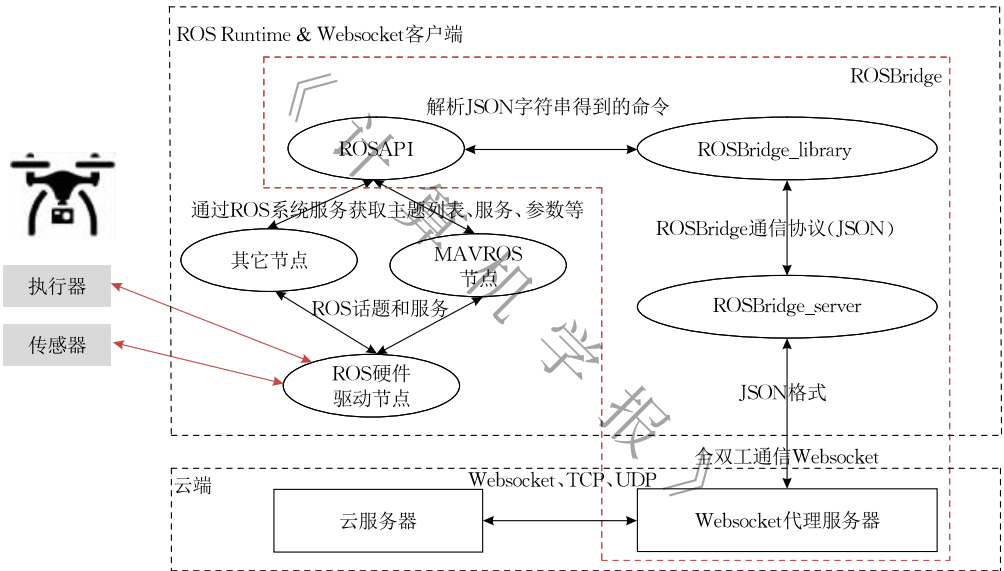


图 2 无人机连接机制的通信模型

通信模型将基于 ROSBridge 的无人机通信连接机制运行在一个代理服务器上,通过与代理服务器连接,无人机上的数据转发和指令接收将不再需要公共的 IP 地址,且与云端的交互均可通过代理服务器间接完成. 具体地,无人机在 ROS 基础上仍然运行 ROSBridge-server 以支持 JSON 格式序列化消息的传输,但不再提供通信服务,而经 ROSBridge-server 序列化产生的 JSON 格式数据将通过 Websocket 代理服务器处理并转发给云控制端. 反之,控制端通过云端提供的服务,将控制指令发送给代理服务器,以 JSON 格式封装的命令经 ROSBridge-server 解析再发给无人机并执行相应操作. 代理服务器的设置,实现了无人机与云端以及控制端的桥接,无人机和控制端向代理服务器发送

消息,代理服务器将相应的数据转发给另一方.

综上,本文提出的基于 ROS 发布订阅的无人机连接机制,通过引入 ROSBridge 和设置代理服务器,使得 UAV3CA 系统框架具有较高的实时性和稳定性,优势具体表现如下: (1) 独立于 ROS Master 主节点,规避了主节点的中心化影响; (2) 代理服务器的设置确保了控制端和无人机的无缝通信; (3) 基于 Websocket 实现的全双工通信节约了带宽资源; (4) Websocket 支持基于 Web 应用的客户端与云端实现同步通信,可实时获取无人机状态; (5) 支持 UAV3CA 系统框架对无人机进行认证授权并对无人机实施有效管理.

3. 2 基于 MAVLink 协议的无人机控制机制

控制是 UAV3CA 系统框架设计的核心内容.

本文在开源飞控程序 ArduPilo 的基础上,选用开源 MAVLink 协议封装无人机操控功能,并提供无人机状态及参数访问接口,使控制端通过云端可远程控制无人机位移并做出预置任务响应.图 3 示意了

一种基于 MAVLink 协议的无人机控制机制,其模块设计可保证消息抵达飞控、命令被正确接收且执行成功,解决网络不稳定导致的命令序列错乱等问题.下面阐述控制机制中各模块的设计细节.

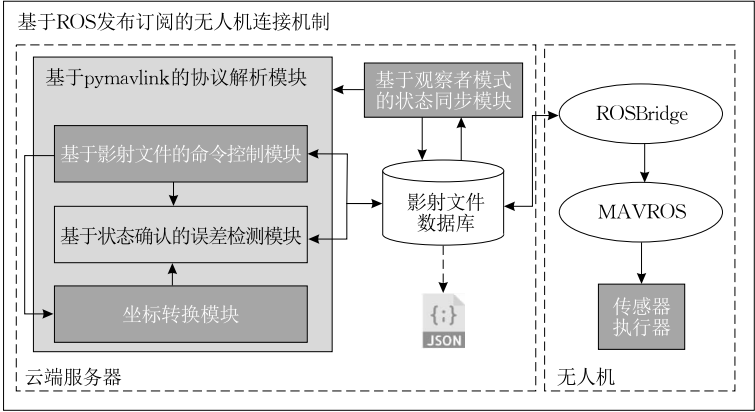


图 3 基于开源 MAVLink 协议的无人机控制机制示意图

基于影射文件的命令控制模块,一方面将 pymavlink 协议解析模块解析出的 MAVLink 指令按照时间戳顺序缓存至影射文件数据库中,另一方面直接将 MAVLink 指令通过 ROSBridge 发送给无人机.影射文件是物理空间的无人机在虚拟空间的影射或化身,是无人机在云端的数字孪生,其通过一个 JSON 格式文档,将无人机上的状态、控制端的期望状态以及命令序列存储于数据库中.每一个无人机对应一个影射文件,并通过状态同步模块读取或写入影射文件,进而同步虚实空间无人机的状态.影射文件的状态同步和指令缓存,给 UAV3CA 命令控制的稳定性带来增益,具体表现为:基于影射文件,(1)控制端可实时无缝地获取无人机状态信息,屏蔽无人机因掉线而无法响应的状态请求;(2)无人机可主动将状态同步到影射文件中,控制端只需请求影射文件即可获得无人机的最新状态,避免了数据重复传输,节约了带宽资源;(3)控制端可将指令按时间戳顺序保存至影射文件,当无人机掉线后重新接入网络,能够及时从影射文件中获取指令,并按时间判断是否执行,有效避免了错误动作.

坐标转换模块检测控制命令是否涉及不同坐标系,并对其做出相应的坐标转换.无人机收到指令后由飞控模块控制执行,并将当前状态上传,以修正状态同步模块中的数据.

MAVLink 通信协议规范了发送和接受微型飞行器数据时的使用规则,但其功能尚不完善,表现如下:(1)消息和消息确认不能保证到达;(2)控制命

令在非安全状态可能会被忽略;(3)命令在完成前可能会被不同的控制源打断.因此,本文采用防御式控制手段,设计了基于状态确认的误差检测模块,其通过不断比对当前状态与预期状态来减小误差,保证指令成功发送给无人机并已安全执行.设计状态确认包含:(1)检查确认无人机是否处于控制命令执行阶段;(2)更改命令前,确保前一命令已执行成功;(3)对状态变化做出反应,若控制端将无人机设置为非 GUIDED 模式,则禁止发送指令;(4)确认命令执行是否在无人机消息传递的正常延迟范围内.基于状态确认的误差检测模块,通过状态同步模块检测指令的执行情况作出相应调整,其控制机制过程的伪代码如算法 1 所描述.

算法 1. 无人机控制机制过程.

输入:与协议对应的控制命令序列(C1,C2,...,Cn)
输出:命令是否正常完成的信号 isSuccess
描述:
1. While (C!=∅&&arguments) do
2. UAVstate synchronize with ShadowFilesDB
3. UAVstate.CommandSequence=C
4. UAVnewstate=pymavlink(parsing,coordinate transformation, Ci) to ROSCon
5. ConfirmCheck (UAVnewstate, synchronize with ShadowFilesDB)
6. GoNext Ci+1
7. End While
8. Return isSuccess

3.3 基于 RESTful 架构的无人机数据共享机制
时空局限性会导致多无人机间无法相互感知,

为了打破无人机间数据信息彼此隔离的状态,本文拟基于云服务器,将无人机信息汇集于云端,并公开云内部数据使得多无人机可以共享数据并感知彼此位置.

Web 开发中,资源可以是服务器上任何一个可命名的抽象概念. 本文所涉及的无人机资源,是将无人机本身抽象为资源,或者将无人机上可获取信息或执行操作的设备抽象为资源. 由于无人机及设备的异质性,其上定义的资源和服务各不相同,因此需在底层抽象封装,在应用层提供统一接口. 如图 4 所示,本文设计了基于 RESTful^[31] 架构的面向 Web 应用的数据共享机制,即将无人机资源抽象为云端的基础设施,基于 PaaS(Platform-as-a-Service,平台即服务)云计算技术将无人机资源封装为云端服务并提供接口,而控制端可通过服务接口直接调用无人机资源. 图 4 中的无人机通过 ROS 发布订阅机制连接到云服务器,每个接入的无人机被认为是具有一组资源或能够提供服务的基础设施. 每个资源用 Web 统一资源标识符 URI(Uniform Resource Identifier)表示. 通过 RESTful 架构,云服务器端可向 Web 客户端提供面向无人机资源的服务接口,而客户端(包括无人机)可通过具有良好标识的统一语义接口与无人机资源进行交互,实现数据共享机制. Web 端采用 Django 技术,并通过 HTTP 协议提供资源交互式操作.

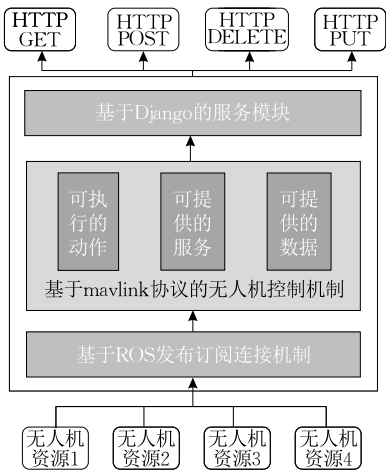


图 4 无人机数据共享机制

多无人机协同飞行以数据共享为前提. 基于 RESTful 架构提供统一的服务接口,无人机资源可被其他客户端访问;且不同无人机上的状态、感知等数据均可通过云端实时共享;云计算节点对收集的数据信息进行综合分析、全局决策,生成面向任务的

协同飞行命令,发送给与目标最优匹配的无人机飞控系统,调度指挥多无人机协同完成任务. 具体以跟踪飞行为例,假设任务区内在航的一架无人机(定义为跟踪无人机)一旦发现目标,将立即向云控制端发送跟踪请求,以申请其他待航的无人机协同飞行跟踪;云端接收到请求后,立即通过网络接入查找在线的可执行跟踪任务的无人机,并根据条件选择与任务匹配最佳的无人机起航. 匹配成功后的无人机(定义为追随无人机)将请求云控制端启动跟踪会话,而云端会将追随控制命令转发给追随无人机、制导其起飞并巡航追随跟踪无人机. 当追随无人机驶入既定轨道,即跟踪无人机被追随,多无人机协同跟踪飞行任务完成.

因此,基于 RESTful 架构的无人机数据共享机制,可实现无人机上资源的抽象和封装并对外提供统一接口. 云控制端通过接口可直接调用无人机资源,一方面实现不同无人机上的数据公开、信息共享,另一方面统一规划、协同操控多无人机完成任务. 这种底层封装技术极大地简化了客户端程序的开发时间及成本,且通过云端 PaaS 资源共享,为多无人机系统的协同调度及飞行控制奠定了基础.

4 模拟实验

4.1 实验环境配置与原型系统实现

4.1.1 实验环境配置

整个 UAV3CA 系统框架的实验环境由三部分构成:开发环境、模拟无人机环境和云服务器环境. 下面介绍各环境的软硬件配置情况.

开发环境构建于惠普 2Plus 笔记本上,其具体配置如表 1 所示. 模拟无人机环境及 DG926 型号四旋翼无人机的具体配置如表 2 所示. 云服务器环境基于 5 台相同型号的 Dell 服务器构建,每台服务器的软硬件配置如表 3 所示.

表 1 开发环境的软硬件配置

名称	详细配置
硬件配置	CPU: Intel Core i7-64700HQ 4 核 8 线程
	主频: 2.6 GHz 内存: 8 GB 主频 2400 MHz
软件配置	操作系统: Ubuntu Kylin 16.04 64 位
	ROS 版本: ROS Kinetic
	开发工具: Eclipse Mars PyCharm
	数据库: MySQL 5.7
	服务器: Tomcat 8
	仿真环境: Gazebo 6.0 语言环境: Java Python

表 2 模拟无人机配置	
名称	详细配置
硬件配置	CPU: Intel Core i7-64700HQ 2 核 4 线程
	主频: 2.6GHz 内存: 1GB 主频 2400MHz
软件配置	操作系统: Ubuntu Kylin 16.04 64 位
	飞控: Ardupilot
	模拟软件: Simulation-In-The-Loop (SITL)
	语言环境: Python 模拟工具: Vmware

表 3 云服务器配置	
名称	详细配置
硬件配置	CPU: Intel E5-1620 3.6GHz
	内存: 32GB 网络: 10/100/100Mbps 无线
软件配置	操作系统: Ubuntu Server 16.04 64 位
	Docker 版本: Docker-ce
	Django 版本: 2.1
	MySQL 版本: 5.7 语言环境: Python

4.1.2 基于 SITL 模拟无人机

本文采用由 Python 开发的适用于不同操作系统的快速工具 DroneKit SITL 来模拟启动无人机,以四旋翼无人机模拟启动为例,SITL 的起始点坐标参数为:

```
dronekit-sitl copter-3.3 --home=  
-35.363261, 149.165230, 584, 353
```

启动后,SITL 在端口 5760 上等待 TCP 连接.由于只有一个连接端口,SITL 无法满足多个客户端同时观察无人机状态的请求,因此使用 MAVProxy 建立网络代理,将模拟无人机发送的 MAVLink 数据包通过 UDP 方式转发到其他非 ROS 设备上的软件终端.下述命令表示 MAVProxy 通过 TCP 连接的 5760 端口接入模拟无人机,并将 MAVLink 数据包转发到 UDP 连接的 14550 和 14551 端口.

```
mavproxy.py --master tcp:127.0.0.1:5760  
-out 127.0.0.1:14550 -out 127.0.0.1:14551
```

SITL 可生成多个实例,且已有的开发脚本解决了 UDP 端口冲突问题.此时,若想使用多个 SITL 实例一次模拟多个无人机,应设置参数 -Instance N 且将 $10 \times N$ 添加到所有端口上.下述命令将在 TCP 连接的 5770 和 5780 端口上启动两个模拟无人机.

```
dronekit-sitl copter -I1  
dronekit-sitl copter -I2
```

本实验将为每个 SITL 实例创建一个虚拟机(VM),并使用虚拟网络和物理网络组合的方式来提供跨 SITL 实例通信.实验基于 Vmware 软件构建虚拟机,其允许多个 SITL 实例在单台机器上和

多台机器上相互通信,若运行 SITL 实例的机器具有无线网络接口,那么模拟无人机能与物理无人机进行通信.另外,在不修改软件的基础上,SITL 还可以模拟网络拓扑效应(如延迟、数据包丢失等).但独立运行一个操作系统内核并模拟实物将会消耗大量的系统资源,极大地限制了单机上 SITL 模拟无人机的数量.为了以最少的资源消耗实现多无人机模拟,本实验设计为每个 SITL 实例分配一个 UDP 端口号.正如图 5 所示,是在单机上基于 SITL 创建多个模拟无人机实例的通信图.

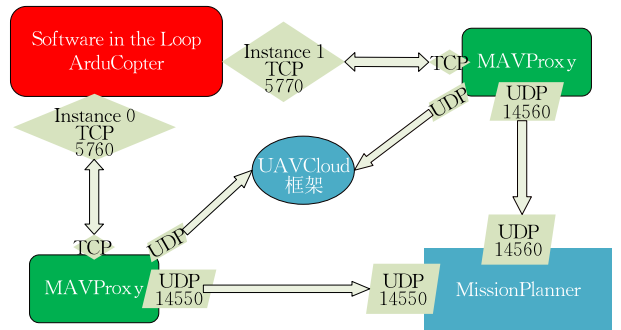


图 5 单机上基于 SITL 创建多模拟无人机实例的通信图

一个 MAVProxy 连接一个 SITL 模拟无人机实例,MAVProxy 从 TCP 接收到的数据流将通过不同的 UDP 端口来分发模拟广播行为.图 5 中两个 SITL 实例分别通过 TCP 5760 和 5770 端口与 MAVProxy 建立连接,而 MAVProxy 分别通过 UDP 14550 和 14560 端口将消息转发到 MissionPlanner,同时 UAV3CA 从 MAVProxy 的其它 UDP 端口接收数据包实现通信.

4.1.3 原型系统实现

UAV3CA 框架原型系统不仅能够实时访问无人机上的信息资源,还能够基于云端智能算法对搜集的数据进行处理和分析,进而对无人机实施飞行控制、任务执行等操作.系统类图如图 6 所示,主类是 Class UAV,其通过 Pymavlink 类实现协议的解析、封装和序列化,通过 Parameter 类获取无人机状态、设置无人机属性,通过注册监听者/观察者接收关于无人机属性更改的异步通知.无人机的 Command 类用于定制 MAVLink 消息,Command-Sequence 类用于操作命令序列,ShadowFile 类提供增删查改接口,进而对无人机实施面向任务的控制. Django-Framework 类用于网络通信,一方面基于 Websocket 交换 JSON 格式的序列化消息,以处理无人机和系统框架之间的数据交互;另一方面基于 Pymavlink 封装的 MAVLink 协议,对外提供可获取无人机资源的 RESTful 风格的 Webservice 接口.

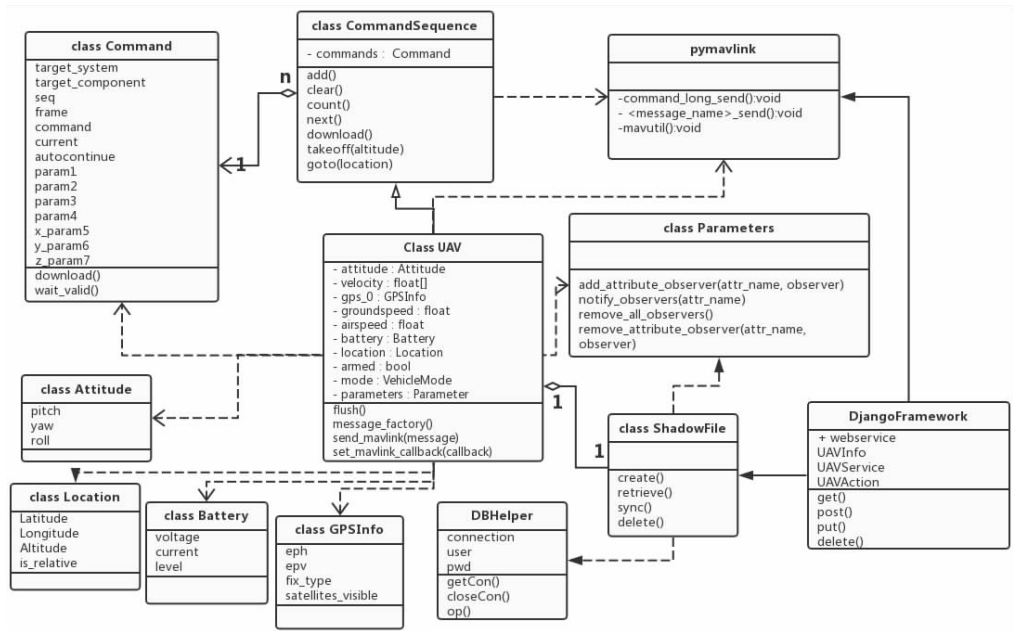


图 6 UAV3CA 框架原型系统的主要类图

控制系统原型基于 Django 框架开发。实验将服务器部署于 Docker 中。服务器启动之后，无人机接入组件将开启服务并实时监听，通过 IP 地址和相应的端口号接收来自外部的连接请求访问。图 7 给出了基于 Web 的控制系统图形界面。地图视图中显示有两架处于运动飞行状态的无人机，具体包含全功能控制面板和 HUD(Head Up Display)。HUD 显示关于无人机的状态信息，如高度、地速、空速、电池、姿态角、GPS 坐标、续航时间等。Web 控制系统使用 JavaScript Websocket 客户端与 UAV3CA 系统框架提供的 Websocket 接口进行连接。当无人机接入系统时，将创建具有唯一颜色标识符的无人机

控制面板部件。控制面板显示修改后的无人机名称，并通过不同标识指示通信链路状态。用户可通过控制面板上的按钮向无人机发送控制命令，对照图 7 左下角 UAV 001 的控制面板，从左到右两行命令依次为：起飞、悬停、准备、拍照、着陆、开始/停止任务、返回、激活/关闭；此外，还有一些其他控制按钮，如更改高度等。所有这些控制命令都是通过云端层提供的 RESTful Webservice 接口基于远程过程调用来执行的。例如起飞控制，Web 控制系统通过 POST 服务发送控制请求 UAVAction/takeoff/{id}/{height}，其中 id 代表无人机的唯一标识符，height 代表起飞高度。

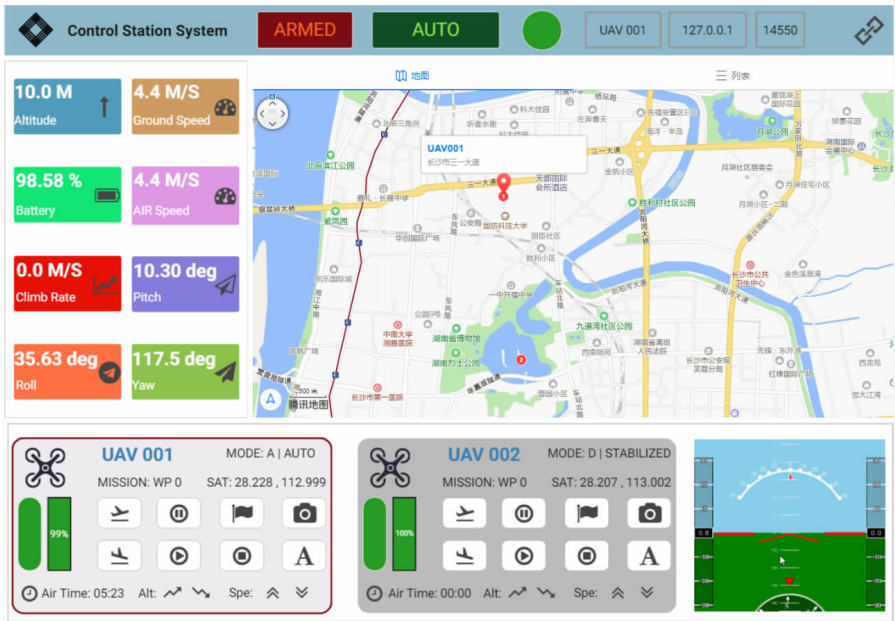


图 7 基于 Web 技术的控制系统原型

4.2 实验结果与分析

为了验证 UAV3CA 系统框架在多无人机协同控制上的有效性,实验将基于无人机模拟技术,首先对单架无人机分别进行基于位置控制、基于速度控制以及面向任务控制的测试,然后基于引领-跟随算法测试多无人机协同飞行控制的能力。

4.2.1 单无人机制导控制

非任务情形下可使用 GUIDED 模式控制无人机。根据目标位置信息的确定与否,UAV3CA 设计了两种制导控制方式,基于位置控制和基于速度控制,下面通过具体实验测试其控制能力。

(1) 基于位置控制

如果目标的位置确定,可通过明确的位置信息导航来控制无人机飞行。UAV3CA 提供的位置控制服务是 UAVAction/simpleGoto/{location}, 还可通过 UAVInfo/setParamater/{xx} 服务设置参数 airspeed 和 groundspeed 来控制无人机的飞行速度。

图 8 呈现的“NUDT”字母航拍轨迹图是基于位置控制实现的无人机飞行轨迹图。图 8(a)记录无人机的初始位置;图 8(b)和图 8(c)均是基于相对坐标系,记录无人机以自身作为参考点的移动轨迹;图 8(d)是在 NED 坐标系下,记录以起始点(HOME)为参考点的无人机飞行轨迹。无人机飞行全程的过程参数设置如下,空速=10 m/s,起飞高度=5 m,绘制“NU”轨迹高度=5 m,绘制“DT”轨迹高度=5 m。

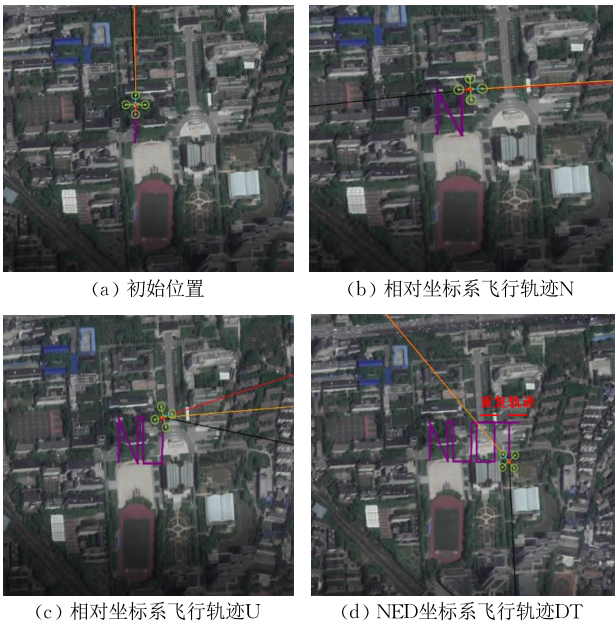


图 8 用位置方式控制无人机形成的“NUDT”轨迹

无人机的状态信息在飞行过程中会发生实时变化,本实验选择三个状态参数——姿态角、经纬度以及高度来检测无人机的飞控状态,并分别绘制了三

个状态在飞行过程中的变化曲线图。

图 9 描绘的是无人机姿态角的变化曲线,其显示偏航角集中在四个区域内变化,对应于无人机飞行轨迹的四个字母。虽然“NUDT”字母的轨迹共有十三个拐点,但图中偏航角的变化明显多于十三次,其原因是目标航迹实现过程中会产生重复路线,比如字母“T”,无人机是先“绘制”横线,然后折返一半航迹再“绘制”竖线。同时,从图 9 中还可以观测到俯仰角和翻滚角均在字母笔画变化较大时产生大的波动。当无人机执行降落命令后,姿态角波动开始变小,并逐渐趋于平稳。图 10 所示是无人机经纬度在飞行过程中的变化曲线。按照既定轨迹“NUDT”飞行,无人机运动轨迹将在南北方向上来回往返,并从西向东逐渐延伸,因此,纬度曲线值在一个区间内来回变化,经度曲线值总体呈增加态势。特别地,经度曲线在 4~5 min 区间内发生两次幅度减少的情形,其原因在于在按字母“DT”轨迹飞行时有两段重复折返,亦可参照图 8(d)中的文字说明。图 11 描绘了无人机整个飞行过程的高度变化,其与初始设置一致;另外,由于是模拟实验,高度曲线值表现稳定。

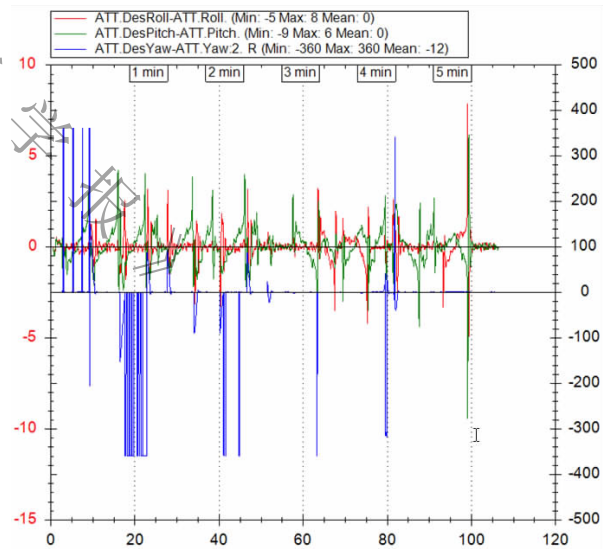


图 9 无人机在“NUDT”轨迹飞行过程的姿态角变化

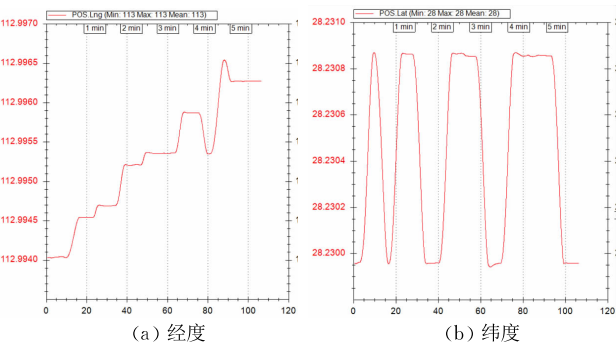


图 10 无人机在“NUDT”轨迹飞行过程的经纬度变化

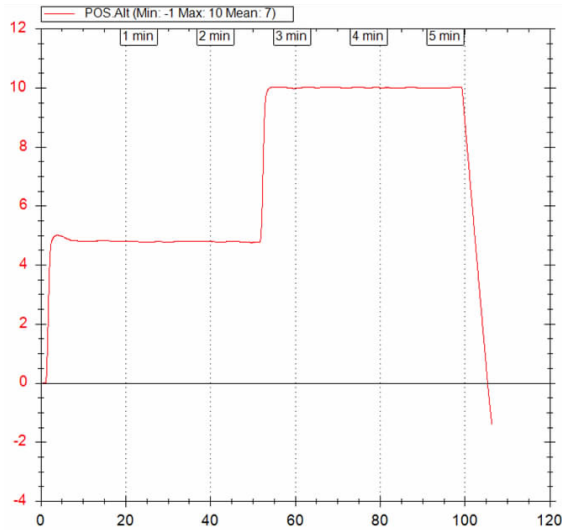


图 11 无人机在“NUDT”轨迹飞行过程的高度变化

(2) 基于速度控制

当无法获取目标位置或者目标位置频繁发生变化时,基于位置的控制方法将不再有效,因此 UAV3CA 设计了基于速度的控制方法,通过相对位移控制无人机动作。

UAV3CA 提供的速度控制服务是 UAVAction/sendVelocity()/ {velocity_x, velocity_y, velocity_z, duration},其基于 NED 坐标系,以 HOME 为参考点向无人机传送速度向量,制导无人机相对位移。发送速度向量是周期性的,因为飞控程序需要持续的速度指令,以避免导航在一段时间内未接收到指令而进入休眠状态。

图 12 给出了一个基于速度控制的无人机飞行轨迹图,无人机呈圆形轨道飞行,持续转弯过渡平滑,这充分说明基于速度的控制方法计算精度高、控制灵活,即通过相对位移制导要优于直接给定位移制导。图 13 给出了无人机在整个圆形轨迹飞行过程中的姿态角、俯仰角以及翻滚角的变化曲线。对比于图 9,虽然无人机的两次飞行轨迹不同,但在两种不同的控制方式下,无人机的俯仰角和翻滚角的总体变化趋势差异不大,因为飞行过程中均在同一高度平稳飞行,不翻转亦不调整高度;但是,基于速度控制飞行的姿态角却在整个区域内发生大幅度持续变化,其主要原因是该方式能使无人机及时感知到位移的变化,并快速做出反应,进而使飞行动作灵活可控。图 14 描绘了无人机的在整个圆形轨迹飞行过程中经纬度的变化曲线。因为是圆形运动轨迹,所以经纬度曲线呈周期性振荡,且三个周期说明无人机画圆周三次。高度不发生变化(图略)。

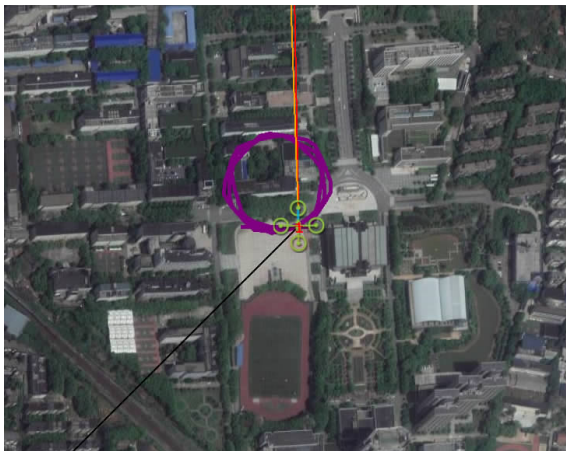


图 12 用速度方式控制无人机形成的轨迹

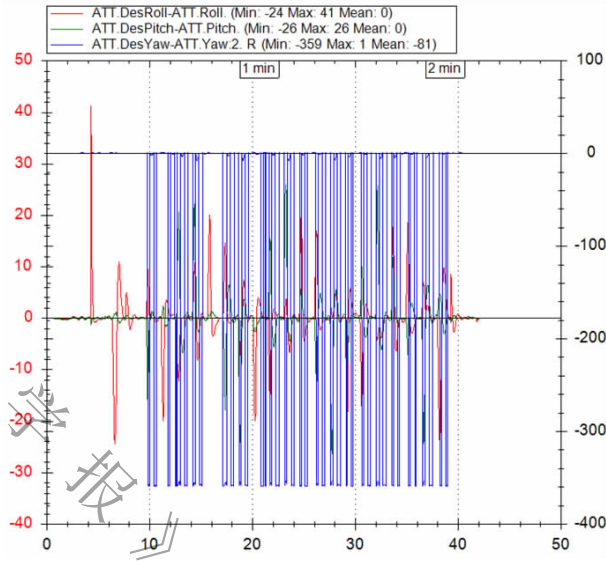


图 13 无人机在速度控制过程中姿态角变化

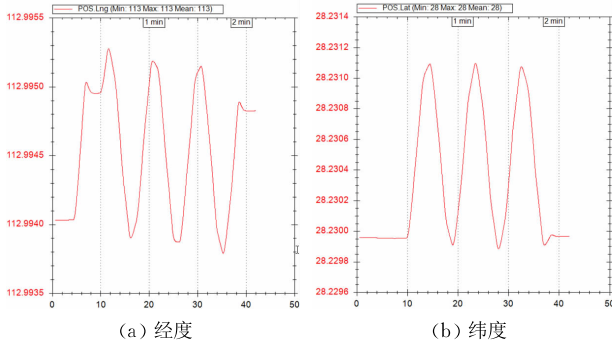


图 14 无人机在圆形轨迹飞行过程的经纬度变化

(3) 面向任务控制

当指派一个任务时,无人机的控制方式将发生变化,首先需要模式切换,其次需要影射文件缓存并分析。面向任务控制是在给定任务的前提下,对无人机飞行路径依据任务规划而实施动态控制的方法。对于预先设定的飞行抵达目标点位,UAV3CA 并不

直接启动无人机,而是先将这一系列的目标点位缓存至影射文件,再根据任务中的假设条件和预期目标分析目标点位,对其进行基本操作如增减目标点位、计算目标点位数量、读取最宜逼近的目标点位并执行当前任务等,亦或基于一定的算法模块和智能驱动规划高级任务设定,执行与之对应的高级飞行控制命令,如无人机逼近目标点位的时序规划和

最优路径的计算等.当规划好的目标点位从影射文件上传至无人机时,无人机的导航模式被转换为 AUTO,飞控程序将依缓存规划输出的控制指令开始执行任务.表 4 是本实验设计用到的任务控制序列,其中第 1 列是依次抵达的目标点位序号,第 4 列是 MAVLink 控制指令的 MSG-ID 号,后续列是经纬度等一系列参数值.

表 4 影射文件缓存的任务控制序列

QGC WPL 110										
0	1	0	16	0.0	0.0	0.0	0.0	28.229958	112.994034	0.000000 1
1	0	0	16	0.0	0.0	0.0	0.0	28.22895510	112.99400930	10.000000 1
2	0	0	16	0.0	0.0	0.0	0.0	28.22785860	112.99395560	10.000000 1
3	0	0	16	0.0	0.0	0.0	0.0	28.22769790	112.99438480	10.000000 1
4	0	0	16	0.0	0.0	0.0	0.0	28.22781140	112.99487830	10.000000 1
5	0	0	16	0.0	0.0	0.0	0.0	28.22838800	112.99491050	10.000000 1
6	0	0	16	0.0	0.0	0.0	0.0	28.22895510	112.99493190	10.000000 1
7	0	0	16	0.0	0.0	0.0	0.0	28.22919860	112.99496140	10.000000 1
8	0	0	16	0.0	0.0	0.0	0.0	28.22921040	112.99546570	10.000000 1
9	0	0	16	0.0	0.0	0.0	0.0	28.22922450	112.99607460	10.000000 1
10	0	0	16	0.0	0.0	0.0	0.0	28.23026670	112.99603160	10.000000 1
11	0	0	16	0.0	0.0	0.0	0.0	28.23060940	112.99550320	10.000000 1
12	0	0	16	0.0	0.0	0.0	0.0	28.23249520	112.99535840	10.000000 1
13	0	0	21	0.0	0.0	0.0	0.0	28.23310250	112.99533690	10.000000 1

图 15(a)是上述任务控制序列对应的可视化地图,图中可观测到共包含 14 个飞行预抵达的目标点位,按顺序依次进行路径规划,从起飞(点 1)到降落(点 14)有序可控.面向任务控制是调用 UAV3CA 提供的任务控制服务 UAVAction/uploadMission/{missionFile},通过 POST 请求将表 4 所示的任务文件上传至云端,经影射文件缓存处理后,再通过 UAV3CA 的无人机模式转换服务 UAVInfo/setMode/AUTO 和任务启动服务 UAVAction/missionStart,将无人机的导航模式设置为自动并开始执行任务.无人机起飞后的面向任务控制的航迹路线如图 15(b)所示.对比图 15(a),无人机起飞的飞行轨迹与任务规划中的目标点位顺序完全一致,且动作按指令控制依次执行,最后在 14 号目标点位执行降落命令并结束任务.图 16 是无人机飞行过程中经纬度的变化曲线,与整个任务控制规划的路线完全一致.

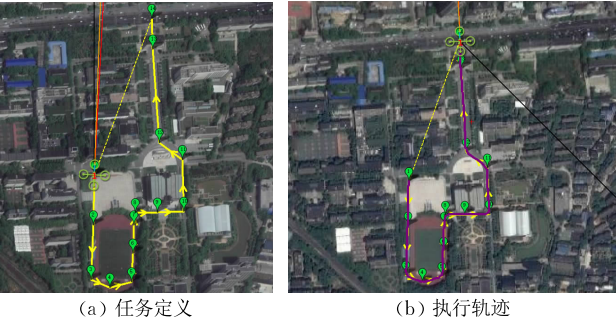


图 15 无人机任务式控制的轨迹图

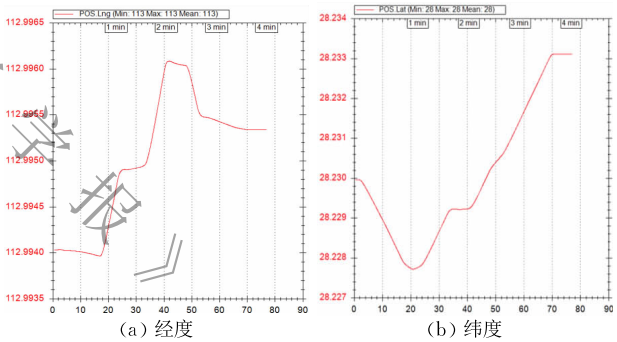


图 16 无人机在任务控制飞行过程的经纬度变化

综上,对比无人机的三种飞行控制方式,基于位置控制的方式最为简单,因为目标点位直接给定且固定不变,控制方式简单且无需大量的计算支撑;基于速度控制的方式比基于位置控制方式的精度和灵活性高,但相对位移的计算需要云端资源支持,以赋予控制方式能力的提升;面向任务控制的方式相对复杂,因为多样化的任务内容不如位置和速度内容单一,因此需要基于一定的分析和处理才能赋予无人机具体的控制命令.因此,UAV3CA 设计了基于影射文件的缓存处理机制,并通过云端计算资源增强无人机的远程控制能力.

4.2.2 多无人机协同控制

实验采用引领-跟随算法测试 UAV3CA 对多无人机协同飞行控制的能力效果,其过程伪代码如算法 2 所描述,通过 RESTful 框架服务接口获取无

人机上资源,实现无人机间数据信息的实时共享,并基于云服务器在多无人机中选定一架为引领者 Leader,其余均为跟随者 Follower. Leader 的路径需要在云端层进行规划,根据任务情况选择不同的方式制导;Follower 可将 Leader 的坐标位置作为参考点,依任务需求选择不同的跟随方式(即 Leader-Follower 不同的编队方式),且与 Leader 的相对位置和追随方向亦可选择不同.不同的协同控制方式均可基于 UAV3CA 实现.本实验是通过对 Leader 的具体飞行轨迹和路线进行规划,进而基于 Leader 的相对位置计算完成 Follower 的飞行航迹来实现多无人机协同飞行控制过程测试的.表 5 所示内容为基于 SITL 模拟启动五架无人机实例的脚本,具体包含每架模拟无人机的初始设置,其创建过程可参照上一小节实验部分.基于实际飞行场景并结合模拟控制端,五架无人机协同飞行启动后的位置变化及其整体飞行过程如图 17 所示,下面具体观测并分析每张图的协同飞行效果.

算法 2. 基于引领-跟随的多无人机协同控制过程.

输入: 引领无人机与跟随无人机的 REST 服务接口
 $wsL, ws1, \dots, wsI$
输出: 协同飞行轨迹
描述:

1. $LeaderUAV = getService(wsL)$
2. $FollowUAV = getService(wsi)$
3. $SetLeader(LeaderUAV)$
4. Place the $FollowUAV$ position
5. $RelativePosition =$
 $CalculateRelativePosition(LeaderUAV, FollowUAV)$
6. All UAV Takeoff
7. $LeaderUAV.goto(position)$
8. $FollowUAV.goto(RelativePosition)$
9. Until $LeaderUAV.stop$

表 5 基于 SITL 模拟 5 个无人机的启动脚本

<pre>#!/bin/bash dronekit-sitl copter -I0 --home=28.2331048,112.9952806,0,0 & dronekit-sitl copter -I1 --home=28.2330859,112.9948944,0,0 & dronekit-sitl copter -I2 --home=28.2331379,112.9957634,0,0 & dronekit-sitl copter -I3 --home=28.2330812,112.9943794,0,0 & dronekit-sitl copter -I4 --home=28.2331899,112.9962355,0,0 &</pre>

本实验任务规划五架无人机起飞时,以“一”字队形等间距逐一排开. UAV3CA 首先基于 ROS 发布订阅的连接机制在多无人机与云服务器间建立远程通信及数据控制链路,并基于 MAVLink 协议的控制机制实时获取在线可控区域内无人机的状

态、感知数据,根据任务要求确立 Leader 位于中间.图 17(b)中地图显示了无人机的初始队形以及它们之间的初始相对位置,图 17(a)是五架无人机在模拟控制端的对应图上队形以及相对位置关系,队形中心圆图标代表 Leader,中心两侧圆图标代表 Follower.控制起飞后,无人机群队形将会依控制端要求随时间推移不断调整变换,且按照一定的路径规划协同飞行.当控制端要求无人机群以竖直方式调整编队飞行时(如图 17(c)所示),UAV3CA 基于 RESTful 架构的数据共享机制实时获取五架无人机的状态数据信息;基于 MAVLink 协议的控制机制在云端层进行数据公开和共享,重新规划 Leader 路径并相对调整 Follower 的飞行航迹;生成的队形变换控制指令发送到无人机节点并执行,图 17(d)完美地呈现出控制端模拟预期设定的五架无人机编队队形,协同一致、精准无误.随着控制端任务需求的进一步转变,编队队形再次变换成“一”字(如图 17(e)),与垂直队形变换任务执行过程类似,五架无人机在地图上的队形如图 17(f)所示,完美协同呈现“一”字.随后,飞行控制随场景环境变化(如需避障等)而变化,如图 17(g)所示,Leader 因飞行途径中遇阻需以偏航角 90° 飞入另一场景上空,Follower 紧随其后变换队形.绕过障碍后,无人机群需恢复原队形,即 Leader 航向再次变换 90° ,Follower 紧随其后呈“一”字排开,如图 17(h)所示.上述两次队形变换中,无论避障还是队形恢复,对集群系统的实时反应和协同控制均要求较高. UAV3CA 框架嵌入的三种创新机制有效地支撑了上述任务需求,包括无人机上数据信息的实时感知和共享(障碍物的位置信息以及无人集群内部的绝对位置和相对位置信息)、无缝可靠的远程连接(通信和数据链路的建立)以及基于云服务器的任务分析和控制命令驱动(队形变换协同控制),过程实现可靠、精准无误,且编队队形模式多样.图 17(j)展示的是按照如图 17(i)控制端的预期目标规划,五架无人机以三角队形飞过下一场景上空.之后控制端进一步要求五架无人机以 Leader 为中心领航,Follower 在 Leader 四个对角呈方形编队跟随飞行,如图 17(k)所示;图 17(l)在地图上精准地展现了这一协同飞行过程.上述协同飞行过程中,通过打印控制端命令发出的时间和 Leader 接收到控制指令的时间,测出控制指令平均延时为 15.2ms ,这依赖于仿真测试环境下网络传输距离短且无外界干扰的情形.

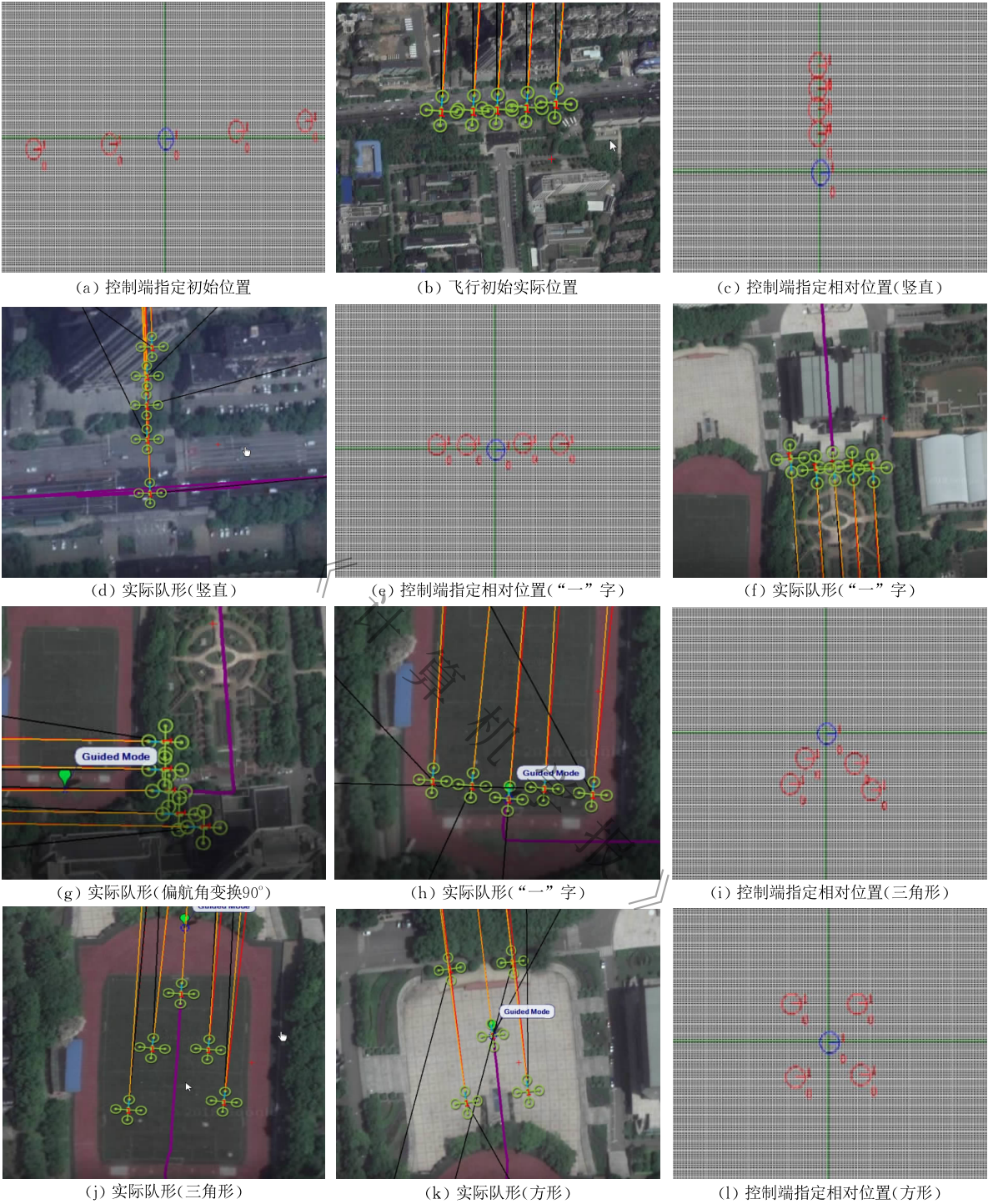


图 17 五架无人机协同飞行过程示意图

通过图 17 中所示的几张地图展示的五架无人机飞行队形与控制端指定模拟编队队形的完美一致性,验证了本文设计的 UAV3CA 对于多无人机协同飞行控制的精准性和有效性,该系统框架在单架无人机的多种可选控制方式上,通过基于云的多无人机资源的抽象封装、数据汇聚、融合、共享、处理和计算实现了对多无人机灵活多变的协同飞行控制,反映在整个飞行过程中数据更新及实时共享(体现

在及时避障和队形恢复上),同时多架无人机由云服务器协同控制并按预期目标任务规划的航迹协同飞行(体现在队形的完美呈现上)。

5 结论与展望

本文提出了一种面向多无人机协同飞行控制的云系统架构 UAV3CA。该系统架构将无人机与云端

相结合,通过对无人机及其上资源封装并对外提供服务接口,使得云及云控制端可远程调用无人机,实时采集无人机上数据并对外公开,在无人机信息资源共享的基础上,基于云端强大的存储和计算能力实现了多无人机的协同飞行控制,提升了多无人机协同完成目标任务的灵活性和智能性. 实验结果从单无人机制导控制到多无人机协同控制两个层面给予充分验证,最终以完美的编队队形展现了 UAV3CA 对多无人机协同控制的有效性. UAV3CA 设计不仅达到了预期目标,而且在控制方式上可进一步依任务需求的不同而变换,即通过不同的参数选择及设置实现不同的协同控制方式,具有一定的可扩展性. 下一步,我们将进一步开展实物实验及其相关验证工作,在实际场景及实测 UAV 集群中进一步验证 UAV3CA 系统框架的功能. 同时,为了适应不断复杂化的应用场景,我们可将任务需求不断深化,进一步推动和丰富云端智能算法模块的设计,以任务为牵引丰富系统的智能应变能力,增强多无人机优选优配、通力合作、共同完成复杂环境下目标任务的协同能力,为 AI 空中机器人的应用扩展推波助澜.

参 考 文 献

- [1] Zhang Tao, Li Qing, Zhang Chang-Shui, et al. Current trends in the development of intelligent unmanned autonomous systems. *Unmanned Systems Technology*, 2018, 1(1): 19-30 (in Chinese)
(张涛, 李清, 张长水等. 智能无人自主系统的发展趋势. 无人系统技术, 2018, 1(1): 19-30)
- [2] Jin Wei, Ge Hong-Li, Du Hua-Qiang, Xu Xiao-Jun. A review on unmanned aerial vehicle remote sensing and its application. *Remote Sensing Information*, 2009, 24(1): 88-92(in Chinese)
(金伟, 葛宏立, 杜华强, 徐小军. 无人机遥感发展与应用概况. 遥感信息, 2009, 24(1): 88-92)
- [3] Humboldt W, Losonsky M. *On Language: On the Diversity of Human Language Construction and Its Influence on the Mental Development of the Human Species*. Cambridge University Press, 2000
- [4] Atzori L, Iera A, Morabito G. The Internet of Things: A survey. *Computer Networks*, 2010, 54(15): 2787-2805
- [5] Inaba M. Remote-brained humanoid project. *Advanced Robotics*, 1996, 11(6): 605-620
- [6] Goldberg K, Siegwart R. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2001
- [7] Han K H, Kim S, Kim Y J, et al. Internet control architecture for internet-based personal robot. *Autonomous Robots*, 2001, 10(2): 135-147
- [8] Kuffner J. Cloud-enabled humanoid robots//*Proceedings of the 2010 IEEE International Conference on Humanoid Robots*. Nashville, USA, 2010: 53-61
- [9] Arumugam R, Enti V R, Liu B, et al. DAVinCi: A cloud computing framework for service robots//*Proceedings of the 2010 IEEE International Conference on Robotics and Automation*. Alaska, USA, 2010: 3084-3089
- [10] Shvachko K, Kuang H, Radia S, et al. The Hadoop distributed file system//*Proceedings of the IEEE Symposium on MASS Storage Systems and Technologies*. Incline Village, USA, 2010: 1-10
- [11] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters//*Proceedings of the 2004 Symposium on Operating Systems Design and Implementation*. San Francisco, USA, 2004: 137-150
- [12] Hunziker D, Gajamohan M, Waibel M, et al. Rapyuta: The RoboEarth cloud engine//*Proceedings of the IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany, 2013: 438-444
- [13] Kuriki Y, Namerikawa T. Consensus-based cooperative formation control with collision avoidance for a multi-UAV system//*Proceedings of the 2014 American Control Conference*. Portland, USA, 2014: 2077-2082
- [14] Gharibi M, Boutaba R, Waslander S L. Internet of Drones. *IEEE Access*, 2016, 4(2016): 1148-1162
- [15] Koubba A, Qureshi B, Sriti M F, et al. A service-oriented cloud-based management system for the Internet-of-drones//*Proceedings of the 2017 IEEE International Conference on Autonomous Robot Systems and Competitions*. Coimbra, Portugal, 2017: 329-335
- [16] Lin C E, Li C R, Lai Y H. UAS cloud surveillance system//*Proceedings of the 2012 IEEE International Conference on Parallel Processing Workshops*. Pittsburgh, USA, 2012: 173-178
- [17] Sanchez-Lopez J L, Fernández R A S, Bavl H, et al. Aerostack: An architecture and open-source software framework for aerial robotics//*Proceedings of the 2016 IEEE International Conference on Unmanned Aircraft Systems*. Arlington, USA, 2016: 332-341
- [18] Sampedro C, Bavl H, Sanchez-Lopez J L, et al. A flexible and dynamic mission planning architecture for UAV swarm coordination//*Proceedings of the 2016 IEEE International Conference on Unmanned Aircraft Systems*. Arlington, USA, 2016: 201-115
- [19] Dong X W, Hua Y Z, et al. Theory and experiment on formation-containment control of multiple multi-rotor unmanned aerial vehicle systems. *IEEE Transactions on Automation Science and Engineering*, 2019, 1(16): 229-240

- [20] Zhao J, Gao F, et al. Channel tracking with flight control system for UAV mmWave MIMO communications. *IEEE Communications Letters*, 2018, 22(6): 1224-1227
- [21] Pham H, La H, et al. A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking //Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver, Canada, 2017: 6648-6653
- [22] Zou Y, Zhou Z, et al. Distributed formation control for multiple vertical takeoff and landing UAVs with switching topologies. *IEEE Transactions on Mechatronics*, 2018, 23 (4): 1750-1761
- [23] Yu Z, Qu Y, et al. Fault-tolerant containment control of multiple unmanned aerial vehicles based on distributed sliding-mode observer. *Journal of Intelligent & Robotic Systems*, 2019, 93(1-2): 163-177
- [24] Wu W, Huang Z, Shan F, et al. CoUAV: A cooperative UAV fleet control and monitoring platform. *arXiv preprint arXiv: 1904.04046*, 2019
- [25] Mell P M, Grance T. The NIST Definition of Cloud Computing. National Institute of Standards & Technology, 2011
- [26] Quigley M, Conley K, Gerkey B P, et al. ROS: An open-source robot operating system//Proceedings of the 2009 ICRA Workshop on Open Source Software. Kobe, Japan, 2009, 3(3.2): 5
- [27] Crick C, Jay G, Osentoski S, et al. Rosbridge: ROS for Non-ROS Users. Robotics Research. Springer, Cham, 2017: 493-504
- [28] Borthakur D. HDFS architecture guide. Hadoop Apache Project, 2008, 53(1-13): 2
- [29] Bernstein D. Containers and cloud: From LXC to docker to kubernetes. *IEEE Cloud Computing*, 2014, 1(3): 81-84
- [30] Anis K. ROS as a service: Web services for robot operating system. *Journal of Software Engineering in Robotics*, 2015, 6(1): 1-14
- [31] Rodriguez A. RESTful Web services: The basics. IBM Developer Works, 2008, 33(2008): 18-18



SHI Dian-Xi, Ph.D., researcher. His current research interests include artificial intelligence, distributed computing, cloud computing and big data processing etc.

HONG Chen, M.S. His current research interests include artificial intelligence, distributed computing, cloud computing etc.

KANG Ying, Ph.D., assistant researcher. Her current

research interests include artificial intelligence, data mining, community detection etc.

JING Song-Chang, Ph.D., assistant researcher. His current research interests include artificial intelligence, data mining, community detection etc.

ZHANG Yong-Jun, Ph.D., researcher. His current research interests include high performance computing, distributed application system etc.

YANG Dong, B.S. His current research interests include wired communication, information security protection etc.

Background

Unmanned aerial vehicle (UAV) is the most important branch of the development of unmanned system in AI. As a result of the advantages of low cost, all weather and zero casualty, UAV has been applied in many fields like border patrol, agricultural plant protection, real-time road-traffic surveillance, package delivery etc. Especially facing such special tasks in the complicated or dangerous environments as remote perception, public facility inspection, disaster site reconnaissance and rescue etc., UAV is able to replace human to carry out some tough tasks with risk. However, the mostly limitation in size and weight makes UAV difficult to enlarge its payload capacity, so as to the restricted storage and computation. Meanwhile, the real-time variability and

complexity of task and scenario pose more and more challenges for the application of UAV. In order to compensate the capacity deficiency of singleton UAV, multi-UAVs cooperation appeals to plenty of attentions, which is said to be a suitable approach and an inevitable trend.

Until now, there are fewer researches on the cooperative control system of multi-UAVs. At present, UAV is either manipulated by the control units within a vision-distance territory or controlled by the remote operators on the ground. Some works attempted to improve the robotic ability by offloading computation to cloud and build network connection based on Internet. In consequence, the concept of IOD (Internet of Drone like IOT) appears, several techniques like

Hadoop Cluster and Map/Reduce distributed computing architecture are introduced at the same time. Nonetheless, the existing related works have remained in the conceptual research, and newly proposed methods have merely achieved the simple summation of functions. In addition, the stability of network connection, data separation, control lack of unified schedule etc. are the core problems to settle all the time.

This work constructs a hardware control system architecture and synchronously realizes the corresponding software functions. By embedding three innovative mechanisms, a seamless connectivity is established between any of multi-UAVs and cloud. Base on the PaaS(Platform as a Service) which is derived from the cloud computing service, the resources on UAV or UAV as a whole resource is abstracted and encapsulated as interfaces, serving for any terminal call. By means of the formidable storage and computing ability of cloud, the

perceived data and information from UAV can be offloaded to cloud and shared among multi-UAVs in real time. Furthermore, the unified dispatching and deploying which are oriented to task, implement the cooperative flight control of multi-UAVs. During our final work, experimental results from the singleton UAV guidance and multi-UAVs cooperation demonstrate the stability of networks communication, the real timing of data transmission and the validity of cooperative control, exhibiting the flexibility and diversity of multi-UAVs' cooperative control modes in terms of the perfect formation.

We are fortunate to be involved in the project “Unmanaged Cluster System Software Architecture and Operational Support Platform”. This work was supported by the National Key Research and Development Program of China under Grant No.2017YFB1001901, and the National Science Foundation of China under Grant No. 91648204.

