Mission-Oriented Miniature Fixed-Wing UAV Swarms: A Multilayered and Distributed Architecture

Zhihong Liu, Xiangke Wang, *Senior Member, IEEE*, Lincheng Shen, Shulong Zhao, Yirui Cong, *Member, IEEE*, Jie Li, Dong Yin, Shengde Jia, and Xiaojia Xiang

***Abstract*—In this article, a multilayered and distributed architecture for mission-oriented miniature fixed-wing UAV swarms is presented. Based on the concept of modularity, the proposed architecture divides the overall system into five layers: 1) low-level control layer; 2) high-level control layer; 3) coordination layer; 4) communication layer; and 5) human interaction layer, and many modules that can be viewed as black boxes with interfaces of inputs and outputs. In this way, not only the complexity of developing a large system can be reduced but also the versatility of supporting diversified missions can be ensured. Furthermore, the proposed architecture is fully distributed that each UAV performs the decision-making procedure autonomously so as to achieve better scalability. Moreover, different kinds of aerial platforms can be feasibly extended by using the control allocation matrices and the integrated hardware box. A prototype swarm system based on the proposed architecture is built and the proposed architecture is evaluated through field experiments with a scale of 21 fixed-wing UAVs. Particularly, to the best of our knowledge, this article is the first work which successfully demonstrates formation flight, target recognition, and tracking missions within an integrated architecture for fixed-wing UAV swarms through field experiments.**

***Index Terms*—Architecture, fixed wing, swarms, unmanned aerial vehicles (UAVs).**

I. Introduction

A. Motivations and Related Work

DUE TO the advantages in flexibility, cost, and environmental adaptability, unmanned aerial vehicles (UAVs) have created tremendous application potential and have been increasingly investigated in recent years. In particular, UAVs are widely used in the areas, such as reconnaissance, surveillance, plant protection, and disaster rescue. However, with the advance in coordination technology, the limitations of using a single UAV to operate missions become more and more apparent. UAV swarms, consequently, have attracted much attention. Through coordination between members, UAV swarms can share the resources of the whole system and can work as a team cooperatively. In this way, UAV swarms can be more competent for large complex missions.

In order to increase the level of autonomy for UAV swarms, a large amount of studies have been proposed in the area of UAV swarming over the past few years. Some proposals focus on the flocking control [1] and the formation control [2], [3]. Some proposals study the mission planning [4], [5] and the target tracking [6]. Some other work concentrates on higher-level topics, including the distributed sensing [7], the collaborative navigation [8], [9], communications [10], etc. However, few studies are revealed in the perspective of the architecture which plays an important role in the system design and implementation.

In particular, Sanchez-Lopez *et al.* [11] proposed an open-source architecture named by AeroStack for multi-UAV systems. This architecture follows a hybrid reactive/deliberative paradigm and includes five layers, i.e., reactive, executive, deliberative, reflective, and social layers. Whereas AeroStack deploys the time-critical control (e.g., attitude control and actuator control) on a nonreal-time system, which may fail to satisfy with the real-time requirements for high-speed UAVs. Grabe *et al.* [12] proposed Telekyb, an end-to-end control framework for controlling heterogeneous UAVs. Although it allows coordination control of multiple UAVs, its scalability is limited. This is because in Telekyb, the high-level control (e.g., mission planning) operates on the ground rather than onboard. Boskovic *et al.* [13] proposed CoMPACT, a six-layered hierarchical architecture for controlling swarms of UAVs. The main advantage of CoMPACT is that it effectively combines top-level mission planning and decision making with dynamic reassignment, reactive motion planning, and emergent biologically inspired swarm behaviors. Nevertheless, CoMPACT splits the mission execution to levels of mission, function, team, platoon, and UAV, and each level requires a manager that cooperates with other UAVs in the corresponding level. This may increase the burden of the task management. Tahir *et al.* [14] proposed a survey-type study for swarms of UAVs, where the characteristics, such as mechanics, functionality, and applications are discussed. Note that these aforementioned works are

Manuscript received January 14, 2019; revised June 11, 2020; accepted October 11, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61906209 and Grant 61973309. This article was recommended by Associate Editor G. Pandey *(Corresponding author: Xiangke Wang.)*

The authors are with the College of Intelligence and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: zhliu@nudt.edu.cn; xkwang@nudt.edu.cn; lcshen@nudt.edu.cn; jaymaths@nudt.edu.cn; congyirui11@nudt.edu.cn; leonlee2009@nudt.edu.cn; yindong@nudt.edu.cn; jia.shde@nudt.edu.cn; xjxiang@nudt.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2020.3033935

evaluated by simulations or experiments for quadrotors, and no field experiments for fixed-wing UAVs are demonstrated. Comparatively, Chung *et al.* [15] proposed a swarm system and demonstrate live-fly field experiments with up to 50 fixed-wing UAVs. However, this work mainly focuses on the system design for UAV flocking, including the autonomous launch, flight, and landing. The collective behaviors and mission coordination are not included in this swarm system.

Although this field of research has brought important contributions, there are mainly two remaining challenges.

1) *Scalability:* Most of the related work is evaluated by experiments of small scales (i.e., two to five). It is known that with the scale increases, system designs are more challenging both theoretically and practically. A scalable architecture that can support a large scale of UAVs is needed.
2) *Versatility:* Existing solutions mainly focus on specified problems or applications. Few to achieve an integrated framework for general purposes. However, high degrees of autonomy for UAV swarms require the ability to support multiple and heterogeneous applications (e.g., flocking, target recognition, and tracking). Therefore, an architecture that integrates diversified functions and missions is desired.

B. Contributions

In this article, we present a multilayered and distributed architecture for mission-oriented fixed-wing UAV swarms. Compared to existing architectures, there are three main contributions.

1) The proposed architecture is built on the concept of modularity and divides the overall swarm system to multiple layers and many modules. It allows each module to focus on its own design and abstracts away the details of other modules. This not only can reduce the difficulty of developing a large system but also can ensure the versatility of supporting diversified missions.
2) The proposed architecture is fully distributed and each UAV performs the decision-making procedure autonomously. By this means, it removes the dependence of the central controller for mission coordination and brings better scalability to UAV swarms.
3) The proposed architecture is not restricted to specified kinds of aerial platforms. Through introducing control allocation matrices and the platform-independent integrated hardware box, different kinds of aerial platforms can be feasibly extended to the swarm system.

Through field experiments with a scale of 21 fixed-wing UAVs, we evaluate the scalability and versatility of the proposed architecture. Several coordination missions, such as formation flights, target recognition, and tracking are demonstrated. Particularly, to the best of our knowledge, this article is the first work to successfully demonstrate formation flight, target recognition, and tracking missions within an integrated architecture for fixed-wing UAV swarms through field experiments. Besides, the experimental results also show that the prototype system based on the proposed architecture outperforms the state-of-the-art work.

C. Article Organization

The remainder of this article is structured as follows. Section II gives an overview of the proposed architecture. The design of the low-level control layer is presented in Section III. Section IV describes the high-level control layer. The communication layer and the human interaction layer are elaborated in Sections VI and VII, respectively. Section VIII provides the results of the field experiments. Section IX concludes this article and indicates future research directions.

II. System Architecture

The full system architecture is outlined in Fig. 1. It mainly consists of five layers: 1) low-level control layer; 2) high-level control layer; 3) coordination layer; 4) communication layer; and 5) human interaction layer.

The low-level control layer is in charge of the flight control (e.g., speed, height, and attitude) and the control allocation. The high-level control layer is responsible of the computation-intensive tasks, such as visual perception, task planning, and guidance control. The tasks performed in this layer abide by observe–orient–decide–act (OODA) procedure. The coordination layer encapsulates the functions in terms of the negotiation (e.g., task allocation) among UAVs for cooperative missions. Through the coordination layer, each UAV can negotiate with other UAVs to obtain free-conflict solutions. The communication layer manages the message transmission among all the UAVs and the ground control systems. It includes the design of the communication infrastructure (from the perspective of hardware) and the communication management (in terms of software). The human interaction layer is deployed on the ground and provides interfaces for visualizing the situation, including the UAV status, the sensed data, and the geographical environment. It also offers interfaces for operators to command the UAV swarm system.

Through slicing the swarm system into five layers with specified functionalities, this architecture reduces the complexity of developing a large system. Moreover, the proposed architecture divides the overall system into many modules that can be viewed as black boxes with interfaces of inputs and outputs. In this way, each module focuses on its own design and abstracts away the details of other modules, which facilitates the extension of different sensor devices and algorithms.

The proposed architecture is fully distributed and brings better scalability. Each UAV performs the decision-making procedure autonomously. In this way, it removes the dependence of central controller for mission coordination. Moreover, this architecture divides the UAV swarm to individual coordination groups according to the mission requirements and the communication availability, which can hold the scale of the states maintained on each UAV for making decision as the number of UAVs increases. Therefore, the scalability of the swarm system can be significantly improved.

In order to satisfy the timing and computing requirements for the controlling of the swarm system in different levels,

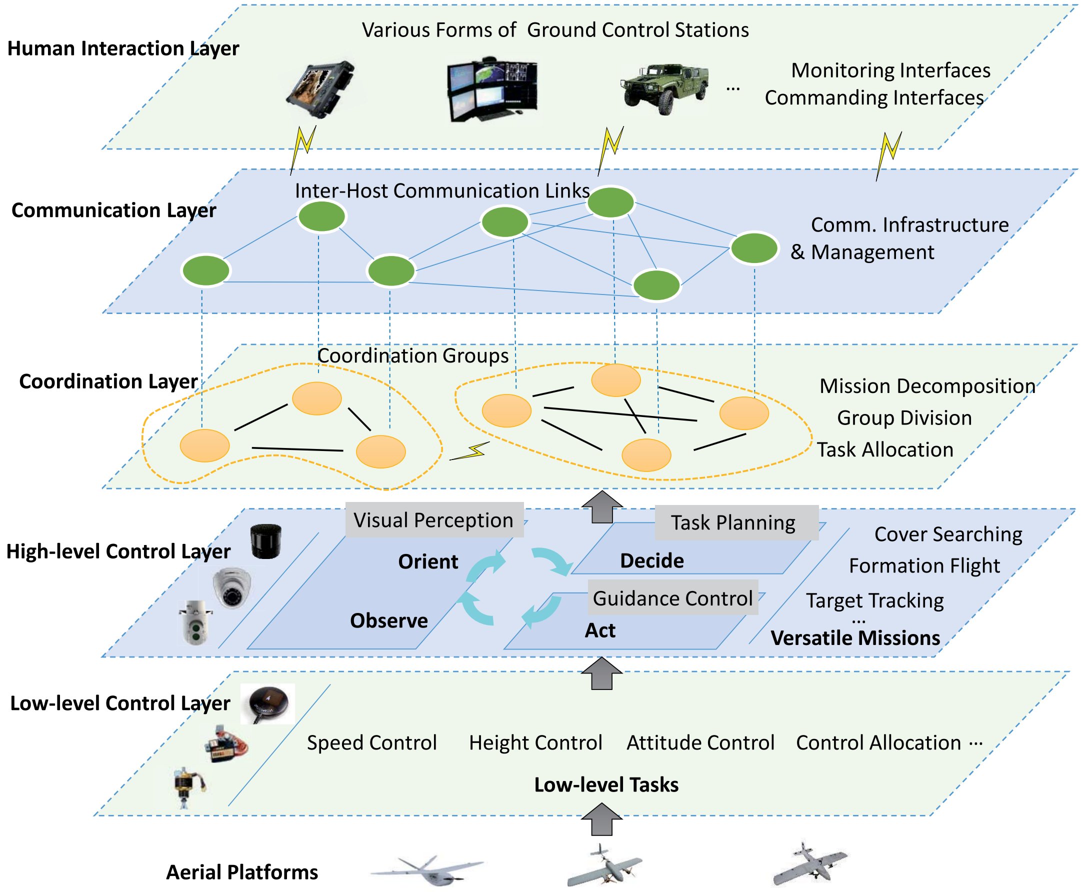


Fig. 1. System architecture.

the proposed architecture leverages two kinds of processing boards. One uses a low-power microcontroller unit and is installed with embedded real-time operating system. In this processing board, the tasks with the strong real-time requirement can be deployed, which is respect to the low-level control layer. The other uses the high-performance microprocessing unit and is installed with a time-sharing operating system. In this processing board, the computation-intensive tasks can be deployed, which is corresponding to the high-level control and the coordination layers. Therefore, this design of the two-level processing not only compensates insufficient computing capacities for the real-time platform but also brings more flexibility for implementing the high-level algorithms.

Note that the proposed system architecture is not restricted to specific kinds of aerial platforms. It is true that different platforms may have different configurations, such as payloads, propulsion mechanisms, shapes, and weights. It is also known that the same flight control signals produce different actuator control outputs for aerial platforms with different configurations. By introducing the control allocation matrices to differentiate the aerial platforms, the swarm system can convert the low-level control signals to compatible actuator control outputs according to the configurations of platforms dynamically. Moreover, for the purpose of designing a lightweight and miniaturized system, the proposed architecture integrates the onboard hardware (e.g., the processing boards, perceptional devices, communication payloads, circuitry, and cooling devices) into a compact box. This box is loosely coupled with the aerial platform. As a result, different kinds of aerial platforms can be feasibly extended to our swarm systems by installing the integrated hardware box. Based on this system architecture, we have accomplished flight experimentations of a swarm with hybrid aerial platforms, including fixed-wing and tilt-rotor aircrafts. In the following sections, we will provide the details of each component of the proposed system architecture.

III. Low-Level Control Layer

The low-level control layer is in charge of the flight control for the UAVs in swarms, which provides each UAV with the ability of accurate flight and adaptation to the complex environments. In this layer, onboard sensors, such as accelerometers, magnetic compasses, and gyroscope, are attached and able to provide the current position and attitude information of the UAV in a timely manner. And this layer accepts the command references of the upper layer and converts to the desired attitude. After obtaining the appropriate pulse-width modulation (PWM) output according to the attitude instruction and the current state of the UAV, the control signal is transmitted to actuators (aileron, elevator, rudder, and throttle).

The fixed-wing UAV is commonly regarded as a six-degree-of-freedom (DOF) rigid body, and it is well known that the dynamic characteristics and control principle of fixed-wing UAVs are quite different from those of quadrotors and helicopters [16]. For example, the fixed-wing UAVs are required to maintain a minimum airspeed to produce enough lift force, resulting in lacking hovering capability. Furthermore, the dynamical model of a fixed-wing UAV is characterized by air-operated complexity, manipulative coupling, and controllable underactuation. It is hard to establish the accurate dynamical of the miniatured model for cost reasons. In addition, cross-coupling dynamic characteristics are generally demonstrated for fixed-wing UAVs, which makes their flight performance vulnerable to both external disturbances and inner effects [17]. Overall, the accurate flight control scheme of the low-cost miniature fixed-wing UAV is of importance and very challenging. Hence, the low-level control layer plays a pivotal role in supporting the whole UAV swarm system.

In our work, there are mainly three aspects in the low-level control layer: 1) speed and height control; 2) attitude control; and 3) control allocation. The accuracy of the speed control ensures the coordination of the UAV in space. Attitude (pitch and roll) control is the most critical part of the flight controller. Its control frequency is usually several times that of the upper layer, and its performance directly affects the safety and stability of the vehicle.

1) Speed and Height Control: Fixed-wing aircrafts rely on wings to generate lift, so the forward flight speed of an aircraft is primarily related to its ability to drive

$$\frac{dV}{dT} = \frac{T - D}{m} - g \sin\gamma \tag{1}$$

where T indicates the thrust of the engine, and D expresses the resistance. From the perspective of capacity conservation, there is a coupling relationship between aircraft speed and height

$$E_T = E_D + E_S = \frac{1}{2}mV^2 + mgh \tag{2}$$

where V and h represent the speed and height of the vehicle, respectively. When controlling the speed of the aircraft, it is necessary to consider both the thrust and the pitch angle of the aircraft. Due to the coupling relationship between the speed and height of the fixed-wing UAV, simply adjusting the drive of the vehicle cannot fully control the speed. Here, we use a fuzzy controller. Tables I and II show the fuzzy inference logic for the controller. More specifically, in order to ensure the precision of the formations for UAV swarms and prevent the UAVs from stalling, we design the fuzzy inference logic I as shown in Table I, where the airspeed and the error of the groundspeed are taken as the inputs and desired thrust signal is taken as the output. We can see from the table that when the airspeed is low, no matter how large the error of the groundspeed is, the desired thrust signal is increased to its maximum. When the airspeed is suitable, the desired thrust signal is adjusted according to the error of the groundspeed.

TABLE I
FUZZY INFERENCE LOGIC OF SPEED CONTROL

EGS AS	≤ -5	(-5, -3]	(-3, -1]	(-1, 0]	(0, 1]	(1, 3]	>3
≤ 8	PB	PB	PB	PB	PB	PB	PB
(8, 9]	PB	PB	PM	PM	PS	PS	ZO
(9, 10]	PB	PM	PM	PS	PS	ZO	ZO
(10, 11]	PM	PM	PS	PS	ZO	ZO	NS
(11, 12]	PM	PS	PS	ZO	ZO	NS	NS
(12, 13]	PS	PS	ZO	ZO	NS	NS	NM
>13	PS	ZO	ZO	NS	NS	NM	NM

AS: Airspeed (m/s); EGS: Error of groundspeed(m/s)
NB: Negative Big; NM: Negative Middle; NS: Negative Small;
PB: Positive Big; PM: Positive Middle; PS: Positive Small; ZO: Zero

TABLE II
FUZZY INFERENCE LOGIC OF HEIGHT CONTROL

EGS α_h	≤ -5	(-5, -3]	(-3, -1]	(-1, 0]	(0, 1]	(1, 3]	>3
≤ -1	PB	PB	PM	PM	PM	PS	ZO
(-1, -0.6]	PB	PM	PM	PS	PS	ZO	NS
(-0.6, -0.2]	PM	PM	PS	PS	ZO	ZO	NS
(-0.2, 0.2]	PS	PS	ZO	ZO	NS	NS	NM
(0.2, 0.6]	PS	ZO	ZO	NS	NS	NM	NM
(0.6, 1]	ZO	ZO	NS	NS	NM	NM	NB
>1	NS	NS	NM	NM	NB	NB	NB

Besides, for ensuring the height of the UAV as stable as possible, we provide the fuzzy inference logic II as shown in Table II, where the error of the ground speed and the normalized height are used as the inputs and the desired pitch angle is used as the output. The normalized height is denoted by $\alpha_h = [(h - h_{\text{desired}})/(h_{\max} - h_{\min})]$, where h, h_{desired}, $h_{\max}$, and $h_{\min}$ represent the current, desired, predefined maximum, and minimum value of the height for the UAV, respectively. Note that the output of the fuzzy controller is transformed to $[-1, 1]$ and divided into seven parts on average.

2) Attitude Control: The most important thing to consider when the aircraft can stably fly is the balance between the lateral stability surface and the vertical stability surface. The requirements of roll angle and pitch angle control are fast, stable, and easy to implement. Many methods can effectively achieve attitude stability such as [18]. In general, the adjustment of the course angle of a fixed-wing aircraft can be seen as a circular turn

$$\dot{\chi} = \frac{g}{V_g} \tan\phi_c \tag{3}$$

where χ represents the heading angle and ϕ_c is the desired control input of roll angle. Adjusting the nose of the body mainly rely on the roll angle, that is, the change of the roll angle brings changes in the course angle. There are many methods used for heading control [19], and what they have in common is their easiness to implement and robustness to disturbance. Here, we adopt a segmented angular rate PI controller to satisfy with the high control frequency (200 Hz) requirement. The control law is shown as follows:

$$u_r = k_p\big(k_P(\phi_d - \phi) - \dot{\phi}\big) + k_i \sum\big(k_P(\phi_d - \phi) - \dot{\phi}\big) \tag{4}$$

where u_r represents the virtual control output of roll angle. ϕ_d and ϕ indicate the desired and actual roll angle, respectively. k_p and k_i are control parameters.

3) Control Allocation: The same flight control signal produces different actuator control outputs for vehicles of different configurations. Due to the different aerodynamic layout and configuration, such as conventional configurations, delta wing, flying wing, double tail, v-tail, etc., the vehicle's actuator outputs are completely inconsistent. In order to be able to adapt to more vehicle platforms, we introduce a concept of the control allocation matrix and ensure that different configurations correspond to different allocation matrices [20].

The design of the allocation matrix is based on parameters, such as the size, weight, and actuator performance of the platforms. With the allocation matrix, we can convert the virtual control signals of the low-level control (e.g., raw, pitch, yaw, and thrust) into compatible control outputs of the actuators for different platforms. The control allocation can be described by

$$\boldsymbol{u}_{\text{actuator}} = A_a^v \boldsymbol{u}_{\text{virtual}} \tag{5}$$

where $\boldsymbol{u}_{\text{actuator}}$ and $\boldsymbol{u}_{\text{virtual}}$ represent the control output of the actuators and the virtual control signals of the low-level control, respectively. A_a^v is the allocation matrix. It should be noted that the dimension of the allocation matrix is closely related to the number of actuators and the dimension of virtual control signals. For instance, the allocation matrix of the classical fixed-wing UAV is given as

$$A_a^v = \begin{bmatrix} \theta_a^r & 0 & \theta_r^r & 0 \\ 0 & \theta_e^p & 0 & \theta_t^p \\ \theta_a^y & 0 & \theta_r^y & 0 \\ 0 & \theta_e^v & 0 & \theta_t^p \end{bmatrix} \tag{6}$$

where θ_a^x, θ_e^x, θ_r^x, and θ_t^x indicate the weights of the aileron, the elevator, the rudder, and the throttle related to the virtual control signals x (i.e.,"r"—the roll angle, "y"—the yaw angle, "p"—the pitch angle, and "v"—the speed), respectively.

IV. High-Level Control Layer

The high-level control layer concentrates on the tasks, such as visual perception, mission planning, guidance control, etc. It follows the OODA loop for realizing swarm autonomously. More specifically, an electro-optical device is attached to this layer and the visual perceptional processing module, which provides the information related to the targets and obstacles, is included in the layer. This represents the observe and orient procedures. In addition, the mission planning module is deployed for producing task plans that can accomplish the user demanding missions. This stands for the decide procedure. Besides, the guidance control that intends to guide the UAVs to reach the desired point in coordination with other UAVs is included, which is implied for the act procedure. The high-level control layer is on top of the low-level control layer but under the coordination layer. It leverages the upper layer to negotiate with other UAVs and produces the guidance control commanding references to the lower layer.

A. Visual Perception

Perception for UAVs aims at gaining awareness of their current state and the environment through onboard sensors. Due to higher requirements of the update rate, the awareness of its current state (e.g., attitude, velocity, and airspeed) is deployed on the low-level control layer, where the sensors, such as IMU, gyroscope, and compass are attached. This module is considered to attach the vision and range devices and is responsible of high-level perceptional processing, including target recognition, target localization, obstacle detection, and situation awareness. For instance, we adapt a monocular camera with a pan-tilt as the vision device in our prototype swarm system.

Target recognition has been studied for years and plenties of proven solutions have been proposed. Recently, this technique has been widely used in unmanned systems [21], e.g., life search in disaster rescue, criminal chase in the urban area, etc. Identifying the interested objects from the incoming image or video data in real time, and thereby providing detection results to other decision module timely becomes an essential procedure of accomplishing the missions. In particular, we use the existing deep-learning-based method named YOLOv3 [22] for detecting the targets and obtaining the corresponding coordinates in the image plane.

Target localization is trying to localize ground-based objects based on image data from UAVs. By using the target recognition results, the pixel location of the target in an image can be obtained. Hence, according to the pixel location, the UAV's attitude and position, and the camera angles, the target localization in world coordinates can be estimated. Besides, in order to improve the accuracy of estimating the status of targets (e.g., positions and velocities), a cooperative target status estimation algorithm using multiple UAVs can be included [7].

Obstacle detection and avoidance for UAVs is an anticipated requirement for autonomous flights, especially in low-altitude maneuvers surrounding with trees, buildings, and other structures. In general, range sensors (e.g., LiDAR, infrared, and ultrasonic) are exploited to measure the distance between the UAV and obstacles [23]. Based on this, the safety area around the UAV can be computed. Other than this category of approaches using active sensors, the vision-based obstacle detection approaches perceive obstacles through passive sensors such as cameras. However, intensive-computation operations, such as image feature tracking and 3-D world information constructing are needed in these approaches.

Note that the perception module is not restricted to the tasks introduced above. Thanks to the modularity inhered from ROS, other perception tasks, such as situation understanding and simultaneous localization and mapping (SLAM) algorithms can also be easily included.

B. Task Planning

The task planning module produces task plans that satisfies the requirements for the commanding missions while also abiding by constrains, such as UAV's payload, endurance, and airspace regulation. This is critical for maximizing the capabilities of UAV swarms as well as the quality of the mission completion. Generally speaking, there are two perspectives of mission planning for UAV swarms. The first is to allocate tasks[1] among multiple UAVs and schedules the tasks for each UAV in a proper sequence. The second is to generate a detailed

[1] A mission can be decomposed into multiple tasks.

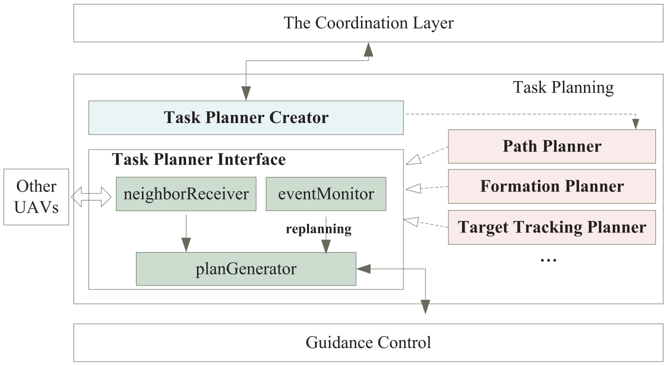


Fig. 2. Task planning architecture.

plan for each task in the sequence. In the proposed architecture, the first perspective is deployed in the coordination layer (see Section V), and the task planning module is in charge of the later perspective.

In order to improve the scalability of the task planning module, we apply the factory design pattern [24]. In this way, planners that have different purposes can be extended feasibly by following the unified interface. The architecture of the task planning scheme is shown in Fig. 2. More specifically, it consists of three types of components: 1) the *task planner interface*; 2) the *task planner factory*; and 3) different kinds of planners. The *task planner interface* declares the methods that each instance of the planners need to implement. The *neighborReceiver* receives the state of the neighbor UAVs and the *planGenerator* develops a coordination plan based on the state of neighbor UAVs. Besides, the *eventMonitor* monitors the event and status at runtime and replan dynamically once some predefined conditions are triggered. The instances of the planners are the actual planners which provide the implementation to the methods declared in the *task planner interface*. The *task planner factory* constructs the actual planners according to the requests from the coordination layer. In our proposed architecture, we include path planner, formation planner, and target tracking planner described as follows.

1) *Path planner* generates collision-free routes that can be satisfied with the task requirements (e.g., destination, threat avoidance, and path length) while taking into account the geometric and physical constraints. In order to generate routes for each cooperative UAV in the swarm, this planner gathers the status of neighbor UAVs and calculates the routes in a distributed manner. Besides, replanning is triggered automatically while emergencies are detected. Here, we adopt a decentralized model predictive control (DMPC)-based path planning method. The whole path planning problem is decomposed into subproblems by rolling time windows (20 s) and each UAV calculates the suboptimal route distributedly in each rolling window. Note that path planning problems have been actively studied and many solutions can be included here.
2) *Formation planner* configures the formation patterns of a UAV swarm according to mission requirement and geographical conditions, e.g., straight line, triangle pattern, and victory pattern (as shown in Fig. 8). After that, it is also responsible for generating the trajectory for each leader UAV and the relative location to its leader UAV for each follower UAV. Here, we use a time-collaboration-based trajectories planning method for leader UAVs that makes each UAV arrive at the corresponding waypoint simultaneously. The relative locations for follower UAVs are predefined with the formation patterns. Besides, the formations can also be dynamically reconfigured as commanded, e.g., while crossing a narrow valley, the formation pattern of the UAV swarm can be changed to a straight line.
3) *Target tracking planner* is in charge of generating routes for arriving at and covering the target area. That is because the area of the moving target is provided with the mission, rather than a precise location. UAV swarms need to search the target in the corresponding area. It is noted that tracking targets in the mission planning level cannot handle the high uncertainty of the target's moving state. In our design, this task is implemented in the guidance control level (see Section IV-C), which is also consistence with the state-of-the-art solution [25], [26].

C. Guidance Control

The guidance control module intends to guide the hosted UAV to reach the desired points or follow the command references produced by the task planning module. More specifically, it uses the UAV status from the low-level control layer and other UAVs as feedback, and produces the control command references, such as desired yaw, speed, and height for the hosted UAV.

Similar to the task planning module, we apply the factory pattern for extending different kinds of guidance control algorithms, e.g., path following, formation control, and target tracking. We can select corresponding kinds of guidance control algorithms according to the task plan produced by the task planning module. Besides, with respect to the same kinds of guidance control, our proposed architecture can extend different control laws by implementing different algorithms and adjust them dynamically according to the mission requirements. Also, even for the same guidance control algorithms, control parameters can be adjusted by configuration or user commands. The guidance control algorithms included in our architecture are as follows.

Path Following: After completing the path planning, each UAV will need to follow the path accordingly. No matter it is coordinated or singular path following, the core goal of each vehicle is to follow the desired path so that the tracking error remains within an acceptable range. One of the most basic capabilities of path following is to effectively resist wind disturbances and keep the aircraft on the desired path.

Due to the coupling of the height and speed of the fixed-wing aircraft, the adjustment of the position is mainly achieved by its speed and heading. For ease of analysis, the problem of path following is usually decoupled into two-dimensional path following and height control. Here, we use a vector field-based approach published in our previous work [27]. This approach designs a vector field around the desired path to determine the orientation of the aircraft head according to the distance from the current position of the aircraft to the desired path. Admittedly, other common path following control methods, such as PLOS [28] and NLGL [29], can also be adopted here.

Formation Control: The formation control intends to control a group of UAVs flying in formation cooperatively, where the formation pattern should be preserved during maneuvers, such as heading change and speed change. Over the past decades, many formation control approaches have been proposed, such as consensus-based approach [30], leader–follower approach [31], behavior-based approach [32], and virtual structure approach [33]. In our design, we adopt a hybrid formation control approaches. For the leader UAVs, we use coordinated path-following control; with respect to the follower UAVs, we use leader–follower coordinated control.

By using the coordinated path following techniques, the guidance control module receives paths that are parameterized according to the desired formation for the leader UAV. These paths are formed by the waypoints requiring the leader UAVs to simultaneously reach. Thus, a control law, which ensures the path following error converges to zero and the consensus of desire speed is achieved, is adopted. As a result, a desired intervehicle formation for leader UAVs can be achieved.

While the leader UAVs fly as the mission required, the follower UAVs follow the corresponding leader UAV and try to form the formation. More specifically, each follower UAV first generates an induced route according to the desired distance between its current position and its desired position. After that, each follower UAV will be guided by the path-following guidance law with velocity adaption.

Target Tracking: The guidance control for target tracking is responsible for guiding the UAV to fly around the targets so that the targets remain in the UAV's detection range. More specifically, once the target is detected by the visual-perceptional module, the UAV flies a circular orbit around the targets and keeps a constant distance with the targets. We adopt a guidance control method based on the Lyapunov vector field for target tracking in our previous work [26]. With respect to single-vehicle tracking, this method controls the heading of the UAV based on a Lyapunov vector field so as to guide the UAV to fly a circular orbit around the target with a specified radius. In terms of the multiple-vehicle tracking, this method builds an additional Lyapunov vector field for controlling both the desired speed and heading of the UAV in order to ensure the intervehicle angular spacing, thereby preventing these UAVs from collision and achieving multiple viewing angles for surveillance.

Note that the obstacle and collision avoidance functionality is also included in this module. More specifically, the condition of performing obstacle and collision avoidance actions can be triggered anytime during the mission execution. Once the condition is triggered, the obstacle and collision-avoidance algorithm produces guidance command references for the host UAV in order to avoid the detected obstacle and collision with other UAVs. This has higher priority than the mission executions. While safety condition is satisfied, UAVs continue the previous mission. Here, we adopt a hierarchical collision-avoidance method in our work [34]. This method partitions the conflict detection region into three levels according to the distance between the obstacle and the UAV. Based on that, different conflict resolution algorithms are used in different levels of the region so as to achieve a better tradeoff in reaction time and optimality.

D. Supervision

The supervision module is in charge of maintaining the system healthy inspection, allowing or disallowing operations based on mission requirements, recording the flight logs, etc. This is an essential module for ensuring the stability and proper functioning of the system. In particular, it consists of *Commander* and *Logger* submodules that are explained below.

Commander inspects the healthy status of all modules, maintains a state machine in terms of the system level (e.g., the current mode), and allows or prohibits actions according to its current state or mission requirements. In addition, *Commander* is in charge of dispatching all the commands given by the ground station. Different commands may need different operators. For example, commands of mission execution, such as target tracking and formation flight need to be dispatched to the task planner; and commands of system management, such as the predefined trajectories loading and configuration changing need to be dispatched to the logging and storage.

Logger logs the state of the system, including the state transaction, trigger events, flight data, and custom logs. Other than this, the perceived images or videos can be stored on the platform selectively. This is very helpful for developers to analyze the UAV's behaviors and performance over the entire process. Besides, configuration files, parameter lists, and predefined waypoint lists are also stored onboard in order to fast fetch for onboard modules.

V. Coordination Layer

The coordination layer is in charge of the tasks related to negotiation (e.g., task allocation) among UAVs for mission coordination. To begin with, it is responsible of dynamically dividing the UAV swarm to individual coordination groups according to the mission requirements and the communication availability. This can restrict the scale of the states maintained on each UAV for making decision as the number of UAVs increases. Besides, it is also in charge of decomposing the mission to tasks and distributing the tasks among the UAVs in the coordination group. Note that there is a task planner in the high-level control layer being in charge of generating the detailed plan for the obtained tasks correspondingly. Moreover, it is responsible of reallocating the tasks as well as the task sequences while conflict is detected.

It is noted that for different missions, the negotiation work among the UAV swarms should be different, for example, the objective of task allocation for formation flight and target tracking is to decide which UAVs are the leaders or the follower and which targets should be tracked by which UAVs, respectively. Therefore, similar to the task planning module,

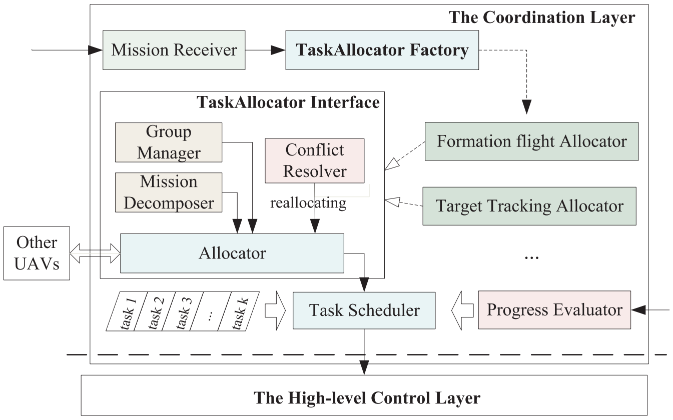


Fig. 3. Overview of the coordination layer.

we apply the factory pattern to include different task allocators. The design of the coordination layer are shown in Fig. 3.

Once a mission is received, the *TaskAllocator Factory* launches an associated task allocator according to the mission type. In each task allocator, it consists of the *Group Manager*, the *Mission Decomposer*, the *Allocator*, and the *Conflict Resolver*. More specifically, the *Group Manager* decides the coordination group based on the mission requirements and communication availability, e.g., clustering algorithms can be used to divide the groups. The *Mission Decomposer* decomposes the mission to tasks according to the preset configuration. The allocator allocates the tasks as well as the task sequence among UAVs in the coordination group. Many task allocation approaches, such as market-based [35] and optimization-based [36] mechanisms can be used here. For instance, we adopt a hierarchical market-based target assignment in the prototype system. In order to hold the scale of negotiation, the UAV swarm is divided by multiple predefined coordination groups. UAVs in the coordination group negotiate their tracking proposals until all the discovered targets are assigned. The *Conflict Resolver* detects the potential conflict and triggers reallocation while needed. After the task sequence is determined, the *Task Scheduler* schedules the tasks one by one in the task sequence queue. Note that there are two kinds of tasks in the perspective of scheduling: 1) blocking and 2) nonblocking tasks. When the blocking task is completed, the sequential task can be launched; whereas the nonblocking tasks are launched, the sequential task can be launched.

VI. Communication Layer

The communication layer is in charge of the messages transmission among all the UAVs and the ground control station (GCS), which is a critical desire for a highly autonomous and cooperative swarm system. This is because the essential tasks, including command execution, coordination messages, and sensed data feedback (e.g., position, image, and video) largely rely on the quality of communications. Furthermore, the communication system should include the following features.

1) The communication system not only needs the air-to-ground and ground-to-air links to support the connectivity between the ground station and UAVs but also requires the air-to-air links to help the information exchanges between different UAVs.
2) It is necessary to support different types of transmissions in the communication system which correspond to different quality of services. For example, the video (presenting in the ground station) and position information (supporting formation flight) have different data rate and latency requirements.
3) It is scalable to the number of UAVs in the swarm system.

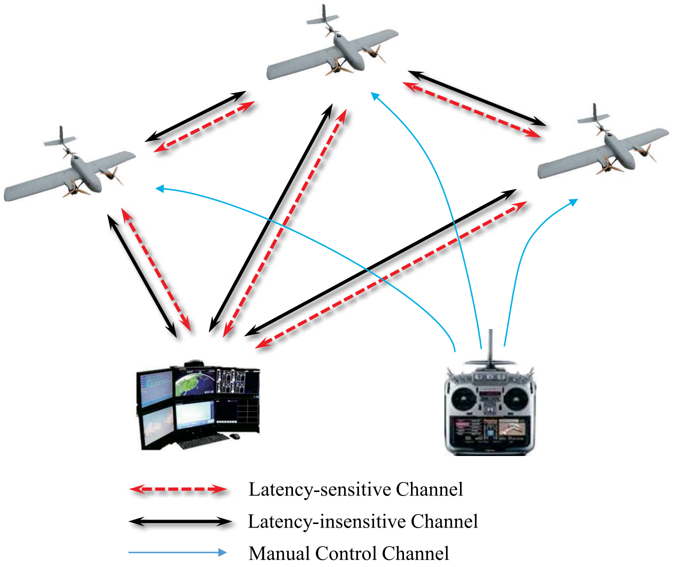


Fig. 4. Different channels of the communication infrastructure.

The traditional approach that relies on the separated link for each UAV is not feasible for UAV swarms any more, since simultaneous package transmission will overwhelm the communication system as the number of UAVs increased. Recently, many solutions based on IEEE 802.11 wireless LANs using infrastructure or *ad-hoc* mode have been proposed [15], [37]. However, most of them are used for short distance scenarios (e.g., in the order of 300 m [37]) and even indoor conditions. This restricts the working radius of swarm systems. In comparison, radio systems provide low-latency link and more than 10 km range, and current products of radio systems support the mesh network and point to multipoint network. Nevertheless, one of the drawbacks is that the bandwidth is relatively narrow, i.e., around 100 kb/s.

In order to satisfy the requirements (mentioned above) of autonomous UAV swarms while leveraging the advantages of different wireless techniques, our communication infrastructure consists of three types of channels, as shown in Fig. 4. First, a latency-sensitive channel based on radio systems that is used for command and coordination. We also propose a custom protocol for exchanging latency-sensitive messages based on mavlink [38]. By using this custom protocol, messages, including location, altitude and commands are simplified and transmitted over this channel. Second, a latency-insensitive channel is used for high-data-rate transmissions, and images and videos are transmitted over this channel. Third, a manual control channel based on a radio system that is used for remote control. Note that we rely solely on the first two links for autonomous control of UAV swarms. For the purpose of protecting UAVs when an accident happens, a remote control link to each individual UAV is needed.

Besides, the communication layer also includes the communication management software module for transferring messages effectively. More specifically, it is in charge of encapsulating and dis-encapsulating the messages, since the protocols used intra- and inter-aircraft are different. Furthermore, with the increase in the scale of swarms, optimizing the swarm communication is also becoming a desire. Optimization approaches, such as priority-based traffic scheduling [39], congestion control [40], and data rate minimization [41] can be included here.

VII. Human Interaction Layer

The human interaction layer is in charge of providing the interaction interfaces for the operator to control the swarm system. There are mainly two kinds of interfaces: 1) monitoring interfaces and 2) commanding interfaces. The monitoring interfaces are used for visualizing the situation of the UAV swarm system, including the status of UAVs, the sensed data, the environment, etc. The commanding interfaces are used for sending commands to the UAV swarm system for accomplishing the required missions. Since the UAV swarms may include a large number of UAVs, traditional GCS, such as QGroundControl [42] and Paparazzi System [43], which provide supervision and control in the flight control level, is not suitable for UAV swarms. This is because the cognition as well as the operating workload is too heavy and will overwhelm the operators. To this end, based on our previous experience of field experiments, we summarize the attributes that the GCS for UAV swarms should have as follows.

Monitoring Interfaces:

1) Information about UAV status and sensed data needs to be analyzed proactively before representation and the interested results should be highlighted.
2) Displays, panels, and information entries can be activated or deactivated selectively.

Commanding Interfaces:

1) Voice assistant can be used for announcing the current situation and sending commands.
2) High-level commands (e.g., search targets, track targets, and develop formation) should be arranged on the main panel. However, this relies heavily on the autonomy level that the swarm system can reach.
3) A workflow pattern of sending a sequence of mission commands can be adopted. High-level commands can be assembled arbitrarily by operators. GCS sends the commands one by one once the acknowledgment of the previous command is received, or triggers the failsafe policies once errors are detected.

Basically, different forms of GCS can be deployed based on the environmental condition, e.g., tablet, vehicles, ships, etc. Besides, due to the convenience for GCS to obtain resources, such as computing capacities, power supply, and high bandwidth network comparing to the aircraft, we can leverage the high-performance servers or even the data center facilities as the back end for GCS to perform complex computation and large-scale data storage. In this way, tasks, such as panoramic mosaic and 3-D modeling, which is time consuming in the traditional view, can be completed in real time.

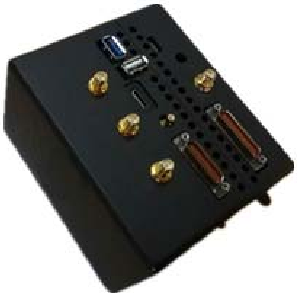


Fig. 5. Integrated onboard control system box.

VIII. Experiments and Results

We have built a prototype swarm system based on the proposed architecture and conducted a set of field experiments in the open environment with the area of 5 km^2. Besides, we have also conducted simulations for cases that real experiments are difficult to carry out. In the following sections, the experimental setup as well as the experimental methodology will be presented.

A. Experimental Setup

1) Field Experimental Setup: The field experimental setup mainly consists of aerial platforms, integrated onboard control system, and the GCS.

Aerial Platform: We use two types of UAVs in the experiments. The first type is a fixed-wing UAV and has a cruise speed of 19 m/s. The wing span of the vehicle is 1.80 m and the body length is 1.22 m. The total weight of the vehicle is 1.1 kg and maximum take-off weight is 4.6 kg. The second type is a tilt-rotor UAV and has a cruise speed of 23 m/s. The wing span of the vehicle is 1.80 m and the body length is 1.600 m. The total weight of the vehicle is 1.2 kg and maximum take-off weight is 6.89 kg. Both of these two UAVs use lithium battery as power supply and have a nominal endurance of 1 h.

Integrated Onboard Control System: For the purpose of designing a lightweight and miniaturized system, we propose an integrated onboard control system box that integrates the processing boards, perceptional devices, communication payloads, circuitry, and cooling devices into a compact container. This box has a weight of 450 g and a size of 108 mm(length) × 108 mm(width) × 110 mm(height), as shown in Fig. 5. More specifically, there are three layers in the hardware architecture. The first layer is mainly composed of a custom printed circuit board which is in charge of the power distribution and providing wiring interfaces. The second layer is mainly composed of a microcontroller unit which is connected to the propulsion system, servos, various sensors, etc. The third layer is mainly composed of a high-performance

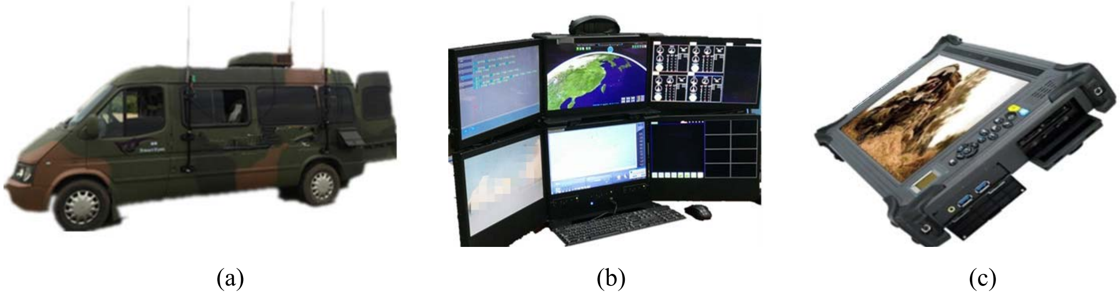


Fig. 6. Various forms of GCSs. (a) Vehicle-mounted GCS. (b) Multiscreen GCS. (c) Tablet GCS.

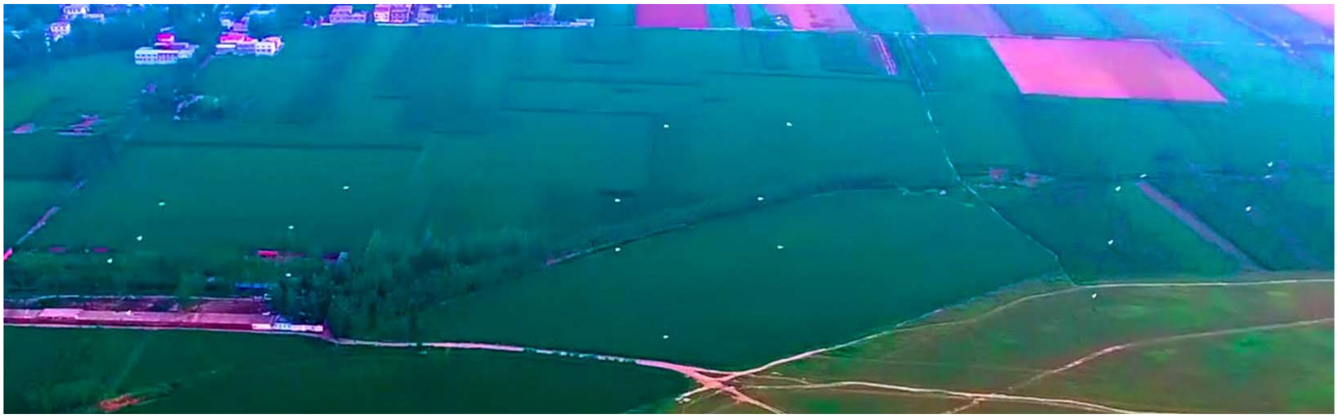


Fig. 7. Snapshot of the 21-UAV formation flight experiment. There are three groups. The seven UAVs in each group form two vertical lines.

processing unit which is connected to telemetry devices and communication payloads.

GCS: Fig. 6(a) shows the proposed vehicle-mounted GCS. This GCS mainly consists of interaction interfaces, communication devices, distributed computing servers, and UPS power supply. More specifically, the interaction interfaces can visualize the UAV status, sensed data, the geographical environment, as well as diagnosis information (e.g., communication status and wind status). Besides, the operator can command the swarm system through the interaction interfaces. Moreover, in order to satisfy different requirements of mobility and portability, we also propose other forms of GCSs as shown in Fig. 6(b) and (c). The tablet GCS is used for UAV pilots or high authorities that want to monitor the status of the swarm system. The multiscreen portable GCS is used in the environment where car is not convenient to drive (e.g., mountainous and lake area).

2) Simulation Set-Up: Simulations are conducted based on MATLAB 2016a. Each UAV is implemented as a separate running function. The communication networks are sparse that each UAV is linked to maximum 25% of the scale. The orbit perturbations are considered, which are modeled as white Gaussian noises with the diffusion coefficient 4 m^2/s in each axis.

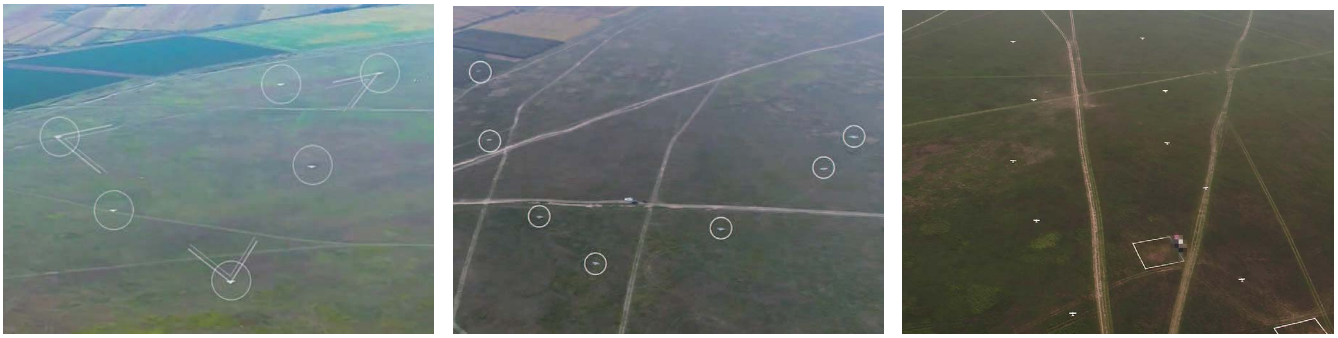
B. Versatility of the Proposed Architecture

In this section, we evaluate the versatility of the proposed architecture. To this end, we conduct a set of field experiments in which the prototype UAV swarm system operates multiple missions based on the proposed architecture. Here, we choose the missions of formation flight and target recognition and tracking as examples.

1) Formation Flight: Fig. 7 shows a snapshot of the 21-UAV formation flight experiment. In this experiment, the UAVs are divided into three groups. In each group, there are two leader UAVs and five follower UAVs. The leader UAVs execute the coordinated path-following guidance control law proposed in Section IV-C, while the follower UAVs execute the leader–follower coordination guidance control law. As shown in the figure, the seven UAVs in each group form two vertical lines.

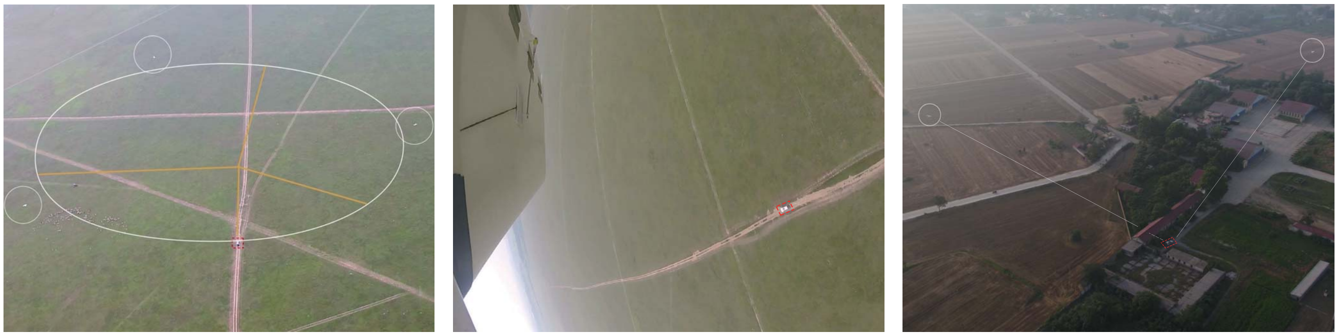
We have also performed formation pattern maintaining and reconfiguration during the experiment. Fig. 8 shows examples of different formation patterns that have performed in the experiment. Note that we adopt different strategies of formation pattern reconfiguration for leader and follower UAVs, respectively. For the leaders, the reconfiguration is achieved through coordinated path planning and following. With respect to the followers, we change their relative positions corresponding to their leaders while switching the formation pattern.

2) Target Recognition and Tracking: Fig. 9(a) shows a snapshot of the target recognition and tracking experiment with three UAVs. More specifically, each UAV captures videos through its electro-optical pod at runtime. UAVs detect the target through the target detection and recognition algorithm. Once a target is detected, by applying target localization and multi-UAV data fusion algorithm, UAVs can obtain the target information and update the trajectory estimation of the target.



(a) (b) (c)

Fig. 8. Formation pattern reconfiguration. (a) Triangle pattern. (b) V pattern. (c) Line pattern.



(a) (b) (c)

Fig. 9. Target recognition and tracking. (a) Cooperative Tracking with three UAVs. (b) Vision from a UAV. (c) Tracking under the case of shelter.

Fig. 9(b) shows a snapshot of the vision from one of the UAVs. In terms of tracking the target, we adopt cooperative standoff target tracking guidance approach in the experiment. Three UAVs fly a circular orbit around the target with a radius of 100 m, and each UAV keeps an angular spacing of 120° with others. This can achieve the surveillance of multiple angles, which is useful in the condition that the target is keeping out by the shelter, such as buildings and trees. As shown in Fig. 9(c), the target is sheltered by the building that one of the UAV cannot get a vision of the target.

Admittedly, due to the time and manpower limitation, we only demonstrate two different missions in the experiments. However, we want to stress that this article is the first work, to the best of our knowledge, which demonstrates formation flight and target tracking missions with an integrated architecture for fixed-wing UAV swarms through field experiments. We believe that other missions can also be supported in the proposed architecture through extension.

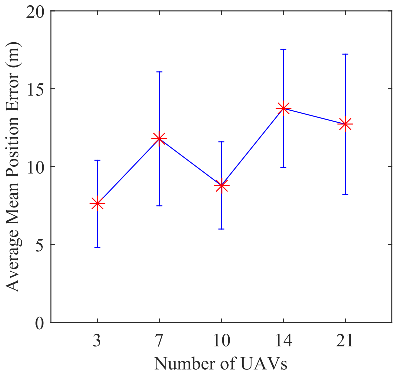


Fig. 10. Average MPE for formation flights with increasing number of UAVs in field experiments.

C. Scalability of the Proposed Architecture

In order to evaluate the scalability of the proposed architecture, we access the swarm system's behavior while increasing the scale of the swarm system. We take the formation flight as an example, and evaluate the performance of formation following behavior through a set of field experiments and simulations with increasing number of vehicles. We use AMPE [44], the average mean position error (MPE) over all swarm members as well as the time average, as the criterion

$$\text{AMPE} = \frac{1}{T_k}\sum_{t=1}^{T_k}\text{MPE}(t) \tag{7}$$

$$\text{MPE}(t) = \frac{1}{N}\sum_{i=1}^{N}\left\|P_i^D(t) - P_i^R(t)\right\| \tag{8}$$

where N is the number of vehicles performing the formation flight. T_k is the number of timestamps during the duration of the formation flight. $P_i^D(t)$ and $P_i^R(t)$ denote the desired and real position of UAV i at timestamp t, respectively. Fig. 10 shows AMPE for formation flights with 3, 7, 10, 14, and

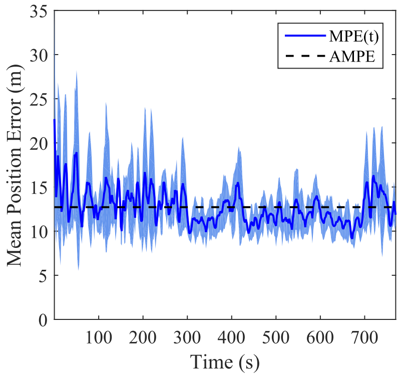


Fig. 11. MPE for a segment during the formation flight of 21 UAVs. The lightly shaded band indicates 1σ.

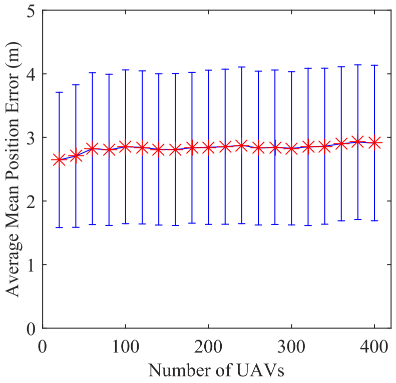


Fig. 12. Average MPE for formation flights with increasing number of UAVs in simulations.

21 UAVs in field experiments. It is clear that as the number of the vehicles increases, AMPE for formation flights hovers around 10 m. We can also see in Fig. 11 that the MPE for a segment during the formation flight floats up or down at the AMPE (12.7 m). In comparison, as shown in [15], the AMPE is 43.5 m for formation flight with 20 UAVs. Fig. 12 shows AMPE for formation flights with a scale of up to 400 UAVs in simulations. It is clear that AMPE hovers over at around 2.5 m as the number of the vehicles increases. Fig. 13 illustrates the 400-UAV formation's trajectories in simulations. It is shown that the formation pattern maintains well. Therefore, the proposed architecture shows good scalability while increasing the scale of the swarm system. Admittedly, the scalability of the architecture relies heavily on the communication infrastructure. And many solutions, such as hierarchical design and multicast can improve the scalability of the communication. However, this is out of the scope of this article. In this article, we focus on the architecture design at the level of the swarm system.

D. Launch and Landing Rate

Generally speaking, the endurance of miniaturized UAV is limited, which directly affects its mission capability. In order to reduce the time overhead of deploying UAV swarms into the sky, fast launch and recovery of UAV swarms is a basic desire. However, compared with a single UAV, launching and recovering a swarm of UAVs are much more challenging, especially for fixed-wing UAV. Rational planning and collision avoidance need to be considered since many UAVs gather around a limited airspace when launching and recovering. In our proposed swarm system, we launch and recover UAVs one by one through short distance taxing. The average taxing distances for launching and recovering are 20 and 50 m, respectively. Fig. 14 illustrates the evolution of a 21-UAV flight experiment, showing the number of UAV aloft since the launching command is sent. It is clear that this flight lasts for 1257 s, and at 110.43 s all 21 UAVs are aloft. Similar to [15], we use the mean time between launches as the criterion to measure the launch rate of UAV swarms, which is

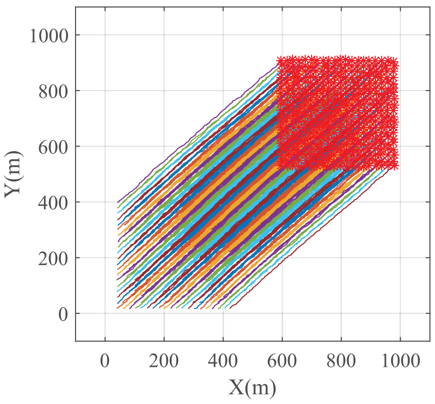


Fig. 13. 400-UAV formation's trajectories in simulations.

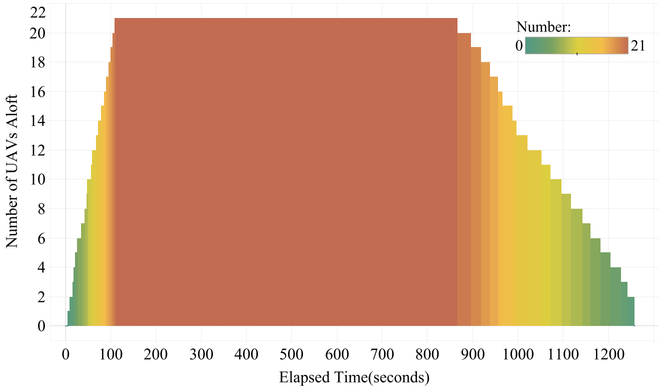


Fig. 14. Illustration of the evolution of the 21-UAV flight experiment, showing the number of UAV aloft with the time elapsed since the launching command is sent and depicting the sequential launch and landing of all UAVs.

$$\text{LauRate} = \frac{1}{N}\left(\tau_{\text{launch}}^{1} + \sum_{i=2}^{N}\left(\tau_{\text{launch}}^{i} - \tau_{\text{launch}}^{i-1}\right)\right) \tag{9}$$

where τ_{launch}^{i} is the time between the ith UAV is aloft and the launching command of the swarm system is sent. Accordingly, we can obtain the *LauRate* of our swarm system is 5.25 s. This is significantly faster than the launch rate in [15], which stays at 33.5 s.

Similarly, we can measure the landing rate by using the criterion as

$$\text{LandRate} = \frac{1}{N}\left(\tau_{\text{land}}^{1} + \sum_{i=2}^{N}\left(\tau_{\text{land}}^{i} - \tau_{\text{land}}^{i-1}\right)\right) \tag{10}$$

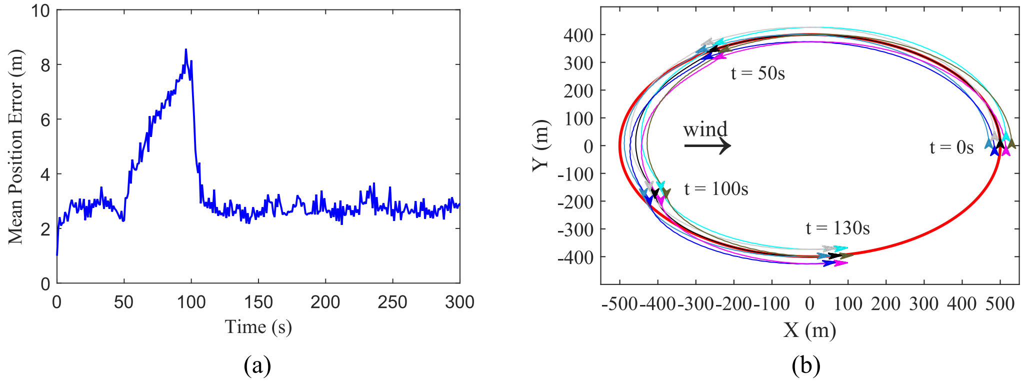


Fig. 15. Formation flight in case of wind disturbance. (a) MPE. (b) UAV trajectories.

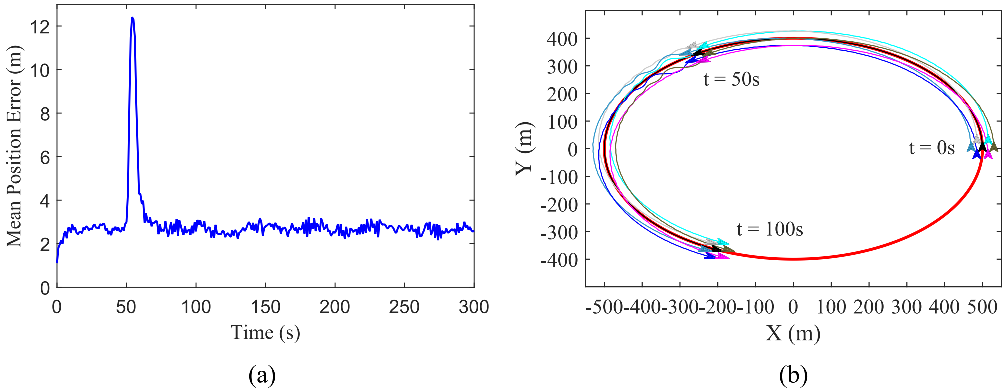


Fig. 16. Formation flight in case of communication interruption. (a) MPE. (b) UAV trajectories.

where τ^1_{land} is the time between the *i*th UAV is landed and the landing command of the swarm system is sent. Accordingly, we can obtain the *LandRate* of our swarm system is 21.71 s. Although Chung *et al.* [15] have not provided the precise land rate for their system, we can see from Fig. 7 in this article that it takes more than 25 min for landing 50 UAVs. Hence, we take the average of this duration to estimate the land rate in [15], which is 30 s. Therefore, the land rate of our swarm system is also faster than the land rate in [15].

E. Robustness to External Disturbance and Uncertainty

To evaluate the robustness of the architecture to external disturbance and uncertainty, we conduct simulation experiments in which wind disturbance and communication interruption is imported during formation flight of UAV swarms. With respect to wind disturbance, 21 UAVs perform an elliptical maneuver with semi-major axis of 1000 m and semi-minor axis of 800 m. There are three groups in this formation and each group forms a regular hexagon. Fig. 15(a) shows the MPE with a gust of wind of 8 m/s lasting 50 s (i.e., from 50 to 100 s) and Fig. 15(b) shows the corresponding trajectories of UAVs.[2] It is clear that the MPE increases when the wind is imported at 50 s. However, after the wind stops at 100 s, the MPE decreases dramatically and recovers to the value when wind starts at 119 s. Thus, the recovery time in this case is 19 s. From Fig. 15(b), we can also see that the UAV trajectories drift toward east under the effect of wind and recover after the wind stops.

In terms of communication interruption, 21 UAVs perform similar maneuver. Fig. 16(a) shows the MPE for formation flight with 5 s communication outage (i.e., from 50 to 55 s) during the flight and Fig. 16(b) shows the trajectories of UAVs. We can see that MPE raises significantly after the communication is interrupted and reaches 12.29 m at 55 s. This is because the follower UAVs keep their heading when they cannot access to their leader. After the communication is recovered, the formation error decreases gradually. The recovery time in this case is 16 s. Admittedly, this simple strategy may incur large error if their leader turns sharply during the communication interruption. More robust strategies such as trajectory estimation will be considered in our future work.

IX. Conclusion

In this article, we presented a multilayered and distributed architecture for mission-oriented miniature fixed-wing UAV swarms and a holistic view of a swarm system, including hardware, software, communication, and human interaction is detailed. Compared to the exiting solutions, the proposed architecture divides the overall system into five layers: 1) low-level control layer; 2) high-level control layer; 3) coordination layer; 4) communication layer; and 5) human interaction layer, and many modules with specified functionalities. As a result,

[2] In order to demonstrate the trajectories clearly, we only depict one group of UAVs in this figure. Other groups of UAVs have similar results.

not only the difficulty of developing a large system can be reduced but also the versatility of supporting diversified missions can be ensured. Besides, the proposed architecture is fully distributed that each UAV performs decision-making autonomously, thereby achieving better scalability. Moreover, different kinds of aerial platforms can be feasibly extended by using the control allocation matrix and the integrated hardware box. We conducted field experiments, including autonomous launch, formation flight, target recognition, and tracking and landing of 21 fixed-wing UAVs, to demonstrate the scalability and versatility of the proposed architecture. The experimental results also show that the launch rate of the prototype system based on the proposed architecture outperforms the state-of-the-art work.

There are some interesting future works in this direction. One possible future work is to extend other collective behaviors and coordination missions, such as search and rescue, forest fire control and cooperative surveillance or mapping, to a larger scale of UAV swarm. Another future work is to enhance the robustness to the complex and uncertain environments.

References

[1] Q. Luo and H. Duan, "Distributed UAV flocking control based on homing pigeon hierarchical strategies," *Aerosp. Sci. Technol.*, vol. 70, pp. 257–264, Nov. 2017.

[2] H. Chen, X. Wang, L. Shen, and Y. Cong, "Formation flight of fixed-wing UAV swarms: A group-based hierarchical approach," *Chin. J. Aeronaut.*, to be published, doi: 10.1016/j.cja.2020.03.006.

[3] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 3299–3304.

[4] X. Bai, W. Yan, S. S. Ge, and M. Cao, "An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field," *Inf. Sci.*, vol. 453, pp. 227–238, Jul. 2018.

[5] L. Jin and S. Li, "Distributed task allocation of multiple robots: A control perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 5, pp. 693–701, May 2018.

[6] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 4, pp. 1537–1548, Apr. 2020.

[7] S. Minaeian, J. Liu, and Y. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[8] R. Wang, J. Du, Z. Xiong, X. Chen, and J. Liu, "Hierarchical collaborative navigation method for UAV swarm," *J. Aerosp. Eng.*, vol. 34, no. 1, 2021, Art. no. 04020097. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/(ASCE)AS.1943-5525.0001216

[9] J. N. Yasin, S. A. S. Mohamed, M. H. Haghbayan, J. Heikkonen, and J. Plosila, "Navigation of autonomous swarm of drones using translational coordinates," in *Proc. Int. Conf. Pract. Appl. Agents Multi Agent Syst.*, 2020, pp. 353–362.

[10] Z. Yuan, J. Jin, L. Sun, K.-W. Chin, and G.-M. Muntean, "Ultra-reliable IoT communications with UAVs: A swarm use case," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 90–96, Dec. 2018.

[11] J. L. Sanchez-Lopez *et al.*, "AEROSTACK: An architecture and open-source software framework for aerial robotics," in *Proc. Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Arlington, VA, USA, 2016, pp. 332–341.

[12] V. Grabe, M. Riedel, H. H. Bulthoff, and P. R. Giordano, "The TeleKyb framework for a modular and extendible ROS-based quadrotor control," in *Proc. Eur. Conf. Mobile Robots*, Barcelona, Spain, 2013, pp. 19–25.

[13] J. Boskovic, N. Knoebel, N. Moshtagh, J. Amin, and G. Larson, "Collaborative mission planning & autonomous control technology (compact) system employing swarms of UAVs," in *Proc. AIAA Guid. Navig. Control Conf.*, 2009, p. 5653.

[14] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, "Swarms of unmanned aerial vehicles—A survey," *J. Ind. Inf. Integr.*, vol. 16, pp. 100–106, Dec. 2019.

[15] T. H. Chung, M. R. Clement, M. A. Day, K. D. Jones, D. Davis, and M. Jones, "Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, 2016, pp. 1255–1262.

[16] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and Control of Mini-Flying Machines*. London, U.K.: Springer-Verlag, 2005.

[17] C. Liu and W.-H. Chen, "Disturbance rejection flight control for small fixed-wing unmanned aerial vehicles," *J. Guid. Control Dyn.*, vol. 39, pp. 2810–2819, Dec. 2016.

[18] S. Zhao, X. Wang, and D. Zhang, "Curved path following control for fixed-wing unmanned aerial vehicles with control constraint," *J. Intell. Robot. Syst.*, vol. 89, no. 3, pp. 107–119, 2017.

[19] D. Zhang and X. Wang, "Autonomous landing control of fixed-wing UAVs: From theory to field experiment," *J. Intell. Robot. Syst.*, vol. 88, nos. 2–4, pp. 619–634, 2017.

[20] J. Pedro and T. Tshabalala, "PI-based fault tolerant control for fixed-wing UAVs using control allocation," *IFAC-PapersOnLine*, vol. 50, no. 2, pp. 181–186, 2017.

[21] O. Kechagias-Stamatis and N. Aouf, "A new passive 3-D automatic target recognition architecture for aerial platforms," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 406–415, Jan. 2019.

[22] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018. [Online]. Available: arXiv:1804.02767.

[23] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[24] J. Morton and J. Odell, *Object Oriented Analysis and Design*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[25] H. Oh, S. Kim, H.-S. Shin, and A. Tsourdos, "Coordinated standoff tracking of moving target groups using multiple UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 2, pp. 1501–1514, Apr. 2015.

[26] J. Xiong and Y. Niu, "Guidance law for multi-UAVs collaborative ground target tracking under obstacle environment," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, Chongqing, China, 2017, pp. 7219–7223.

[27] S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen, "Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 8, pp. 2897–2904, Aug. 2020.

[28] M. Kothari, I. Postlethwaite, and D.-W. Gu, "A suboptimal path planning algorithm using rapidly-exploring random trees," *Int. J. Aerosp. Innovat.*, vol. 2, no. 1, pp. 93–140, 2010.

[29] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *J. Guid. Control Dyn.*, vol. 30, no. 6, pp. 1718–1728, 2007.

[30] X. Wang, Z. Zeng, and Y. Cong, "Multi-agent distributed coordination control: Developments and directions via graph viewpoint," *Neurocomputing*, vol. 199, pp. 204–218, Jul. 2016.

[31] M. A. Dehghani and M. B. Menhaj, "Communication free leader–follower formation control of unmanned aircraft systems," *Robot. Auton. Syst.*, vol. 80, pp. 69–75, Jun. 2016.

[32] H. Qiu, H. Duan, and Y. Fan, "Multiple unmanned aerial vehicle autonomous formation based on the behavior mechanism in pigeon flocks," *Control Theory Appl.*, vol. 32, no. 10, pp. 1298–1304, 2015.

[33] A. Askari, M. Mortazavi, and H. Talebi, "UAV formation control via the virtual structure approach," *J. Aerosp. Eng.*, vol. 28, no. 1, 2013, Art. no. 04014047.

[34] Y. Wang, X. Wang, S. Zhao, and L. Shen, "A hierarchical collision avoidance architecture for multiple fixed-wing UAVs in an integrated airspace," 2020. [Online]. Available: arXiv:2005.14455.

[35] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, Aug. 2009.

[36] Z. Wu, Z. Li, Z. Ding, and Z. Li, "Distributed continuous-time optimization with scalable adaptive event-based mechanisms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3252–3257, Sep. 2020.

[37] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013, pp. 120–124.

[38] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga. (2013). *MAVLink: Micro Air Vehicle Communication Protocol*. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[39] J. Huang, H. Wang, Y. Qian, and C. Wang, "Priority-based traffic scheduling and utility optimization for cognitive radio communication infrastructure-based smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 78–86, Mar. 2013.

[40] M. Rajesh and J. Gnanasekar, "Congestion control using aodv protocol scheme for wireless ad-hoc network," *Adv. Comput. Sci. Eng.*, vol. 16, nos. 1–2, p. 19, 2016.

[41] Y. Cong, X. Zhou, and R. A. Kenney, "Finite blocklength entropy-achieving coding for linear system stabilization," *IEEE Trans. Autom. Control*, early access, Mar. 10, 2020, doi: 10.1109/TAC.2020.2979763.

[42] (2019). *QGroundControl*. [Online]. Available: http://qgroundcontrol.com

[43] B. Gati, "Open source autopilot for academic research—The Paparazzi system," in *Proc. IEEE Amer. Control Conf. (ACC)*, Washington, DC, USA, 2013, pp. 1478–1481.

[44] I. Navarro and F. Matia, "A proposal of a set of metrics for collective movement of robots," in *Proc. Workshop Good Exp. Methodol. Robot.*, 2009, pp. 1–6.

Zhihong Liu received the B.S. degree in computer science from the South China University of Technology, Guangzhou, China, in 2009, and the M.S. and Ph.D. degrees in computer science from the National University of Defense Technology, Changsha, China, in 2011 and 2016, respectively.

He was a visiting student with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, from 2013 to 2015. He is currently an Assistant Professor with the College of Intelligence Technology, National University of Defense Technology. His research interests include the coordination of multiple UAVs, UAV swarming, and visual serving.

Xiangke Wang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control science and engineering from the National University of Defense Technology, Changsha, China, in 2004, 2006, and 2012, respectively.

Since 2012, he has been serving as a Lecturer, an Associate Professor, and a Professor with the College of Intelligence Science and Technology, National University of Defense Technology. He was a visiting student with the Research School of Engineering, Australian National University, Canberra, ACT, Australia, from 2009 to 2011. He has authored or coauthored two books and more than 100 publications in peer-reviewed journals and international conferences, including IEEE TRANSACTIONS, *International Journal of Robust and Nonlinear Control*, CDC, IFAC, and ICRA. His current research interests focus on the control of multiagent systems and its applications on unmanned aerial vehicles.

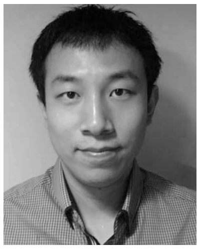
Lincheng Shen received the B.S., M.S., and Ph.D. degrees in automatic control from the National University of Defense Technology (NUDT), Changsha, China, in 1986, 1989, and 1994, respectively.

In 1989, he joined the Department of Automatic Control, NUDT, where he is currently a Full Professor and serves as the Dean of the Graduate School. His research interests include unmanned aerial vehicles, swarm robotics, and artificial intelligence.

Dr. Shen has been serving as an Editorial Board Member of the *Journal of Bionic Engineering* since 2007.

Shulong Zhao received the B.S. degree in automation from Beihang University, Beijing, China, in 2011, and the M.S. and Ph.D. degrees in control science and engineering from the National University of Defense Technology, Changsha, China, in 2013 and 2017, respectively.

He is currently a Lecturer with the National University of Defense Technology. His research interests include data-driven control and curved path following of UAV.



Yirui Cong (Member, IEEE) received the B.E. degree in automation from Northeastern University, Shenyang, China, in 2011, the M.Sc. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2013, and the Ph.D. degree from Australian National University, Canberra, ACT, Australia, in 2018.

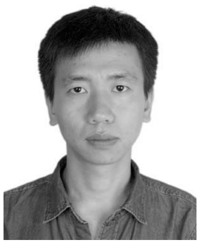
He is currently a Lecturer with the National University of Defense Technology. His research interests are in the fields of communication theory and control theory.

Jie Li received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2006, 2008, and 2014, respectively.

From 2014 to 2019, he was a Lecturer with NUDT. Since 2020, he has been an Associate Professor with NUDT. His research interests include swarm intelligence, emergent behavior, multiagent cooperation, distributed control, and collective decision making.

Dong Yin received the B.S. degree in network security, the M.S. degree in control theory and engineering, and the Ph.D. degree in control science and engineering from Northwestern Polytechnical University, Xi'an, China, in 2004, 2007, and 2011, respectively.

In 2011, he joined the Faculty of the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China, where he is currently an Associate Professor with tenure in 2017. His research interests include swarm networking and relative positioning.



Shengde Jia received the B.S. degree in automatic control from the Harbin Institute of Technology, Harbin, China, in 2008, and the M.S. and Ph.D. degrees in control science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011 and 2015, respectively.

He is currently an Assistant Professor with the College of Intelligence Technology, NUDT. His research interests include machine learning and UAV control.

Xiaojia Xiang received the B.S., M.S., and Ph.D. degrees in automatic control from the National University of Defense Technology (NUDT), Changsha, China, in 2003, 2007, and 2016, respectively.

He is currently an Associate Professor with the College of Intelligence Science and Technology, NUDT. He works in the field of mission planning, and autonomous and cooperative control of unmanned systems.