

Research Article

Modelling and Simulation of Distributed UAV Swarm Cooperative Planning and Perception

Haifeng Ling ¹, Hongchuan Luo ^{1,2}, Haisong Chen,¹ Linyuan Bai,¹ Tao Zhu ¹,
and Yanjun Wang²

¹College of Field Engineering, Army Engineering University of PLA, Nanjing 210003, China

²No. 94789 Unit of PLA, Nanjing 210003, China

Correspondence should be addressed to Hongchuan Luo; luohc2004@outlook.com

Received 29 March 2021; Revised 14 May 2021; Accepted 19 May 2021; Published 30 May 2021

Academic Editor: Mahmut Reyhanoglu

Copyright © 2021 Haifeng Ling et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an emerging topic, the swarm of autonomous unmanned aerial vehicles (UAVs) has been attracting great attention. Due to the indeterminacy of sensors, distributed cooperative swarms have been considered to be efficient and robust but challenging to design and test. To facilitate the development of distributed swarms, it has been proposed to utilise a simulation platform for cooperative UAVs using imperfect perception. However, the existing simulation platforms cannot satisfy this demand due to a few reasons. First, they are designed for a specific purpose, and their functionalities are difficult to extend. Second, the existing platforms lack compatibility to be applied to different types of scenarios. Third, the modelling of these platforms is too simplified to simulate flight motion dynamic and noisy communication accurately, which may cause a difference in performance between the simulation and real-world application. To address the mentioned issues, this paper models the problem and proposes a simulation platform for distributed swarm cooperative perception, which addresses software engineering concerns and provides a set of extendable functionalities of a cooperative swarm, including communication, estimation, perception fusion, and path planning. The applicability of the proposed platform is verified by simulations with the real-world application. The simulation results demonstrate that the proposed system is viable.

1. Introduction

A swarm of autonomous unmanned aerial vehicles (UAVs) has been an emerging topic in the fields of manufacture, disaster rescue, and the military [1]. Autonomous unmanned swarms are expected to outperform a single complex UAV in terms of flexibility and robustness, thus offering enhanced adaptability, survivability, and fault tolerance. For instance, in strike coordination and reconnaissance missions, a swarm of UAVs can cover a given area in a shorter time than a single one, and a failure of any swarm member will not cause a failure of the entire mission, as shown in Figure 1. However, due to the complexity of the joint decision, the larger the swarm is, the higher the demand for cooperative planning and perception algorithm will be. To achieve the aforementioned benefits, the interswarm cooperative planning and perception algorithm, as well as many other factors, need to be studied in-depth.

First, the interswarm planning and perception algorithm should be fully distributed. Otherwise, the swarm is vulnerable to the failure of certain member. Take a centralised swarm as a counterexample, where a swarm is controlled by a central member (leader), who fuses the perceptions of members and makes a planning decision for them. Once the leader fails, the swarm also fails; thus, the swarm is considered to be nonresilient. Second, the interswarm planning and perception algorithm should be intelligent. The swarm performance should increase with the swarm scale. Lastly, to be applicable in the real-world, physical constraints of the swarm, such as packet loss in communication and noise influenced sensor perception, need to be taken into account.

Studies on the swarm cooperative planning and perception have been limited by many factors, such as the difficulty of a real-world experiment due to the lack of funds to build a UAV swarm, a risk of damage caused by a fall down [2] of a

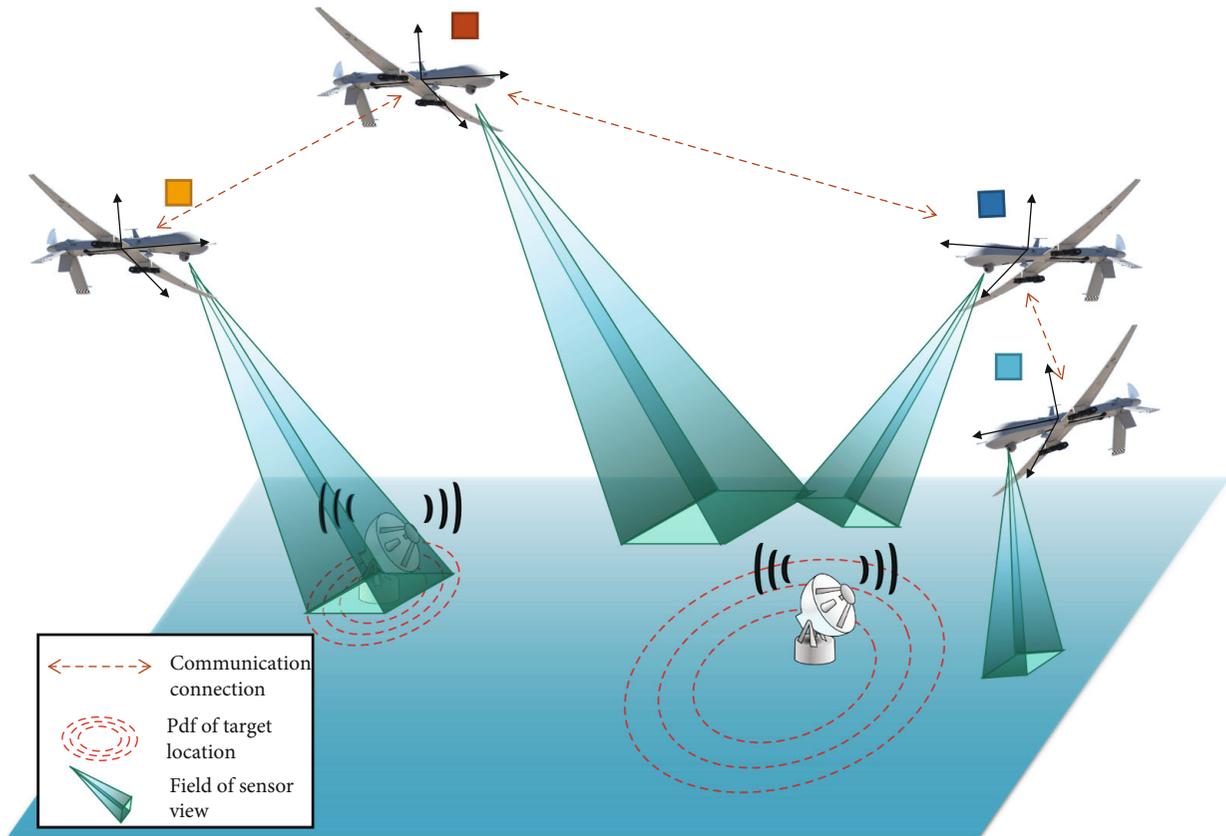


FIGURE 1: Illustration of a swarm of UAVs executing strike coordination and reconnaissance missions.

UAV in the swarm, and the regulation that constrains UAV flight in urban areas. Therefore, simulation-based validation and testing can significantly facilitate the study of multiple UAV swarm planning and control. Vásárhelyi et al. [3] optimised the parameters of the proposed decentralised guiding algorithm, which enabled large swarms of autonomous drones to navigate in confined spaces seamlessly. The optimised algorithm was tested in real-world applications using a swarm of 30 drones, and the obtained test results coincided well with the expectations. Researchers in the ETH-MAV team in the 2017 Mohamed Bin Zayed International Robotics Challenge tested their swarming algorithm for competition [4] before a real flight using Gazebo [5] as the simulation environment. In most studies, the proposed algorithms have been verified only by numerical simulation. Researchers from the US Air Force Research Laboratory [6] proposed a multiple autonomous vehicle visiting routine planning (VRP) algorithm and validated it by numerical simulations. A framework for generating feasible trajectories in motion coordination problems proposed in [7] was also validated by simulations. Shao et al. [8, 9] developed a state estimator-based minimal learning parameter (SE-MLP) observer using simulation verification to deal with uncertainties appearing in individual quadrotors of the swarm.

Currently, there are many commercial and open-source simulation platforms for multi-UAV planning and control. The MultiUAV2 [10] released by the US Air Force Research Laboratory, which is part of the MATLAB and Simulink soft-

ware, represents a simulation platform capable of simulating multiple UAVs that cooperate to accomplish tactical missions. However, this platform may cause a copyright problem to the research group without a commercial license. The CoUAV [11] enables rapid implementation of simulation for multiple UAVs, and its source code and demo are available for public access, but in this platform, a member in a UAV swarm is controlled by a ground control centres, which forms a centralised rather than a distributed scheme. The “Infoplanner” simulation platform that was open-sourced by Schlotfeldt et al. [12] is capable of simulating decentralised cooperative planning using an imperfect sensor. However, due to its specific design and lack of documentation, extensions to the system, such as an extension of a sensor model or communication setting, are challenging. An expandable, maintainable, and commercial license-free simulation platform is unavailable nowadays.

To address the mentioned challenges, this paper proposes a simulation platform that provides flexibility in defining varieties of motion dynamics, sensor models, and communication settings; convenience in integration different objectivities with planning algorithm; out-of-the-box trajectory plotting and multiround Monte Carlo simulation; and no requirement of commercial license. The proposed simulation platform is developed with software engineering concerns, such as maintainability and expansibility, and is applied in a military operation study [13, 14].

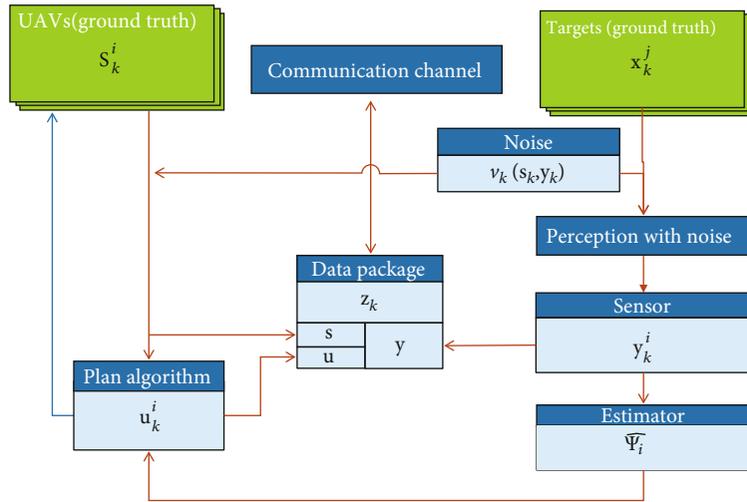


FIGURE 2: Overall structure of the problem formulation.

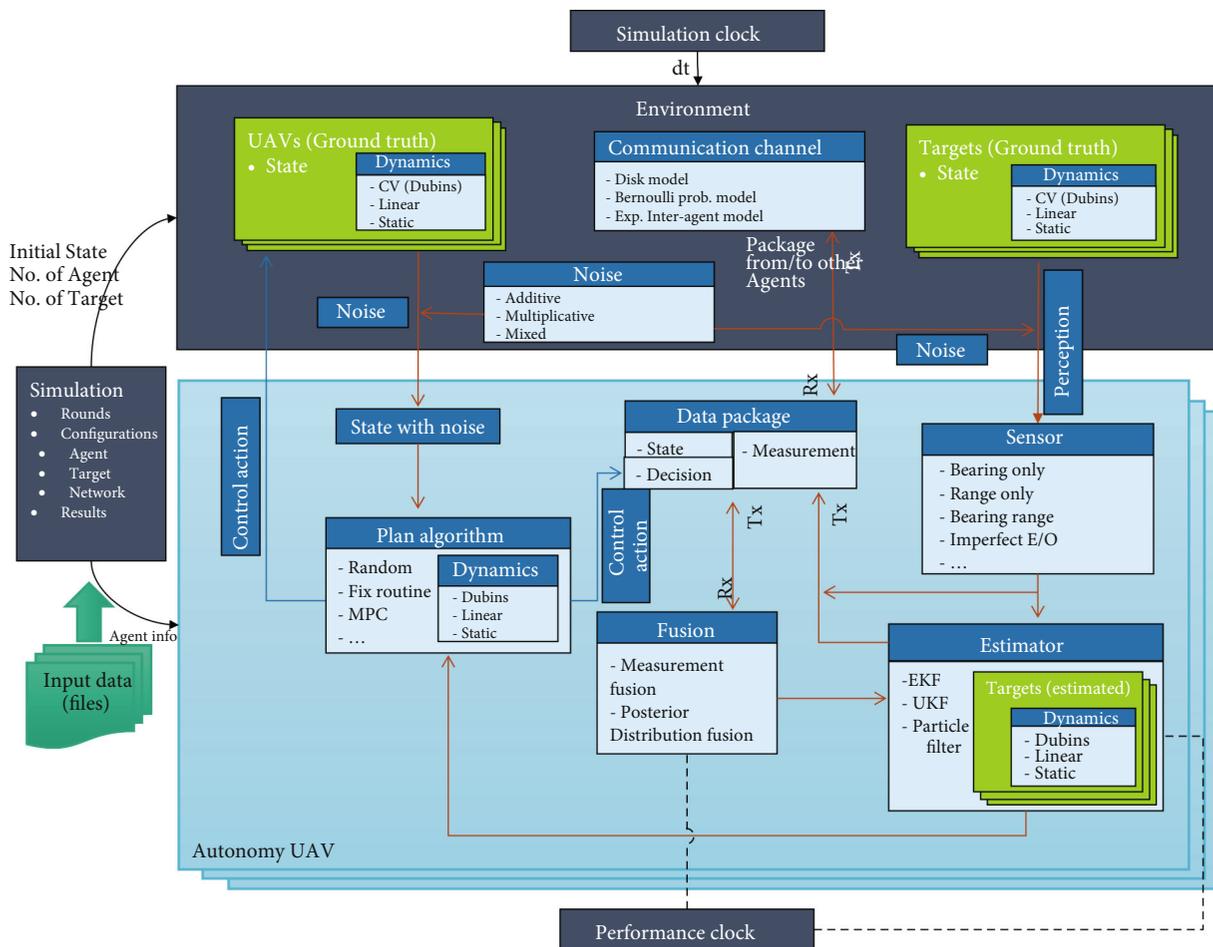


FIGURE 3: The structure of the proposed domain model.

TABLE 1: Simulation parameters.

Parameter	Value	Parameter	Value
Area of interest	100 km × 100 km	Target start position	(200, 500)
Unit length	100 m	Unit logical step	1 min
True target motion model \mathbb{P}_t	$\mathbb{P}_t(\mathbf{x}_{k+1} \mathbf{x}_k) = f(\mathbf{x}_k) + \text{noise}_1$ $\text{noise}_0 \sim N(0, D)$	Determined part of target state transition $f(x_k)$	$\begin{bmatrix} x_k^E + 6 \cos 9^\circ \\ x_k^N + 6 \sin 9^\circ \\ \varphi_k + 9^\circ \\ \alpha \end{bmatrix}$
D in noise_0	$\begin{bmatrix} 3 & & & \\ & 3 & & \\ & & 1 & \\ & & & 0 \end{bmatrix}$	Planning target motion model \mathbb{P}'_t	$\mathbb{P}'_t(\mathbf{x}_{k+1} \mathbf{x}_k) = \mathbf{x}_k + \text{noise}_1$ $\text{noise}_1 \sim N(0, 15)$
UAV start position	(-170, -400) (-300, -550) (-430, -4.00) (-300, -250)	UAV start heading	$\varphi_0 = 0$
UAV linear speed	$v = 90$ km/h	UAV start energy amount (fuel)	$\gamma_0 = 1500$ ml
Constant fuel consumption	$c = 3$ ml	Control space (turning option)	$u = \{-15^\circ, 0^\circ, 15^\circ\}$
Turning fuel consumption	$d = 2$ ml	Number of samples for planning	$n_s = 200$
Number of particles	$n_p = 200$	Sensor parameter φ_{fov}	$\varphi_{fov} = 30^\circ$
Sensor parameter μ	$\mu = 0.8$	Sensor parameter ξ	$\xi = 0.25$
Sensor parameter α	$\alpha \sim \text{Rayleigh}(0.96)$	Sensor parameter $g_G(\psi)$	$g_G(\psi) \sim N(0, 2)$
Sensor parameter $g_t(\psi)$	$g_t(\psi) \sim t(1)$	Objective weight $\eta_k(\gamma)$	$3.5 \times \gamma$

The remainder of this paper is organised as follows. Section 2 introduces the system modelling. Section 3 presents simulation cases of cooperative locating radio emitter with an airborne radio receiver to demonstrate the usefulness of the proposed platform. Lastly, Section 4 concludes the paper and presents future work directions.

2. System Modelling

2.1. Overview. In this section, the swarm planning and perception procedure is modelled in a general pattern, as shown in Figure 2. Also, the system design and architecture of the simulation platform which are aimed at supporting rapid and robust development and implementation are introduced, and they are shown in Figure 3. To apply the simulation platform to different scenarios, UAV platforms, perception apparatus, and target motion characteristics, the system modelling adopts a highly abstract way which highlights the interaction among the UAVs, targets, perception, and communication. Every part of the modelling can be further replaced with a specific detailed model. The replacement is also known as concrete implementation of an abstract representation in software engineering.

2.2. Problem Formulation. The problem is formulated in a lockstep discrete way wherein the states of each module are computed synchronically. This design enabled that the simulation can run faster or slower than real time and pause at any time. Consider a swarm consisting of

n_a UAVs with a discrete motion dynamic. Then, it can be written that

$$\mathbf{s}_{k+1}^i = f^i(\mathbf{s}_k^i, \mathbf{u}_k^i), \quad i \in A = \{1, \dots, n_a\}, \quad (1)$$

where vector $\mathbf{s}_k^i \in \mathcal{S}^i \cong \mathbb{R}^{n_s}$ represents the n_s dimensional state of UAV i at time k and vector $\mathbf{u}_k^i \in \mathcal{U}^i$ is the control action applied to the UAV i at time k , which is one of all possible decisions \mathcal{U}^i .

Further, consider n_t targets with Markovian state transition so that

$$\mathbf{x}_{k+1}^j \sim \mathbb{P}_t(\mathbf{X}_{k+1}^j | \mathbf{X}_k^j), \quad j \in T = \{1, \dots, n_t\}, \quad (2)$$

where \mathbf{x}_k^j is the state vector of a target j at time k , \mathbf{X}_k^j is the set of all possible states of the target j , and $\mathbb{P}_t(\cdot)$ is the transition probability of the target j . It should be noted that the semi-Markovian system can be converted into the Markovian system by extending the state. According to real-world application experience, it is assumed that a target has no awareness of the existence of the swarm and that its own underlay state transition model is unavailable for the swarm. Considering the Bayesian interference, a swarm model target state transition can be expressed as $\mathbf{x}_{k+1}^j \sim \mathbb{P}'_t(\mathbf{X}_{k+1}^j | \mathbf{X}_k^j)$, $j \in T = \{1, \dots, n_t\}$, where \mathbb{P}'_t is derived from the swarm's prior knowledge, and in the most accurate case, it holds that that $\mathbb{P}'_t = \mathbb{P}_t$.

In a realistic application, perceptions of a UAV are imperfect. The perception result obtained based on the states

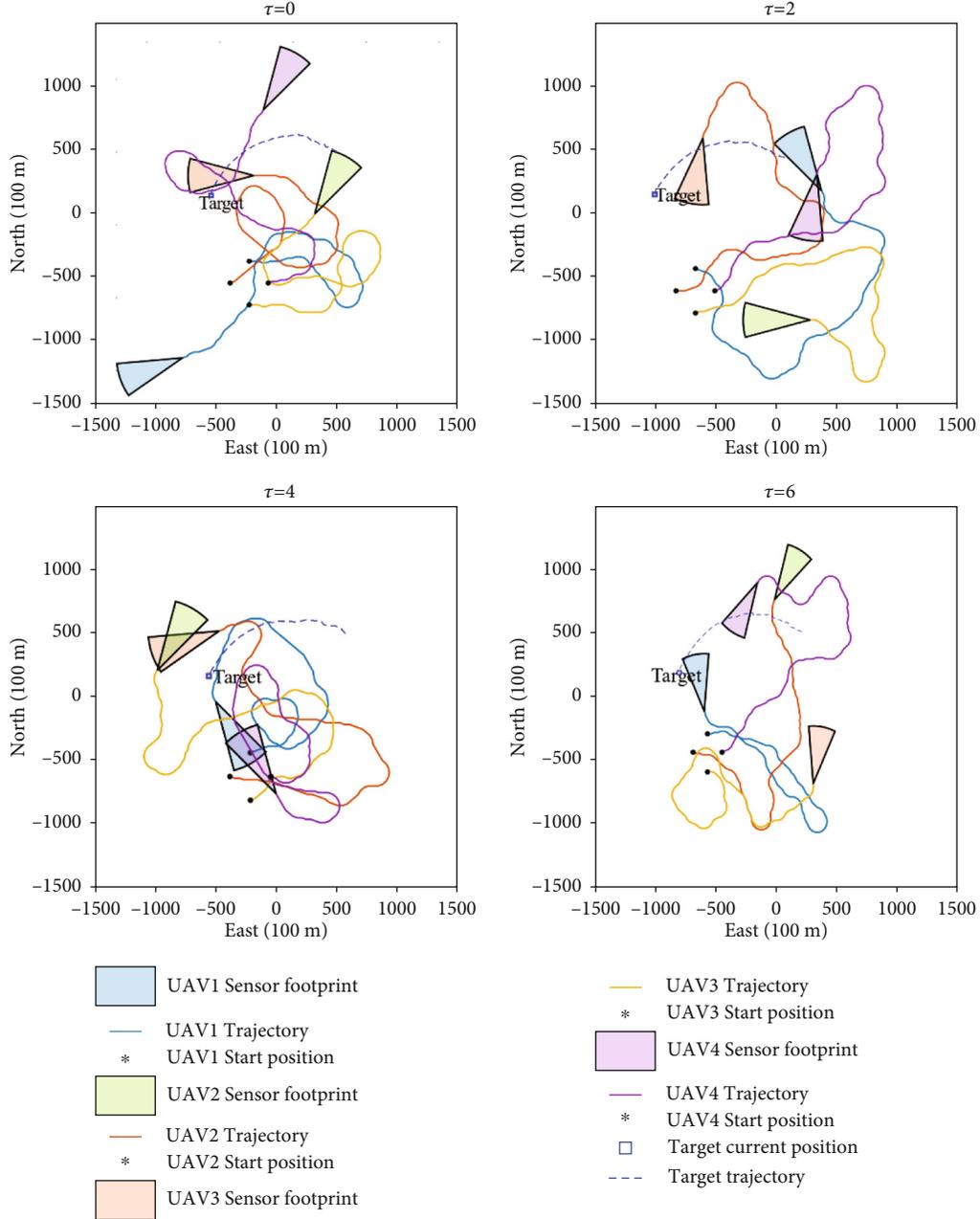


FIGURE 4: The simulated trajectory. In all four cases, the initial conditions except for the receding horizon step τ were the same. In the upper-left figure, $\tau = 0$ stands for greedy exploit.

of UAVs i and target j and environmental noise can be modelled as follows:

$${}^i\mathbf{y}_k^j = h\left(\mathbf{s}_k^i, \mathbf{x}_k^i, \mathbf{v}_k\left(\mathbf{s}_k^i, \mathbf{x}_k^i\right)\right), \quad (3)$$

where ${}^i\mathbf{y}_k^j \in \mathbb{R}^{n_y}$ denotes the perception result of a target j observed by a UAV i at time k , \mathbf{v}_k is the noise dependent on the state and environment, and $h(\cdot)$ models the properties of perception.

This work considers an imperfect interswarm communication network, where every two members $l, m \in A$, can share their states, perceptions, and control decisions if needed. Due

to the environment interference and packet loss, the communication between UAV can be modelled as a probabilistic model of the received packet. Assume ${}^l_m\mathbf{z}_k$ is the received packet sent by UAV m to UAV l at time k , \mathbf{Y}_k^m is the UAV m 's perception result of all targets and ${}^l_m\mathbf{Z}_k$ is the set of all cases of ${}^l_m\mathbf{z}_k$. In addition, denote the probability function of communication distribution as $\mathbb{P}_c(\cdot)$. Then,

$${}^l_m\mathbf{z}_k \sim \mathbb{P}_c\left({}^l_m\mathbf{Z}_k | \mathbf{Y}_k^m; \mathbf{s}_k^l, \mathbf{s}_k^m, \mathbf{u}_k^l, \mathbf{u}_k^m\right), \quad l, m \in T = \{1, \dots, n_a\}. \quad (4)$$

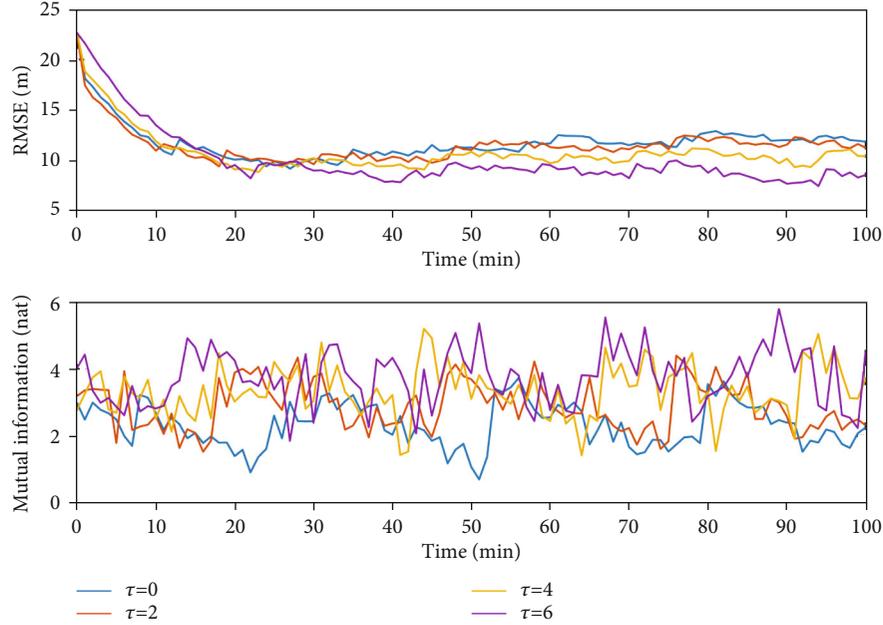


FIGURE 5: Average RMSE and mutual information of 100 simulation runs.

TABLE 2: Mean and standard deviation of RMSE at the end of simulation and mutual information during the simulation from 100 simulation runs.

Scenario	Mean and standard deviation of RMSE at the end of simulation and mutual information during the simulation			
	RMSE mean (m)	RMSE Stdev	Mutual information mean (nat)	Mutual information Stdev
$\tau = 0$	11.871	0.3922	2.322	0.4538
$\tau = 2$	11.200	0.3948	2.893	0.4209
$\tau = 4$	10.360	0.3454	3.348	0.4702
$\tau = 6$	8.505	0.3512	3.746	0.4317

In this work, the decision process on cooperative planning and perception is considered as a distributed optimal control problem with the above constraints. Assume that a quantity of interest Ψ represents the yields of the behaviours of UAV members; then, it can be written that

$$\begin{aligned}
 U^* &= \underset{U}{\operatorname{argmax}} \Psi(\mathbf{s}; \mathbf{x}; \mathbf{y}; \mathbf{z}), \\
 \text{s.t. } &\begin{cases} (1), (2), (3), (4), \\ \mathbf{u}^i \in \mathcal{U}^i, \end{cases} \quad (5)
 \end{aligned}$$

where U^* denotes the optimal control decision of all UAV members in a swarm, U is the set of all control decisions available for UAVs, and $(\mathbf{s}; \mathbf{x}; \mathbf{y}; \mathbf{z})$ is a set consisting of the UAV state, target state, perception result, and communication packet. The underlying form of Ψ is determined by the mission of a swarm. For instance, in the motion planning problem, Ψ can be defined as a negative integration of the

throttle along the path under the preference of saving energy. A distributed control decision is made individually by a swarm member aiming to maximise the union quantity of yields of the entire team. The decision can be made by one of the three possible approaches of the control theory: closed-loop control (feedback rule), open-loop control (planning once), and open-loop feedback control (planning and replanning along the receding horizon). Due to the distributed scheme of the problem, every UAV makes a decision based on only its own information and received packet. Let $\hat{\Psi}_i$ be the local estimation of the union yield of a swarm by UAV i . The problem can be rewritten as

$$\begin{aligned}
 \mathbf{u}_i^* &= \underset{\mathbf{u}_i}{\operatorname{argmax}} \hat{\Psi}_i(\mathbf{s}_i; \mathbf{y}_i; \mathbf{z}_i), \\
 \text{s.t. } &\begin{cases} (1), (2), (3), (4), \\ \mathbf{u}^i \in U^i, \\ i \in \{1, \dots, n_a\}, \end{cases} \quad (6)
 \end{aligned}$$

where $(\mathbf{s}_i; \mathbf{y}_i; \mathbf{z}_i)$ is the set consisting of the i th UAV's state, perception results of UAV i , and all communication packets received by UAV i .

2.3. Platform Implementation. During the implementation of the simulation platform, issues concerning maintainability and functional encapsulation were addressed. Since the simulation platform development is a complex task for a small research group, the main purpose of the system design process is to manage complexity and reuse codes while ensuring that the previously formulated problem can be implemented [15].

The Python programming language is chosen as the primary programming language because its popularity ensures

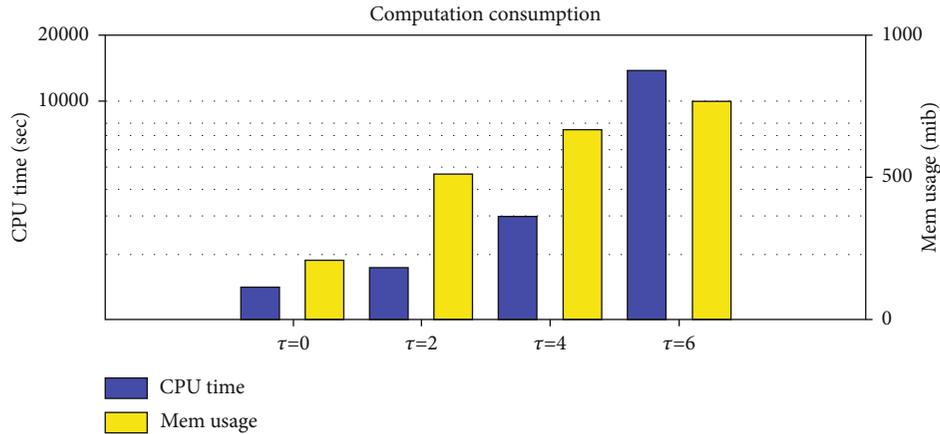


FIGURE 6: Computation cost of 100 simulation runs. The CPU time (blue) is presented in the logarithmic scale, and memory usage (yellow) is displayed in the linear scale.

the availability to developers in small research groups, development productivity, an abundance of relevant packages, and amplexness of reference cases. Namely, Python is a developer-friendly language, which is easy to learn and code even for a beginner, and a novice researcher can learn Python basics in a relatively short time. In contrast, the other alternative high-level languages, such as C++ and Java, and emerging programming languages, such as Go, requires much more time to master, so a small research group may lack a developer with enough experience in programming in these languages. In addition, unlike MATLAB that requires a paid license, Python has a prosperous open-source community providing many free-of-charge implementations of an interpreter. The features of Python language, including dynamic typing, being object-oriented, and coroutine supporting, which are included in all Python versions since version 3.7, can greatly improve development productivity. Being well-known, one of the main Python tendencies is to writes less code with the help of concise and expressive grammar. The packages managed in PyPI or other source code hosting sites such as GitHub also promote Python, thus increasing its popularity. The SciPy adds efficient matrix and statistics manipulation function to Python, while the SimPy makes building a discrete-time simulation much easier for Python using coroutine.

The key to managing complexity is constructing a domain model representing the essentials of a problem to be addressed [16]. As the distributed cooperative planning and perception problem has been formulated before, the next step is to refine concepts from the problem formulation and to design the information flow between components. The overall structure of a domain model is displayed in Figure 3. The presented structure uses ubiquitous language during the implementation of the proposed platform and bridges the gap between the formula and code.

A class, which originates from object-oriented programming (OOP), helps a developer to integrate data with function, enabling structure reuse and keeping the interface intact while preserving the ability to modify details and implementations. Being well-known, the scattered duplicated

codes can significantly jeopardise the maintainability of software and lead to certain defects; in particular, codes cannot be reused directly before certain modifications. To prevent the deterioration of code in advance, the structure presented in Figure 3 is encapsulated in a class in the proposed simulation platform. It should be noted that the object-oriented syntax of Python is concise and expressive.

The proposed platform consists of two major top-level components, *Environments* and *Autonomy UAV*. The *Environments* component enacts the pivotal role as a simulator of the real world and preserves the whole ground truth data in its subcomponents. Being isolated from the ground truth data in the *Environment*, *Autonomy UAV* has multiple instances enacting each UAV in a swarm and retrieving noise-contaminated data generated by the *Noise* component and ground truth data from *Environment*. Subcomponents of *Environment* are *UAVs*, *Targets*, *Communication Channel*, and *Noise*. Information exchange between UAVs is facilitated by introduction of the *Communication Channel* component, which judges the availability of packets according to the communication model. The subcomponents of *Environment*, *UAVs*, and *Targets* compute the state evolution using the dynamics assigned by *Simulation*. Note that in the proposed solution, the dynamic can be different from that in *Plan Algorithm* in that a simplified proxy model can be used in planning. *Sensor* component in *Autonomy UAV* defines the format of perception data and provides a likelihood model for the *Fusion* component and *Plan Algorithm* component. Combining the perception data and *Fusion* model, the *Estimator* component estimates the target state using the target model assigned to *Autonomy UAV* component, which is also not necessarily the same as that in *Targets*. The *Plan Algorithm* provides the control based on the UAV state, estimation result, and packed data obtained from *Communication Channel* component. The control action, UAV state, and estimation are gathered as packed data and sent to the other *Autonomy UAV* components, as well as to the *Environment* component, to change the UAV state.

The remaining three components of the top level are *Simulation*, *Simulation Clock*, and *Performance Clock*.

Simulation loads parameters in *config* files, calls functions to initialize the simulation, and saves simulation data for plotting. *Simulation Clock* maintains a discrete event priority queue and drives the simulation forward. *Performance Clock* records the computation resource usage data when a feature is enabled. This component is implemented in the aspect-oriented scheme using “Wrapper” (a.k.a. decorator), a tool in Python, which allows performance monitoring without modification in a simulation code.

3. Cooperative Locating Radio Emitter with Airborne Radio Receiver

This section introduces the simulation platform configured to model a simulation case of real-world applications, where a swarm of UAVs equipped with an airborne radio receiver attempts to locate a moving ground target emitting radio signals with limited airborne fuel. The main properties of this task are as follows. First, and most importantly, the swarm is completely autonomous and distributed, so trajectories of UAVs in the swarm are planned online according to the distributed cooperative decision of the swarm rather than pre-defined by operators. Second, due to the nonlinearity and presence of non-Gaussian noise in sensor perception, the posterior distribution is highly nonunimodal, and the standard deviation criteria have a limited guiding significance in the planning of swarm trajectories. Thus, a quantity of interest must be designed to indicate the location of a moving target, as well as energy conservation. Lastly, due to the manoeuvrability constraint of a fixed-wing UAV, the planning of swarm trajectories must consider future returns in advance.

3.1. Energy-Aware Motion Model of UAV in Swarm. It is assumed that during the locating process, swarm members are flying at fixed speed and height [17] and action candidates are at finite discrete yaw angle, which is expressed as

$$\mathbf{s}_{k+1} = \begin{bmatrix} s_{k+1}^E \\ s_{k+1}^N \\ \varphi_{k+1} \\ \gamma_{k+1} \end{bmatrix}^T = \begin{bmatrix} s_k^E + v \cos(\varphi_{k+1})\Delta t \\ s_k^N + v \sin(\varphi_{k+1})\Delta t \\ \varphi_k \\ \gamma_k - c \cdot \Delta t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \Delta t \\ d \cdot \Delta t \end{bmatrix} \mathbf{u}_k, \quad (7)$$

where s^E and s^N denote a swarm member’s coordinates in a north-east coordinate plane, which is defined by the air zone; v is a fixed velocity; φ is the yaw angle; $\gamma \geq 0$ is a real number representing the remaining energy or power of a UAV (once $\gamma = 0$, a UAV fails); $c > 0$ is a fix energy consumption ratio for flying; and, lastly, $d = \eta(\mathbf{u})$ denotes the steering cost.

3.2. Radio Emitter Target Model. In the simulation scenario, two different models of a radio emitter are used in the *Environment* and *Planning* components of the platform for computing the ground truth and the planning of UAV, respectively. The state vector of a moving target can be expressed as $\mathbf{x} = [x^E \ x^N \ \alpha]^T$, where x^E, x^N are the target coordinates in the north-east coordinate plane, the same as

that of swarm members, and $\alpha > 0$ is an unknown constant relating to the transmit power of the radio emission. It should be noted that the swarm has no prior knowledge about \mathbb{P}_t .

3.3. Sensor Model of Airborne Radio Receiver. The airborne radio receiver is assumed to precept the range and bearing of a radio emitter based on the emission signal’s strength and direction-of-arrival. The perceptions are polluted by noise, and in accordance with reality, the bearing perception is polluted by non-Gaussian noise.

The strength perception of radio emission can be modelled using the relative emission electronic power as follows:

$$\text{rss} = \mu \log_{10} \left((s^E - x^E)^2 + (s^N - x^N)^2 \right) - \alpha, \quad (8)$$

where $\mu > 0$ is a constant related to the performance of a signal receiver and α is a Rayleigh distribution noise introduced by the environmental and thermal noise in the radio receiver [18]. It should be noted that radio emission is undetectable when $\text{rss} \leq 0$, representing the fact that the sensitivity of a radio receiver is limited.

The bearing perception is modelled as follows:

$$\text{bear} = \tan^{-1} \left(\frac{x^N - s^N}{x^E - s^E} \right) + \omega, \quad (9)$$

where ω denotes the non-Gaussian noise introduced by a random distortion of the radio waveform. The random variable ω is generated from a weighted mixture of Gaussian distribution and a “long-trailing” noise distribution (Student t distribution) [19], which can be expressed as

$$\omega = \xi g_t(\psi) + (1-\xi)g_G(\psi), \quad (10)$$

where g_t is the probability density function of the t distribution, g_G is the probability density function of the Gaussian distribution, and $\xi \in [0, 1]$ is the weight representing the non-Gaussian degree. It should be noted that, in real-world scenarios, radio emission can be detected only in a limited range of direction-of-arrival due to the structural characteristics of a radio receiver antenna.

Thus, the sensor model of an airborne radio receiver can be defined as

$$\mathbf{y}_k = \begin{cases} \begin{bmatrix} \text{rss} \\ \text{bear} \end{bmatrix} & \text{if } \text{rss} \geq 0, \quad |\text{bear} - \varphi_k| \leq \varphi_{fov}, \\ \emptyset, & \text{else,} \end{cases} \quad (11)$$

where φ_{fov} is a constant representing the maximum angle of the field of view with respect to the yaw of the UAV.

3.4. Channel Model of Interswarm Communication. The detection, a.k.a. perception results, are sent with the UAV states \mathbf{s} to other UAV members via a packet loss channel. The packet loss probability is directly proportional to the distance between the two communicating UAVs. The two outcomes, *succeed to reach* and *fail to reach*, follow the

Bernoulli distribution, whose parameters can be obtained from empirical data [20]. Assume ${}^l_m\beta_k(\mathbf{s}_k^l, \mathbf{s}_k^m)$, where m represents a UAV and l represents the ground vehicle, and ${}^l_m\beta_k(\mathbf{s}_k^l, \mathbf{s}_k^m)$ denotes the probability distribution of the Bernoulli distribution at time k ; then, it can be expressed as

$${}^l_m\beta_k(\mathbf{s}_k^l, \mathbf{s}_k^m) = 0.5 \times \operatorname{erfc} \left(\frac{\sqrt{(\Delta s^E)^2 + (\Delta s^N)^2} - \alpha_0}{\alpha_1} \right) - \alpha_2, \quad (12)$$

where α_i represents constant experimental parameter.

Then, (9) can be written as

$$\mathbb{P}_c \left({}^l_m\mathbf{Z}_k | \mathbf{s}_k^l, \mathbf{s}_k^m \right) = \begin{cases} {}^l_m\beta_k(\mathbf{s}_k^l, \mathbf{s}_k^m), & \text{if } {}^l_m\mathbf{Z}_k = \{\mathbf{y}_k^m, \mathbf{s}_k^m, \mathbf{u}_k^m\}, \\ 1 - {}^l_m\beta_k(\mathbf{s}_k^l, \mathbf{s}_k^m), & \text{if } {}^l_m\mathbf{Z}_k = \emptyset. \end{cases} \quad (13)$$

3.5. Bayesian Inference-Based Site Locating. There are many uncertainty sources along with the perception information flow from a UAV to the ground vehicle, including noise, fault detection, and communication loss. To estimate the location of a lost pilot, as well as the estimation accuracy, a distributed iterative particle filter is utilised. Assuming there is poor prior information about the location of the lost pilot, the prior distribution $P(s_t)$ is chosen as a uniform distribution on the map. Particle sets with weights $\langle \hat{s}_t^k, w^k \rangle$ are sampled from the prior distribution. The particle set evolves with each perception result or packet reception by the forward state prediction (13) and weight update, which can be expressed as

$$w^{k+1} \propto w^k P(Z_k | s_t). \quad (14)$$

In order to mitigate the particle degeneracy problem, a standard importance resampling step is introduced after the particles' weights are updated.

3.6. Quantity of Interest for Cooperative Decision. As a characterization of uncertainty, the mutual information of the radio emitter state and accumulated perceptions is chosen as one of the two quantities of interest used to evaluate perception quantity. Due to the imperfectness and nonlinearity of UAV perceptions, the radio emitter location acquired by the Bayesian inference is always a nonunimodal probability distribution. In such a situation, other measures like covariance are not appropriate [21]. Another quantity of interest is the sum of distances between the target and UAVs with a negative coefficient related to energy γ .

Assume $\operatorname{dis}_i = \sum_i^{n_a} \sqrt{(\mathbf{s}_i^N - \mathbf{x}^N)^2 + (\mathbf{s}_i^E - \mathbf{x}^E)^2}$ represents the sum of distances between the UAVs and the target; then, the objective Ψ in (10) can be expressed as

$$\Psi = \sum_i^{n_a} [I(\mathbf{Y}, \mathcal{Z}) + \eta(\gamma_i) \cdot \operatorname{dis}_i], \quad (15)$$

where $I(\cdot)$ denotes the mutual information that can be derived from [22]; \mathcal{Z} is the union set of perceptions of a UAV member perceived by itself and received from other members, which can be further decomposed as $\mathcal{Z} = \{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_{n_a}\}$; and $\eta(\gamma) < 0$ is the coefficient that characterizes the tendency to approach the target with respect to the remaining energy.

3.7. Distributed Cooperative Planning. It is hard to obtain a solution to (15) in the closed form. First, except for a special form of a dynamic system, it is hard to obtain an optimal feedback law of U^* as a closed form even in a centralised scheme [23]. In this study, the global optimal control law is approximated with an open-loop independent decision series in a time receding horizon τ , which is resolved in each time step. Assume $\bar{\Psi}_k = I_k(\mathbf{Y}, \mathcal{Z}) + \eta_k(\gamma_i) \cdot \operatorname{dis}_k$ represents the quantity of interest at time k ; then, Ψ in (10) can be approximated by

$$\Psi \approx \sum_k^{k+\tau} \bar{\Psi}_k, \quad (16)$$

where $\mathbf{u}_{k+1}^i, \mathbf{u}_{k+2}^i, \dots, \mathbf{u}_{k+\tau}^i$ are the independent decision series of a UAV i . A detailed explanation of coordinate descent was given in [12], and the receding horizon control in cooperative perception was presented in detail in [24].

The last problem to solve is the estimation of the quantity of interest in the receding horizon $k+1, k+2, \dots, k+\tau$. In this work, the sampling-based method [25] is used. Assume that samples of the target state are $\hat{\mathbf{x}}_k^{(q)} \sim \hat{p}(\mathbf{x}_k | \mathbf{x}_{k-1})$ and $\hat{\mathbf{y}}_k^{(q)} \sim \hat{p}(\mathbf{y}_k | \mathbf{x}_k, \mathbf{u}_k)$, where $\hat{p}(\mathbf{x}_k | \mathbf{x}_{k-1})$ and $\hat{p}(\mathbf{y}_k | \mathbf{x}_k, \mathbf{u}_k)$ are derived from the weighted particle set generated by the Bayesian inference; then, an approximation of $\bar{\Psi}_k$ is expressed as

$$\bar{\Psi}_k \approx \tilde{\Psi}_k = \frac{1}{N} \sum_m^N \Psi \left(\hat{\mathbf{x}}_k^{(q)}, \hat{\mathbf{y}}_k^{(q)}, \mathbf{u}_k \right). \quad (17)$$

Inspired by the motion primitive method, a motion primitive graph can be constructed by taking s_k as the root node, selecting or sampling \mathbf{u}_k^i from the available control action space \mathcal{U}^i as the edge, connecting edge \mathbf{u}_k^i to the candidate node $\hat{\mathbf{s}}_{k+1}^i$, and calculating the quantity of interest $\tilde{\Psi}_k$ in each candidate node $\hat{\mathbf{s}}_{k+1}^i$ as the weight of the edge. Because the decision space is discrete and $\tilde{\Psi}_k$ can be enumerating and evaluated, the scale of the motion primitive graph is limited. The optimal control action sequence can be obtained by a search algorithm like A*.

3.8. Simulation Results. The simulation is conducted in a square area of interest of 10,000 km². The UAV swarm is formed by four small fixed winged UAVs with a two-cylinder propeller. The target is moving in a constant linear velocity of 18 km/h and a turning rate of 9 degrees/min. The target's motion is also disturbed in speed and turning rate (see noise₀ in Table 1). The UAVs have no prior

knowledge of the target's motion and assume the target is randomly walking (see planning target motion model \mathbb{P}'_t in Table 1). The parameters of the airborne sensor for perception and other parameters for simulation initialization are given in Table 1. To eliminate the randomness and explore the effect of τ , four simulation scenarios with different τ were designed, and simulations were conducted 100 times for each scenario.

The trajectory snaps of one of the simulations four different τ values are presented in Figure 4. Due to the uncertainty property, the trajectories of both the target and the swarm are different in each run. Thus, the overall estimation performances, including accuracy and credibility, are depicted in an average performance of 100 simulation runs (see Figure 5 and Table 2). The accuracy was measured by the average root mean square error relating to the target true position (the lower the error was, the better the performance

was), which was calculated by $RMSE_k = \sqrt{(\bar{x}_k^E - x_k^E)^2 + (\bar{x}_k^N - x_k^N)^2}$, where \bar{x}_k^E, \bar{x}_k^N are the maximum a posteriori estimation values. The credibility was measured by the average mutual information derived from the particle set of the swarm member; the higher the amount of the average mutual information was, the better the credibility was.

The results in Figure 5 and Table 2 show that the higher τ resulted in higher mutual information and lower RMSE. This indicates that the increase in the time horizon τ can help to improve the performance of the swarm in the way of earlier turning for gaining more quantity of interest and avoiding extra turns.

The computation resource consumption of the simulation is plotted in Figure 6. Based on the results in Figure 6, the improvements in accuracy and credibility were obtained at the cost of an exponential increase in the computation cost. In addition, under the exhausting energy (see the end part of Figure 5), the swarm tended more to approaching the target than gathering the perception, which coincides well with the designer expectation.

4. Conclusion

This paper presents an extensible and maintainable simulation platform for distributed swarm cooperative perception planning considering the uncertainty in communication and perception. Simulation cases of evaluating the Bayesian inference-based estimation under imperfect perception and evaluating distributed cooperative planning are considered to demonstrate the operating principle and usefulness of the proposed simulation platform.

Since the modelling and implementation of the simulation consider the real-world constraint, such as unstable communication and noisy perception, the proposed solution can be beneficial to distributed cooperative swarm development and application.

However, the proposed simulation platform can be further improved by implementing additional mode functionalities, such as a new sensor model, communication model in complex terrain, and cooperative planning algorithm based

on reinforcement learning, which will be part of our future work.

Data Availability

The simulation condition data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declared that they have no conflicts of interest.

Acknowledgments

This work was supported by the Military Science Research under Grant JY2019B0XX and Frontier Innovation of National Defence under Grant 17-163-11-ZT-XXX-XX.

References

- [1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.
- [2] X. Shao, Y. Shi, and W. Zhang, "Fault-tolerant quantized control for flexible air-breathing hypersonic vehicles with appointed-time tracking performances," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 2, 2021.
- [3] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018.
- [4] R. Bähnemann, M. Pantic, M. Popović et al., "The ETH-MAV team in the MBZ International Robotics Challenge," *Journal of Field Robotics*, vol. 36, no. 1, pp. 78–103, 2019.
- [5] *Gazebo* Available from: <http://gazebosim.org/>.
- [6] S. Rasmussen, K. Kalyanam, S. Manyam, D. Casbeer, and C. Olsen, "Practical considerations for implementing an autonomous, persistent, intelligence, surveillance, and reconnaissance system," in *IEEE Conference on Control Technology and Applications (CCTA)*, vol. 2017, pp. 1847–1854, 2017.
- [7] H. Jaleel and J. S. Shamma, "Distributed submodular minimization and motion planning over discrete state space," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 932–943, 2020.
- [8] X. Shao, Z. Cao, and H. Si, "Neurodynamic formation maneuvering control with modified prescribed performances for networked uncertain quadrotors," *IEEE Systems Journal*, pp. 1–12, 2020.
- [9] X. Shao, Y. Shi, W. Zhang, and H. Cao, "Neurodynamic approximation-based quantized control with improved transient performances for MEMS gyroscopes: theory and experimental results," *IEEE Transactions on Industrial Electronics*, p. 1, 2020.
- [10] S. Rasmussen, J. W. Mitchell, P. Chandler, C. J. Schumacher, and A. L. Smith, "Introduction to the MultiUAV2 simulation and its application to cooperative control research," in *Proceedings of the 2005, American Control Conference*, vol. 2005, pp. 4490–4501, IEEE, 2005.
- [11] W. Wu, Z. Huang, F. Shan et al., "CoUAV: a cooperative UAV fleet control and monitoring platform," 2019, <http://arxiv.org/abs/190404046>.

- [12] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *Ieee Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.
- [13] H. Luo, H. Ling, T. Zhu, and Z. Kang, "Complete coverage path planning of UUV for marine mine countermeasure using grid division and spanning tree," in *2019 Chinese Control And Decision Conference (CCDC)*, pp. 5016–5021, IEEE, 2019.
- [14] H. Ling, H. Luo, L. Bai, T. Zhu, Q. Wang, and L. Yu, "Cooperative loitering munition swarm online patrolling route planning with nonlinear seeker measurement," *Mathematical Problems in Engineering*, vol. 2020, 10 pages, 2020.
- [15] S. McConnell, *Code Complete*, Wash, Microsoft Press, Redmond, 2004.
- [16] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley, Boston, 2004.
- [17] G. Gramajo and P. Shankar, "An efficient energy constraint based UAV path planning for search and coverage," *International Journal of Aerospace Engineering*, vol. 2017, 13 pages, 2017.
- [18] N. H. Nguyen, K. Doğançay, and W. Wang, "Adaptive estimation and sparse sampling for graph signals in alpha-stable noise," *Digital Signal Processing*, vol. 105, p. 102782, 2020.
- [19] D. Middleton, "Non-Gaussian noise models in signal processing for telecommunications: new methods and results for class A and class B noise models," *IEEE Trans Inform Theory*, vol. 45, no. 4, pp. 1129–1149, 1999.
- [20] M. Stachura and E. Frew, "Communication-aware information-gathering experiments with an unmanned aircraft system," *Journal of Field Robotics*, vol. 34, no. 4, pp. 736–756, 2017.
- [21] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, "Particle filter based entropy," in *2010 13th International Conference on Information Fusion*, pp. 1–8, IEEE, Edinburgh, 2010.
- [22] J. Ajgl and M. Šimandl, "Differential entropy estimation by particles," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11991–11996, 2011.
- [23] A. Bagchi, *Optimal Control of Stochastic Systems*, Prentice Hall, New York, 1993.
- [24] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering," in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, 2019.
- [25] E. H. Aoki, A. Bagchi, P. Mandal, and Y. Boers, *Particle Filter Approximations for General Open Loop and Open Loop Feedback Sensor Management*, Memorandum/Department of Applied Mathematics, 2011.