Intelligent UAV Swarm Cooperation for Multiple Targets Tracking

Zhou, Longyu; Leng, Supeng; Liu, Qiang; Wang, Qing

DOI
10.1109/JIOT.2021.3085673

Publication date
2021

Document Version
Final published version

Published in
IEEE Internet of Things Journal

Intelligent UAV Swarm Cooperation for Multiple Targets Tracking

Longyu Zhou, Supeng Leng, *Member, IEEE*, Qiang Liu, *Member, IEEE*, and Qing Wang, *Senior Member, IEEE*

***Abstract*—With the advantages of easy deployment and flexible usage, unmanned aerial vehicle (UAV) has advanced the multi-target tracking (MTT) applications. The UAV-MTT system has great potentials to execute dull, dangerous, and critical missions for frontier defense and security. A key challenge in UAV-MTT is how to coordinate multiple UAVs to track diverse invading targets accurately and consecutively. In this article, we propose a UAV swarm-based cooperative tracking architecture to systematically improve the UAV tracking performance. We design an intelligent UAV swarm-based cooperative algorithm for consecutive target tracking and physical collision avoidance. Moreover, we design an efficient cooperative algorithm to predict the trajectory of invading targets accurately. Our simulation results demonstrate that the swarm behaviors stay stable in realistic scenarios with perturbing obstacles. Compared with state-of-the-art solutions, such as the matched deep *Q*-network, our algorithms can increase tracking accuracy by 60%, reduce tracking delay by 23%, and achieve physical collision-avoidance during the tracking process.**

***Index Terms*—Mobile target tracking, prediction, scheduling, unmanned-aerial-vehicle (UAV) swarm intelligence (SI).**

I. Introduction

MULTIPLE targets tracking (MTT) for the Internet of Things (IoT) has been intensively investigated with the advance of low-cost and miniaturized devices. These devices can be beforehand deployed around a forbidden region or placed in an examination area to detect and track invaders [1]. However, such MTT systems have limited capability to track multiple invaders in complicated scenarios. For instance, the MTT applied to the frontier defense may incur many blind detection zones where invaders can easily steal into neighboring countries. Recently, unmanned aerial vehicles (UAVs) is regarded as a potential paradigm to advance the development of MTT systems. The natural advantage of UAVs is to sense and track mobile targets on an extensive scale with flexible mobility. This improves the tracking capability of MTT systems in scenarios, such as damage assessment, disaster monitoring, and border patrol [2]. Taking the security of the nuclear power station for an instance, UAVs can discover and track illegal invaders, such as flying birds and adverse drones, for ensuring manufacturing safety.

However, several challenges affect the UAV tracking performance in MTT systems. The first challenge is the dynamic mobile trajectories of invading targets. Without knowing the invading purposes of mobile targets, UAVs cannot directly predict multiple time-varying trajectories by inertial measurements. The diverse target trajectories can also make UAV scheduling critical to track multiple targets simultaneously. In this case, UAVs may cause the loss of tracking targets, especially when targets invade with high speeds. The second challenge is the limited energy budget on UAVs. Keeping consecutive tracking is crucial for UAVs in many practical scenarios. Because of the limited energy budget, UAVs cannot steadily maintain their expected speeds to enforce real-time tracking. The last challenge is the physical collisions among UAVs. UAVs may cause collisions since onboard sensors cannot obtain comprehensive flight states of neighboring UAVs due to the sensing limitations of sensors. Therefore, physical collisions need to be avoided among UAVs.

These challenges are closely related to the decision making of UAVs. With the limited communication resource of UAVs and the difficult constructions of fundamental infrastructures in many complicated tracking regions (e.g., forest), cloud computing [3] and mobile-edge computing (MEC) [4], [5] cannot provide real-time tracking decision. The decision latency can be reduced in a local centralized manner, where several UAVs are elected as cluster heads to control topology and tracking [6]. Whereas, the cluster heads cannot provide sufficient computing resources to schedule tracking nodes accurately and collaboratively. Thus, it is necessary to propose new solutions for the rapid decision making of UAVs. Leveraging a distributed pattern can be a potential solution to implement task assignments for low-latency computing.

In the distributed pattern, UAVs as agents can self-driven enforce self-driven data sensing and computing for rapid target trajectory prediction. However, the intelligent self-driven capacity can be restricted concerning an individual UAV. Swarm intelligence (SI) technology is an efficient solution to

Manuscript received April 24, 2021; revised May 14, 2021; accepted May 26, 2021. Date of publication June 3, 2021; date of current version December 23, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61731006, and in part by the Science and Technology Program of Sichuan Province, China, under Grant 2019YFH0007. *(Corresponding author: Supeng Leng.)*

Longyu Zhou and Supeng Leng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518000, China (e-mail: zhoulyfuture@outlook.com; spleng@uestc.edu.cn).

Qiang Liu is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324000, China (e-mail: liuqiang@uestc.edu.cn).

Qing Wang is with the Computer Science Department, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: qing.wang@tudelft.nl).

Digital Object Identifier 10.1109/JIOT.2021.3085673

enable UAVs to cooperate harmoniously. UAVs can share sensing information and tracking paths to facilitate collaborative tracking. The SI technology can be adapted to the distributed pattern smoothly to control sophisticated UAV swarm tracking. Previous studies on SI emphasize improving the swarm stability as well as robustness for discovering communicative coordination strategies [7]–[10]. The exploration time is unknown and undetermined, which is directly relevant to the response time of collaborative tracking. In this context, this method cannot be applied to UAV-MTT directly due to the limited energy carried by UAVs. The gap lacks to be filled to ensure the cooperative swarm tracking under the confined available energy.

In this article, we exploit UAV swarm cooperation for consecutive and accurate tracking, resembling nature systems with SI. We propose an intelligent UAV swarm-based cooperative tracking algorithm. The UAV swarm can establish ordered MTT without collisions in a limited amount of time. Wherein, a cooperative algorithm for multitarget trajectory prediction is proposed to improve the prediction accuracy. This algorithm is significantly primary for multitarget scenarios while ensuring a low execution latency. The main contributions are summarized as follows.

1) We propose a new UAV swarm-based cooperative tracking architecture to avoid the task execution process being hindered by ineffectual resource scheduling. Instead of independently executing self-sensing tasks of UAVs, our architecture can flexibly integrate their sensing and computing resources to facilitate rapid data collection and cooperative computing. Mutual improvement of sensing and computing capabilities is realized for efficient MTT. Specifically, a Lyapunov-based multiobjective optimization model is formulated to exploit the optimal task response strategy with the constraints of limited system budget and safe flight distances among UAVs.
2) Based on the collaborative task execution strategy, we first propose an energy consumption equilibrium scheme to maximize the flight time of the UAV swarm for consecutive MTT. This scheme can dynamically allocate UAVs to track the optimal targets with the tradeoff between flight distances and residual energy. Moreover, UAVs can cooperatively discover the optimal tracking paths to reduce the tracking time for the minimal probability of target loss.
3) We propose an intelligent UAV swarm-based cooperative tracking algorithm with the consideration of many-to-many tracking association among UAVs and targets. Unlike traditional solutions, our algorithm can synchronously share the valid sensing data and tracking status for rapidly reacting to the change of trajectories with the optimal association relation. Meanwhile, the time on exploring the optimal association policy is reduced while maintaining synchronous and consistent convergence.

The remainder of this article is organized as follows. Some latest research, including target tracking, multiagent, and SI algorithm, are discussed and summarized in Section II. The system model is presented in Section III. Section IV analyzes the MTT and formulates the tracking model. The UAV swarm-based cooperative tracking algorithm is introduced in Section V. The simulation results are presented in Section VI. Section VII concludes this article.

II. Related Work

Some researchers have studied mobile target tracking, cooperative multiagent, and SI in recent years. Some target tracking solutions are investigated to enhance the tracking utility. Besides, heuristic and evolutionary algorithms are investigated for highly efficient cooperative resource scheduling in tracking applications.

For accurate trajectory prediction, Jondhale and Deshpande [11] implemented GRNN as an alternative to obtaining location estimates of a single target moving in 2-D scenarios, which were then refined using the Kalman filtering (KF) framework. A performance metric is utilized in trajectory optimization to maximize target observation in a 2-D constrained environment in [12]. To reduce energy consumption on communications, Sun *et al.* [13] proposed to enable sustainable communication services in solar-powered UAV networks based on energy harvesting technologies. The effectiveness of the proposed scheme was verified by simulations. Instead of harvesting solar energy, we reduce UAVs' energy consumption by optimizing the associations among UAVs and targets in complicated MTT environments. Zhao *et al.* [14] proposed a UAV-assisted NOMA network, wherein UAV trajectories are optimized by the cooperation between UAVs and base stations. However, processing massive data in a wide range of monitoring areas is challenging.

Multiagent algorithms have been investigated for coping with the above-mentioned challenges with intelligent computation. Pei [15] proposed heterogeneous multiagent systems (HMASs) with disturbances and directed graphs for consensus tracking problems to improve the tracking accuracy. To address the nonlinear trajectory, a novel Deep Reinforcement Learning approach was proposed to control multiple UAVs with the ultimate purpose of tracking multiple first responders (FRs) in challenging 3-D environments in the presence of obstacles and occlusions in [16]. To reduce the probability of loss of targets, Shi *et al.* [17] investigated a novel localization system based on multiagent networks, where multiple agents serve as mobile anchors broadcasting their time-space information. Nevertheless, multiple target tracking is not considered in these studies, which is more challenging with multiobjective optimization.

SI algorithms are appropriate for MTT applications, which also pose challenges to swarm cooperation and huge data processing. Coping with the huge data, a random projection forest optimizing the K-nearest neighbors (KNNs) search algorithm was proposed for data mining and knowledge discovery [18]. In real-world scenarios, large-scale multiobjective optimization is challenging. Wang *et al.* [19] developed a UAV-based target tracking and recognition system using an intelligent gimbal system with the capabilities of accurate camera positioning, fast image processing, and multimodality

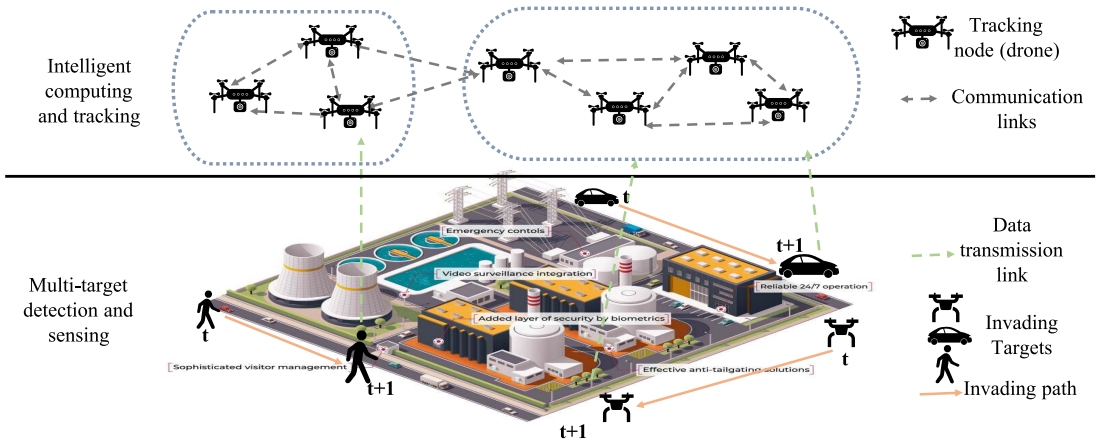


Fig. 1. Illustration of a UAV swarm-based MTT application scenario.

information fusion. To improve the aforementioned issue and achieve better tracking tasks, an intelligent collaborative navigation and control method was proposed in [20]. The authors proposed swarm-intelligence-based localization (SIL) and clustering schemes with UAVs for emergency communications [6], [21].

The above investigations of target tracking focus on tracking accuracy while ignoring the reality of dynamic topology. Previous multiagent algorithms can solve this issue with self-learning abilities. But these methods are difficult to be directly applied to MTT due to complicated interaction schemes. An intelligent UAV swarm-based cooperative tracking method can be a potential solution to implement accurate MTT in a complicated environment with uncertain obstruction.

III. System Model

We propose a UAV swarm cooperative tracking architecture for MTT. Task queuing and computing models are formulated to simultaneously execute sensing and computing operations for rapid tracking. The transmission and energy consumption models are formulated to analyze transmission latency among UAVs and the consumed energy.

The proposed architecture is illustrated in Fig. 1. We consider a scenario with K invading targets. The set of targets is $\mathcal{K} = \{1, 2, \ldots, K\}$. We assume that M UAVs, the set of UAVs $\mathcal{M} = \{1, 2, \ldots, M\}$, are leveraged to detect and drive these unlicensed invaders in a 3-D environment. In the proposed MTT architecture, different cooperative patterns on two layers are designed for accurate sensing and real-time computing. On the bottom layer, local cooperation is proposed for accurate data sensing. UAVs can capture diverse targets invading with different mobile trajectories by onboard sensors, such as cameras and ultrasonic. Based on the sensing data, UAVs can predict and analyze the trajectories of the targets. On the top layer, UAVs can cooperatively process the sensing data to rapidly track the targets. Cooperative computing is based on the task queuing model that we will introduce in the next section. The energy consumption model, incorporating flight, and task execution, are also formulated to optimize the processing of target tracking.

A. Task Queuing Model

In the process of tracking, UAVs use onboard sensors to capture the invading targets that are within their sensing ranges. These sensors, such as depth cameras, infrared sensors, and ultrasonic sensors, can rapidly collect different data that will be sent to control process units for further processing. The data, namely computing tasks, contains surrounding environmental information, current states and positions of the targets, and states of neighboring UAVs. However, the intensive task workloads may cause high-latency computing due to the limited computing capability on an individual UAV. Conversely, some UAVs might quickly obtain computing results with their narrow sensing horizon. To enable low-latency computing for the UAV swarm system, we design a collaborative computing scheme to reduce the computing time on MTT.

We denote $Q_i(t)$ as the computing task loads at UAV i in time slot t, where $t \in \{1, 2, \ldots, T\}$. It is consisted of three parts.

1) The current sensing data $S_i^k(t)$ acquired by on-board sensors of the UAV i.
2) The data $y_{i,j}^k(t)$ that is transmitted by the UAV j for the cooperative computing.
3) The remaining data $Q_i^k(t-1) - y_i^k(t-1)$ from the last time slot $t-1$, where $y_i^k(t-1)$ denotes that the task loads are computed by the UAV i at the time slot $t-1$.

The $S_i^k(t)$ incorporates textual data collected by the ultrasonic and image data acquired by the camera. The textual data can be utilized to predict trajectories of targets while the image information can be only used to discover invading targets. Moreover, all the sensing data can be represented as digital values. As shown in Fig. 2, the UAV i senses $S_i^k(t)$ task loads from the target k in the time slot t. Wherein, $x_i^k(t)$ task loads can be computed by the UAV i and $y_{i,j}^k$ task loads can be offloaded to the nearby resource-available UAV j. The UAV j will send the computed results to UAV i (red arrows in Fig. 2). The computing results are utilized to predict the trajectories of targets. Based on this cooperative computing mechanism, UAV i can efficiently complete the computing process. An indicator $a_{i,j}$ is used to represent the cooperative computing offloading from UAV i to UAV j. There exists

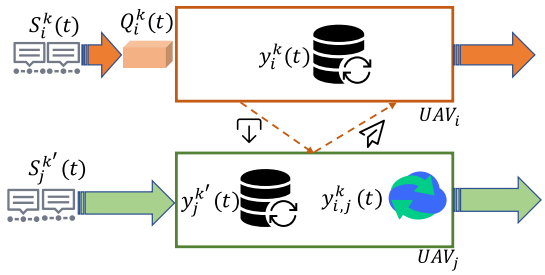


Fig. 2. Illustration of a UAV swarm-based cooperation computing.

cooperative computing offloading between UAV i and UAV j when $a_{i,j} = 1$; otherwise, $a_{i,j} = 0$.

The queue updates the queue size after UAV i acquires the sensing data of target k, which can be represented as

$$Q_i^k(t+1) = \left[Q_i^k(t) - \sum_{j=1}^{M} a_{i,j} y_{i,j}^k(t) \right]^+ + S_i^k(t) \tag{1}$$

where $[x]^+ \triangleq \max\{0, x\}$.

B. Transmission Model

Next we present the transmission model which is used to analyze the wireless communication data rates among UAVs. Sensing data is shared with neighboring UAVs for efficient data processing. We assume that Wi-Fi 6, the latest Wi-Fi technology based on the IEEE 802.11ax standard [22], is used by the UAVs for data transmission. The model of the data rate between UAV i and UAV j can be written as follows:

$$r_{i,j} = \sum_{l}^{L} B_{i,j}(l) \log_2\left(1 + \frac{P_i(l) g_i(l)}{\sigma^2}\right) \tag{2}$$

where L is the number of subcarriers, $B_{i,j}(l)$ is the transmission bandwidth of subcarrier l between UAV i and UAV j, $P_i(l)$ and $g_i(l)$ are the transmission power and power gain of subcarrier l, $g_i(l) \sim f(x|v, \delta)$ is a standard rice distribution with $v = 0$ and $\delta = 0.5$, and $\sigma \sim N(0, \delta^2)$ is the zero mean Gaussian random variable with a standard deviation of δ.

C. Energy Consumption Model

The energy consumption model is formulated to quantify the change of involved energy for UAVs during MTT, which incorporates computing consumption, transmission consumption, and flight consumption. The computation model is formulated to optimize the whole system energy consumption at each time slot. For time slot t, we consider the local computing for executing y_i bits at each UAV $i \in \mathcal{M}$, where $y_i = \sum_{k=1}^{K} y_i^k(t)$. Let b_i denote the number of center process unit (CPU) cycles acquired for computing an input bit. The total number of CPU cycles for y_i can be $b_i y_i$. Based on dynamic voltage and frequency scaling (DVFS) technology [23], UAV i can adjust the CPU working frequency $f_{i,w}$ for each cycle w to control the energy consumption, where $f_{i,w} \in (0, f_{i,w}^{\max})$, and $f_{i,w}^{\max}$ is the maximal CPU frequency of UAV i. The execution time of local computing at UAV i can be expressed as $\sum_{w=1}^{b_i y_i} (1/f_{i,w})$. Due to the fact that UAV is a kind of low-power device, we do not consider the impact of temperature on CPU working frequency. From [24], we know that power consumption is proportional to the cubic frequency. The local energy consumption at UAV i can be given by

$$E_i(t) = \sum_{w=1}^{b_i y_i} \kappa_i f_{i,w}^2 \tag{3}$$

where κ_i is the efficient capacitance coefficient that depends on the clip characteristic of UAV i.

In the collaborative computing mode, for each time slot t, the number of CPU cycles of UAV j can be represented as $\sum_{w=1}^{b_j y_j^c} (1/f_{j,w})$, where y_j^c is the received task load (in bits) from other UAVs, and $y_j^c = \sum_{i=1, i \neq j} y_{i,j}(t)$. The relevant energy consumption can be written as

$$E_j^c(t) = \sum_{w=1}^{b_j y_j^c} \kappa_j f_{j,w}^2. \tag{4}$$

At each time slot t, the UAV i can transmit $y_i^c(t)$ task loads (in bits) to other UAVs for cooperative computing. Assume that channel quality for each UAV is the same and constant at each time slot t, the transmission rate $r_{i,j}, \forall j \in \mathcal{M}, i \neq j$ between any two UAVs is the same. The transmission power p_i and a constant circuit power p_i^c (by filter, etc) are considered to define the total transmission power. The transmission energy consumption at UAV i is represented as

$$E_i^r(t) = \left(p_i + p_i^c\right) \frac{y_i^c(t)}{r_{i,j}} + E_i^b. \tag{5}$$

It is noted that the energy consumption for transmitting the computing results among UAVs is very low, which can be defined as a constant energy consumption E_i^b.

Except for the energy consumption caused by computing and communication, flight energy consumption also needs to be analyzed. The consumption can be directly related to flight paths. The flight speeds are affected by the speeds of the invading targets. The paths can be shortened based on the selection of the tracking targets. It is noteworthy that the energy consumption caused by UAV self-rotation and patch angle is ignored for simplifying analysis. The energy consumption of a unit path of the UAV i is denoted as $E_i^f(t) = \varphi(v_{i,t})$, where $\varphi(x)$ is a mapping function with nonlinear character [25]; $E_i^f(t)$ is variable in different time slots due to the diverse speeds of the targets. Zeng *et al.* [25] have shown that flight energy consumption is relevant to flight speed, and the UAV flying energy consumption is more than that of the hovering.

IV. Problem Formulation

Based on the proposed swarm cooperative tracking architecture, an integer programming model is formulated to analyze the tracking performance using Lyapunov optimization. The model aims to jointly optimize the system energy consumption and prediction errors of trajectories, with the constraints of execution latency and quality of communication.

A. System Latency and Flight Distance Analysis

In MTT, it is necessary to continually maintain low-latency execution response due to the high-speed mobility of targets. In this case, we give an average accumulative execution latency constraint based on the swarm collaboration communication method in [26]. Specifically, the communication quality among UAVs may be frequently unstable due to the dynamic trajectories of the UAVs. To rapidly transmit data among UAVs, UAVs can process parts of data in advance and then transmit the relevant computing results and the residual data to neighboring UAVs. The operation can reduce the transmission workload for low-latency interaction due to the negligible size of the computing results. The latency constraint is represented in (6), shown at the bottom of the page, where $[y]^* \triangleq \max\{1, y\}$; T_k is maximum tolerated time of the target k; $\eta_{i,k}$ is an indicator, where $\eta_{i,k} = 1$ denotes that UAV i can sense the target k. Further, to avoid physical collisions among UAVs during the tracking process, we also take into account the safe flight distance between any two UAVs. Let $d_{i,j}(t)$ denote the distance between UAV i and j at time slot t. The constraint that $d_{i,j}(t) \geq \varpi_{\min}$ is ensured, where $t \in T$ and $i, j \in \mathcal{M}$.

B. System Energy Consumption Analysis

How to maintain consecutive target tracking is always intractable for battery-powered UAVs. For UAVs, a large part of the energy can be consumed based on flying operations and frequent data communications. In this regard, each UAV can utilize its partial computing resource (budget (Joules)) to execute the current tasks at the time slot t. We can control the budget of each UAV to avoid excessive energy consumption. Let $\upsilon_i(t)$ (Joules) denote the budget of the UAV i at the time slot t that can be spent on computing, flying, and transmitting data. The average cumulative budget Υ_i (Joules) is defined to restrict the energy consumption of UAV i. We can give the corresponding energy consumption constraint in (7), shown at the bottom of the page.

Equation (7) gives us some guides of system energy consumption optimization. We can draw that the consumption of UAV i can be reduced with the increase of the number of neighbor UAVs. However, the number of neighbors can not infinitely increase due to the limited number of UAVs. In this case, we can always discover resource-available neighbor UAVs when the computing resource of UAV i is insufficient.

C. Trajectory Prediction Analysis

For predicting the trajectories of mobile targets, the popular extend Kalman filter (EKF) is efficient. The main idea of EKF is similar to the traditional KF, in which the nonlinear motion is approximated to linear motion based on the Taylor series. There exists two estimation processes: 1) *prediction* and 2) *update*. For UAV i, in the *prediction* stage, the coordinate of UAV i at time slot $t+1$ is formulated as $x_{t+1|t.} = Fx_t + \omega_t$, where F is transfer matrix and ω_t is standard Gaussian white noise. The prediction evaluation equation is defined as $P_{t+1|t.} = F \times P_t \times F^T$. In the *update* stage, a covariance $S_{t+1} = H_{t+1} P_{t+1|t.} H_{t+1}^T$ is defined which is a convex function. Based on the covariance function, we can acquire the Kalman gain K_{t+1}, which is given by $K_{t+1} = P_{t+1|t.} \times H_{t+1}^T \times S_{t+1}^{-1}$, where H is the measurement matrix. The updated coordinate of UAV i is regarded as real coordinate value which is represented as $x_{t+1} = x_{t+1|t.} + K_{t+1}\widetilde{y}$, where $\widetilde{y}$ is measurement residual (i.e., the difference of the measurement value and the evaluation value). Unfortunately, EKF accumulates the prediction error when the tracking time increases. To address this challenge, we propose a UAV swarm-based cooperative prediction algorithm which will be detailed in Section V. The error of UAV i at time slot t+1 is defined as $\alpha_i^k(t+1) = (x_{t+1} - x_{t+1|t.})$. The accumulative mean error expectation is given as

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \left(\sum_{i=1}^{M} \alpha_i^k(t+1) \right) \leq \Lambda_k \tag{8}$$

where Λ_k is the maximal toleration error.

D. Optimization Objective Formulation

We employ the Lyapunov drift-plus-penalty framework [27] to formulate a feasible optimization objective to jointly optimize the system energy consumption, the prediction accuracy, and the physical collision avoidance. Based on the framework, we tradeoff the low energy consumption and the low execution latency by importing a virtual queue. This virtual queue can push UAVs to execute tasks of input queues on schedule with the penalty scheme while avoiding extreme energy consumption. For simplicity, we use $A_k(t) = \sum_{i=1}^{M} \alpha_i^k(t)$ and $B_k(t)$ replace the left-hand side of (6) and (8), respectively. The virtual queues are represented as

$$G_k(t+1) = G_k(t) + A_k(t) - \Lambda_k \tag{9}$$

$$H_k(t+1) = H_k(t) + B_k(t) - T_k \tag{10}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \left(\frac{\sum_{i=1}^{M} \sum_{w=1}^{b_i y_i} \frac{1}{f_{i,w}}}{\left[\sum_{i=1}^{M} \eta_i^k\right]^*} + \frac{\sum_{j=1}^{M} \left(\sum_{w=1}^{b_j y_j^c} \frac{1}{f_{j,w}} \right)}{\left[\sum_{j=1}^{M} a_{i,j}\right]^*} + \frac{\sum_{j=1}^{M} \frac{\sum_{k=1}^{K} y_i^c(t)}{r_{i,j}}}{\left[\sum_{i=1}^{M} a_{i,j}\right]^*} \right) \leq T_k \tag{6}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \left[E_i(t) + E_i^f(t) + \frac{\sum_{j=1}^{M} \left(E_{i,j}^c(t) + E_{i,j}^r(t) \right)}{\left[\sum_{j=1}^{M} a_{i,j}\right]^*} \right] \leq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \upsilon_i(t) = \Upsilon_i \tag{7}$$

where $G_k(t)$ and $H_k(t)$ are virtual queues; $A_k(t)$, and $B_k(t)$ are regarded as input flows (i.e., will be served); Λ_k and T_k represent digestive data (i.e., have been served).

The Lyapunov optimization model is formulated as

$$L(t) = \frac{1}{2}[L_1(t) + L_2(t)] \tag{11}$$

where

$$L_1(t) = \sum_{i=1}^{M}\sum_{k=1}^{K} \omega_k^2 Q_i^{k^2}(t) \tag{12}$$

$$L_2(t) = \sum_{k=1}^{K} \omega_k^2 \Big(G_k^2(t) + H_k^2(t)\Big) \tag{13}$$

where $L(t)$, the drift function, is viewed as a measure of the total queue backlogs, in which $L_1(t)$ and $L_2(t)$ are real and virtual queue backlogs, respectively; ω_k is a weight value of target k, which is limited in the range [0, 1].

The optimization model can be expressed as

$$\mathbb{F}(t) = \Delta(\mathbb{L}(t)) + V\mathbb{E}\sum_{i=1}^{M}\Bigg[E_i(t) + E_i^f(t) + \sum_{j=1}^{M}\Big(E_{i,j}(t) + E_{i,j}^r(t)\Big)\Bigg] \tag{14}$$

where V is a set of weight values to tradeoff the queue length and the system energy consumption, and

$$\Delta(\mathbb{L}(t)) \triangleq \mathbb{E}[L(t+1) - L(t)]. \tag{15}$$

Therefore, the optimization model is represented as

$$P1: \min \ \mathbb{F}(t) \quad \text{s.t.} \begin{cases} C1: r_{i,j} \le r_{i,j,\max} \\ C2: \sum_{j=1}^{M} a_{i,j} = 1 \quad \forall t \in T \\ C3: a_{i,j}, \eta_i^k \in \{0, 1\} \\ C4: d_{i,j}(t) \ge \varpi_{\min} \end{cases} \tag{16}$$

where $C1$ is the transmission rate constraints, $C2$ and $C3$ are integer constraints which guarantee that each UAV only receives a task from other UAVs at each time slot t, and $C4$ is the flight distance constraint.

V. Multitarget Tracking With UAV Swarm

In this section, we aim for accurate and consecutive target tracking. On one hand, considering multiple high-speed mobile targets, we design a cooperative prediction algorithm to obtain high prediction accuracy. On the other hand, we propose an intelligent UAV swarm-based cooperative tracking algorithm to improve tracking efficiency while avoiding physical collisions.

A. Multiple Mobile Targets Prediction

Recalling the mentioned accuracy constraint, a swam cooperative prediction algorithm is proposed based on the particle swarm optimization (PSO) framework. UAVs that have detected targets run the EKF algorithm and obtain target coordinates. Based on each target coordinate, we design a square area as

Algorithm 1 Cooperative Multiple Targets Prediction

// **Definition:** $k_{\max} = 100$.

Input: Velocity $V_i(k)$; Inertia parameter r; Accelerate coefficients: q_1, q_2; Random values: r_1, r_2; D dimensional exploring region; population size N.

Output: The predicted locations of targets.

```
Initialize EKF parameters
for each UAV has detected targets do
    Predict the target's coordinate using x_{t+1|t} = Fx_t + ω_t
    Update the coordinate using P_{t+1|t+1}
    Calculate Kalman gain K_{t+1} and target's location
end for
Identify a small searching region
Deploy N particles to explore this region randomly
Initialize the fitness function: g(X_{i,d}) = 0, X_d^gb = ∞
for each particle i ∈ N do
    for each dimension d ∈ D do
        Compute fitness functions g(X_{i,d}) and g'(X_{i,d})
        X_d^gb = arg min_{X_d} g(X_{i,d})
        X_d^pb = arg min_{X_d} g'(X_{i,d})
    end for
end for
while k ≤ k_max do
    for each particle i ∈ N do
        for each dimension d ∈ D do
            Compute velocity V_{i,d}(k+1) using Eq. (17)
            Update the coordinate using Eq. (18) based on the
            limitation of exploring region
            Compute fitness functions g(X_{i,d}) and g'(X_{i,d})
        end for
    end for
    k=k+1
end while
```

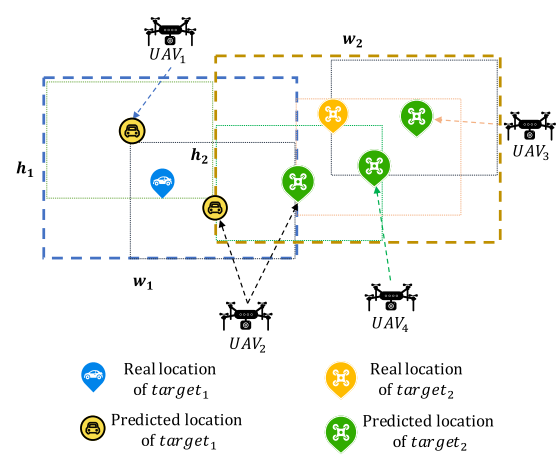


Fig. 3. Illustration of cooperative prediction for multiple targets.

particle exploration spaces. As shown in Fig. 3, UAV_1 and UAV_2 detect the $target_1$ while UAV_2, UAV_3, and UAV_4 detect the $target_2$. All the involved UAVs make preliminary prediction estimations based on the EKF, and then obtain final prediction results with the aid of the heuristic exploration. The specific pseudocode is summarized in Algorithm 1. It is noteworthy that

an individual UAV may predict multiple targets (like UAV_2 of Fig. 3). Compared with the standard version of PSO, the run time by using our proposed scheme can be reduced with the h_i height and w_i width exploration space, where $i \in \mathcal{K}$. Besides, each dimension selects the corresponding optimal position and velocity from different samples.

Specifically, particles are randomly deployed in the initialization. Position and velocity of the particle i at the dimension d are updated as

$$V_{i,d}(k+1) = rV_{i,d}(k) + q_1 r_1\Big(X_d^{\mathrm{gb}} - X_{i,d}(k)\Big) + q_2 r_2\Big(X_d^{\mathrm{pb}} - X_{i,d}(k)\Big) \tag{17}$$

$$X_{i,d}(k+1) = X_{i,d}(k) + V_{i,d}(k+1) \tag{18}$$

where r is the inertia parameter, q_1 and q_2 are acceleration coefficients, and r_1 and r_2 are random values selected from the range [0, 1]; X^{gb} and X^{pb} are global and local optimization indicators, which are updated by the fitness functions. The global and local fitness functions are defined as

$$X_d^{\mathrm{gb}} = \arg\min_{X_d} g\big(X_{i,d}\big) = \sum_{j=1}^{M} \vartheta_j \Big\| X_{i,d} - h_j \Big\|_2 \tag{19}$$

$$X_d^{\mathrm{pb}} = \arg\min_{X_d} g'\big(X_{i,d}\big) = \sum_{j=1}^{M} \vartheta_j' \Big\| X_{i,d} - h_j' \Big\|_2 \tag{20}$$

where $\vartheta_j, \vartheta_j' \in \{0, 1\}$ denotes whether UAV j senses targets or not; h_j and h_j' are global prediction value and local prediction value. Therefore, the proposed Algorithm 1 can meet the time-sensitive requirement and accurate trajectory prediction.

B. Intelligent UAV Allocation for Multiple Targets Tracking

The Lyapunov draft-plus-penalty algorithm can only obtain the draft-plus-penalty upper bound [27]. To reach the lower bound, we propose an intelligent UAV swarm-based cooperative tracking algorithm.

As shown in Fig. 4, we design our algorithm based on the actor–critic framework, including a target network and a Q-network. A UAV is treated as an agent in the actor–critic framework. We quantify the target tracking process as a stochastic game (SG) problem which is described as a tuple $\{S, \{A_i\}, \mathcal{T}, \{u_i\}\}$, where S is the state space of each agent; $\{A_i\}$ is the action space of the agent i; $\mathcal{T}$ is state transfer function; and $\{u_i\}$ is reward function. The state space is expressed as $S = \{B_{i,j}, Q_i^k, E_i, \Lambda_k, T_k, \Gamma_k\}$. The actions space of the agent i is composed as $A_i = \{x_i, v_i, g_i\}$, where x_i is current position; v_i is moving velocity; g_i is pitch angle. The deterministic policy transition function $\mathcal{T}{:}S \times A_i \to S$. Due to the fact that an agent can only observe its own local information, the observation of each agent is given to neighboring agents for assistant evaluation, so the reward of the agent i is $r_i^t(s_t, a_t) = (1/[u_i(t)]) \sum_{j=1}^{u_i(t)} [\Delta E_j + \Delta L_j + \Delta I_j]$, where $[\Delta *] \equiv E_j^{t-1} - E_j^t + L_j^{t-1} - L_j^t + I_j^{t-1} - I_j^t$, L_j and I_j are the execution latency and the tracking accuracy, respectively. To balance the energy consumption at each agent, we propose the energy consumption equilibrium method to ensure cooperative swarm actions. The rewards are updated by (21), shown at the bottom of the page, where the temporal smoothed rewards $e_j^t(s_j, a_j) = \lambda e_j^{t-1}(s_j, a_j) + r_j^t(s_j, a_j)$. The parameters α_i and β_i are set to 5 and 0.05, respectively. The policy function training in the Q-network is used to evaluate the current tracking action. The optimal policy iteration expression is given based on the Bellman equation

$$Q_i^*(s_t, a_t) = u_i(s_t, a_t) + \lambda \max_{\mu_{\theta,\theta^-}} Q_i^*\big(s_{t+1}, \mu_{\theta,\theta^-}(s_{t+1})\big) \tag{22}$$

where λ is a discount parameter; μ_{θ,θ^-} is a policy expression of the policy network (i.e., $a_{t+1} = \mu_{\theta,\theta^-}(s_{t+1})$); θ^- is the policy parameters of other agents as the auxiliary variable except the agent i.

The object that maximizes the expected reward is denoted as $J_i(\mu_\theta) = \mathbb{E}_{s\sim\rho_{\mu_{\theta_i}}^{\mathcal{T}}}[u_i(s, \mu_{\theta_i}(s))]$, where $\rho_{\mu_{\theta_i}}$ is the discount state distribution under the policy μ_{θ_i} and the transition function $\mathcal{T}$; $\rho_{\mu_{\theta_i}} \triangleq \int_S \sum_{t=0}^{\infty} \lambda^{t-1} \psi(s_{t+1} = \mathcal{T}_{\mu_\theta}(s)|s)ds$; $\psi(s)$ is the state probability distribution function. The object of total agents rewards can be denoted by $J(\mu_\theta)$ as follows:

$$J(\mu_\theta) = \mathbb{E}_{s\sim\rho_{\mu_{\theta_i}}^{\mathcal{T}}}\left[\sum_{i=1}^{M} u_i\big(s, \mu_{\theta_i,\theta^-}(s)\big)\right]. \tag{26}$$

All the tracking actions need to be directed toward the maximum rewards. The policy gradient of the object $J(\theta)$ is

$$u_i^t(s_t, a_t) = r_i^t(s_t, a_t) - \frac{\alpha_i}{M-1}\sum_{j\neq i}\max\Big(e_j^t\big(s_j, a_j\big) - e_i^t(s_i, a_i), 0\Big) - \frac{\beta_i}{M-1}\sum_{j\neq i}\max\Big(e_j^t(s_i, a_i) - e_j^t\big(s_j, a_j\big), 0\Big) \tag{21}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\rho_{\mu_{\theta_i}}^{\mathcal{T}}}\left[\sum_{i=1}^{M}\sum_{j=1}^{M} \nabla_\mu Q_i^{\mu_{\theta_i}}\Big(s_j, \mu|_{\mu=\mu_{\theta_i,\theta^-}(s_j)}\Big) \cdot \nabla_{\theta_i}\mu\big(s_j, \mu_{\theta_i,\theta^-}\big(s_j\big)\big)\right] \tag{23}$$

$$L_Q(\theta) = \mathbb{E}_{s\sim\rho_{\mu_{\theta_i}}^{\mathcal{T}}}\left[\sum_{i=1}^{M}(u_i(s_t, \mu_\theta(s_t))) + \lambda Q_i^{\theta_i'}\big(s_{t+1}, \mu_{\theta_i,\theta^-}(s_{t+1})\big) - Q_i^{\theta_i}\big(s_t, \mu_{\theta_i,\theta^-}(s_t)\big)\right] \tag{24}$$

$$\nabla_{Q_\theta} L_Q(\theta) = \mathbb{E}_{s\sim\rho_{\mu_{\theta_i}}^{\mathcal{T}}}\left\{\sum_{i=1}^{M}\Big[(u_i(s_t, \mu_\theta(s_t))) + \lambda Q_i^{\theta_i'}\big(s_{t+1}, \mu_{\theta_i,\theta^-}(s_{t+1})\big) - Q_i^{\theta_i}\big(s_t, \mu_{\theta_i,\theta^-}(s_t)\big)\Big] \cdot \nabla_{\mathrm{d}Q_{\theta_i}} Q_i^{\theta_i}\big(s_t, \mu_{\theta_i,\theta^-}(s_t)\big)\right\} \tag{25}$$

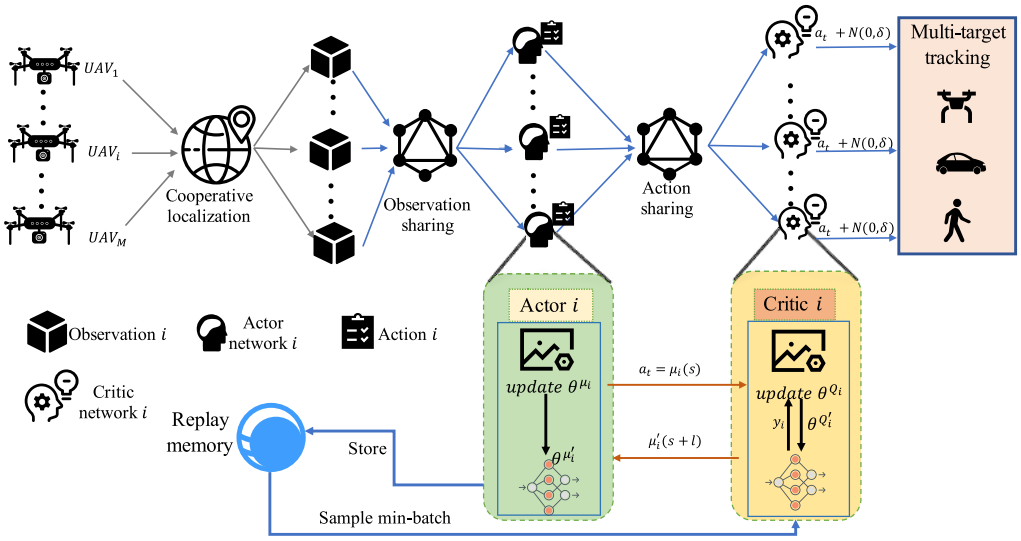


Fig. 4. SI-based cooperative scheduling for MTT.

Algorithm 2 UAV Swarm-Based Cooperative Tracking

// **Definition:** $\gamma = 0.99$.

Input: Network parameters Q_θ, μ_θ, $Q_{\theta'}$, $\mu_{\theta'}$; soft update weight γ; replay buffer **F**; system parameters.

Output: The tracking decisions.

```
Predict the trajectories of targets using Algorithm 1
for each episode in all the rounds do
    Initialize Gaussian Noise for action exploration
    Set an initial policy μ and receive initial observation
    for each time slot t do
        for each agent i ∈ M do
            a_{i,t} = μ_{θ_i}(s_t) + GN_t
            Receive each reward R_{i,t}
            Store state-action pairs into replay memory F
            Sample a min-batch of R transitions from F
            for each sample in R do
                Execute the energy consumption equilibrium
                using Eq. (21)
                Compute Q-value using Eq. (23)
            end for
            Compute the loss gradients: Σ_{r=1}^{R} ∇_{μ_θ} L(θ)
            Compute the rewards gradients: Σ_{r=1}^{R} ∇_{Q_θ} J(θ)
            Update the network based on the SGD
            Update the network parameters: μ_{θ'}, Q_{θ'}
        end for
    end for
end for
```

computed in (23), shown at the bottom of the previous page. The derivation process of (23) is similar to that in [28]. In each episode, the tracking action is estimated by parameter Q^θ in the Q-network. The parameter is optimized by the loss function $L_Q(\theta)$ shown in (24), at the bottom of the previous page. The gradient of the loss function is given by (25), shown at the bottom of the previous page. The total rewards and the convergence states are obtained based on the stochastic gradients descent (SGD) method. The algorithm is detailed in Algorithm 2.

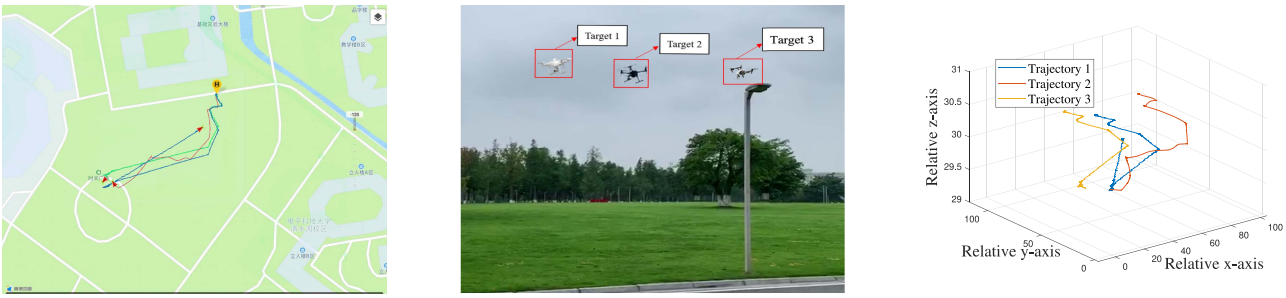
Algorithm Complexity Analysis: In our proposed architecture, UAVs can cooperatively track multiple targets based on their optimal paths from self-driven learning. Furthermore, UAVs can change the current tracking targets based on information sharing for the maximal reward. Comparing with existing algorithms, the prediction accuracy is significantly improved by UAV cooperation. For the complexity of our algorithm, first, the cooperative prediction algorithm can consume $O(k_{\max}ND)$ time with nested loops, where N and D are the numbers of particles and dimensions. Second, we obverse that tracking actions can be outputted once with $O(1)$ for each iteration. The complexity of the primary network can be calculated by matrix inversion operation with $O(k(\theta))$, where $k(\theta)$ is a function with the number of hidden layers θ. The complexity of the whole algorithm can be represented as $O(k_{\max}NDk(\theta))$. This complexity is not higher than that of traditional deep reinforcement learning, while our proposed algorithm can improve the multitarget prediction accuracy.

VI. Performance Evaluation

In this section, we present our evaluation results. We build an MTT simulator based on TensorFlow. To emulate a practical tracking scenario where all the invading targets could have high moving speeds, we collect measurement data with three drones that fly with predefined trajectories in outdoor boundary-free areas. The flight records are cached by mobile targets that are equipped with global position system (GPS). We implement the proposed algorithm based on multiagent determined deep policy gradient (MA-DDPG) architecture. The main parameters we use are summarized in Table I.

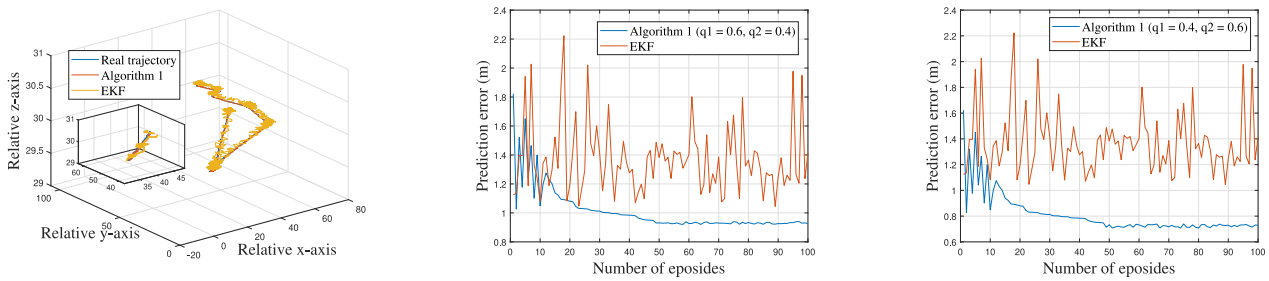
We use four benchmarks for the performance comparison.

1) *Matched Deep Q-Network [29]:* This scheme uses a deep Q-network (DQN) framework for MTT.
2) *Noncooperative:* Compared to our scheme, the joint tracking is executed based on the MA-DDPG architecture but states and actions are not shared among UAVs.



(a) (b) (c)

Fig. 5. Three target flight trajectories, collected by DJI Phantom 4, DJI Matrice 210, and DJI Mavic, respectively, are represented in (a), the flight starting point marked as H, the endpoint marked as red aircraft, and flight trajectory marked by distinguishable colors are displayed in the GaoDe Map. The real-world scenario selected in the University of Electronic Science and Technology of China (UESTC) is shown in (b). The three target flight trajectories data is transferred and displayed in a 3-D Cartesian coordinate for convenient data processing, which is shown in (c).



(a) (b) (c)

Fig. 6. Prediction errors comparison between EKF and our proposed algorithm. (a) Prediction comparison for Trajectory 1. (b) Prediction error for Trajectory 2. (c) Prediction error for Trajectory 3.

TABLE I
SIMULATION PARAMETERS

Parameter description	Value
$\varpi_{\min}$	3 m
Number of targets	3
Number of UAVs	8
Transmission bandwidth among agents	[50 MHz, 100 MHz]
Transmission power of each agent	33 dBm
The average moving velocity of each agent	72 km/h
Average moving velocity of each target	64 km/h
The path of target tracking	600 m
The maximum size of transmission data	100 MBytes
Required deadline of targets	[0,5] seconds

3) *Greedy:* This scheme explores the shortest physical path to associate UAVs and targets based on MA-DDPG.
4) *Random:* Decisions on tracking the targets are randomly made based on MA-DDPG without self-learning.

A. Acquirement of Flight Trajectories of Targets

Fig. 5 shows the practical flight trajectories where the flight data is obtained by GPS. All the flight trajectories are recorded in Fig. 5(a) which are obtained by the DJI Phantom 4, DJI Matrice 210, and DJI Mavic. Fig. 5(b) provides the flight real-world environment in UESTC. In Fig. 5(c), the GPS flight data is transferred into the Cartesian coordinate system using the seven-parameter method [30].

B. Numerical Analysis of Tracking Performance

Fig. 6 presents the comparison between the EKF algorithm and our proposed algorithm. In Fig. 6(a), the green line, the yellow line, and the blue line present the real trajectories, our prediction results, and the EKF prediction results, respectively. The robustness of our algorithm can be confirmed by comparing it to the EKF algorithm. Fig. 6(b) and (c) show prediction errors comparison with respect to the iteration. The tendency can be converged by slight oscillations which can be resulted from the exploration toward some wrong directions. In summary, the prediction accuracy can be enhanced based on different learning parameters in our algorithm. The EKF that is not capable of the nonlinear prediction with Taylor series approximation can cause higher errors. Fig. 7 provides the performance of prediction error under different acceleration factors. We can draw that different acceleration factors result in different prediction results in the proposed algorithm. Wherein, the performance is excellent when $q_1 = q_2$. The acceleration factor can conduct the particle exploring direction which can directly affect the exploration results. Our proposed scheme can theoretically obtain zero prediction error when particles reach the optimal point during exploration. However, the gap that the prediction error between the optimal one and the proposed scheme still exists due to uncertain exploring directions. In the numerical analysis, our algorithm can improve the accuracy by about 60% comparing with the EKF algorithm, which is a constructive enhancement in MTT. Besides, our algorithm is robust and can cope with those moving targets with significant flight turns.

Fig. 8 shows the convergent tendency of all the agents, where the tendency is reduced with the increasing of iterations. We have the following observations from this figure. First, our proposed algorithm can make the training loss maintain

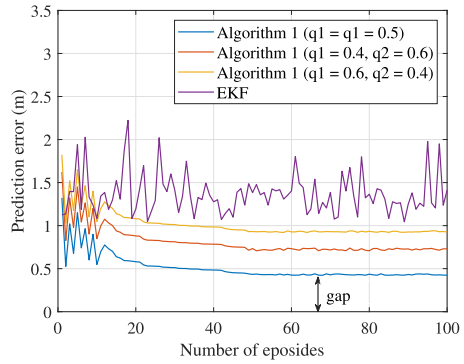


Fig. 7. Average error comparison versus iterative number.

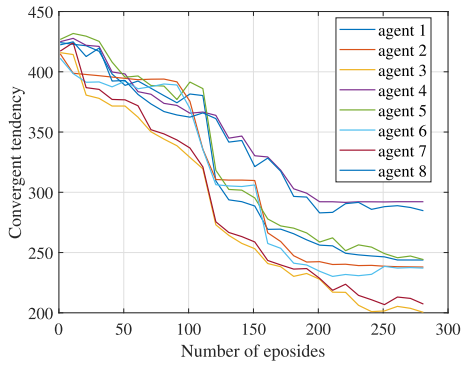


Fig. 8. Convergence tendency of each agent versus iterative number.

a stable convergence. Moreover, the proposed energy equilibrium scheme makes all the agents reach approximately the same convergent status. In this case, an efficient tracking decision is obtained with the maximal system reward.

In Fig. 9, we compare the cumulative average system energy consumption. The decrease in the system energy consumption is not significant in the early stages of our proposed algorithm. However, our algorithm obtains the optimal system energy consumption compared with the four benchmarks when the episode increases. For the simulation results, we observe that our proposed algorithm reduces the system energy consumption of the four benchmarks (matched DQN, noncooperative, greedy, and random) by 19.2%, 22.8%, 28.9%, and 32.5%, respectively. This result shows that our algorithm can boost the systematic energy efficiency to meet consecutive target tracking.

In Fig. 10, the performance of physical collision is presented where the results are obtained by simulating the interaction among agents. The physical collision is estimated scientifically following the minimal tolerated distance $\varpi_{\min}$, and the task queuing length is recorded as the episode increases. We can observe that all the schemes have outgoing performances with which the collision frequency is reduced. However, the collision avoidance cannot be solved in the four benchmarks during tracking. By contrast, our proposed algorithm can achieve collision-free when the number of episodes reaches about 70. This implies that cooperative interaction among UAVs not only can reduce energy consumption but also can reduce the probability of physical collisions among UAVs.

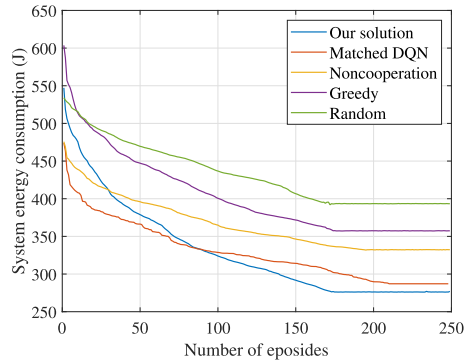


Fig. 9. System energy consumption versus iterative number.

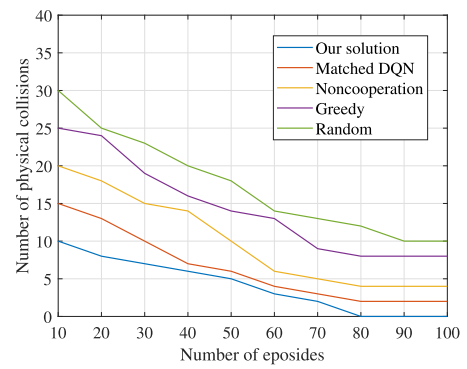


Fig. 10. Number of physical collisions versus iterative number.

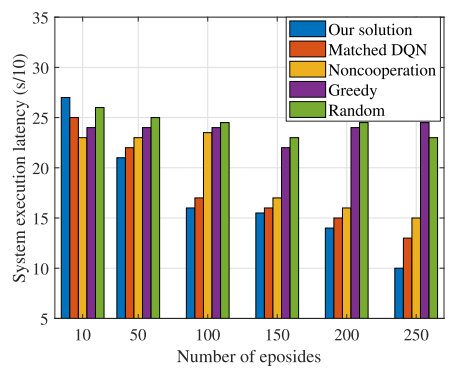


Fig. 11. System response latency versus iterative number.

Fig. 11 provides the comparison of our system execution latency with that of the four benchmarks. We can draw the following observations. First, all the types of execution times are reduced with the increasing number of episodes except for the random scheme. Besides, the proposed algorithm performs better than other benchmarks in terms of convergent speed. This implies that the integration of sensing and computing resources can reduce system execution latency. The cooperative architecture can accelerate the execution response for real-time tracking. In the numerical analysis, the proposed algorithm can reduce the latency of the matched DQN scheme, the noncooperative scheme, the reedy scheme, and the random scheme by 30%, 41.1%, 56.5%, and 60%, respectively.

In Fig. 12, we provide the comparison of cumulative average energy consumption. To reduce energy consumption, each agent should consume as little energy as possible to execute

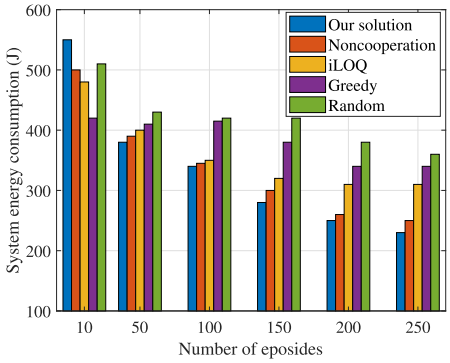


Fig. 12. System energy consumption comparison with the iLOQ scheme.

their tasks. The existing benchmark is introduced to compare with our algorithm. The benchmark is named iterative linear-quadratic-Gaussian (iLQG) with the distributed alternating direction method of multipliers [31]. As a whole, the system energy consumption is reduced gradually with the increase of the episodes. Our proposed algorithm can reduce 28.5% of the energy consumption of iLQG. This result confirms that our algorithm is beneficial to target tracking with uncertain mobile trajectories.

VII. Conclusion

This article investigated the MTT problem with the execution delay and energy consumption. Based on Lyapunov optimization, we formulated a multiobjective optimization model which is subjective to the prediction accuracy, execution delay, and physical collision constraints. Furthermore, we developed an intelligent UAV swarm-based cooperative tracking algorithm. The algorithm can adapt to complicated environments and dynamic task assignments for real-time and consecutive tracking. Compared with existing benchmark schemes, our evaluation results showed that our proposed algorithm can reduce the system energy consumption and the system execution latency while improving the trajectory prediction accuracy.

References

[1] S. Xiao, T. Li, C. Tang, and Y. Cao, "Coverage adaptive optimization algorithm of static-sensor networks for target discovery," *Chin. J. Electron.*, vol. 28, no. 2, pp. 398–403, 2019.

[2] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019.

[3] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.

[4] Q. Wang, B. Rengarajan, and J. Widmer, "Increasing opportunistic gain in small cells through energy-aware user cooperation," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6356–6369, Nov. 2014.

[5] S. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1855–1867, Mar. 2020.

[6] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019.

[7] Q. Lin *et al.*, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[8] Y. Tan and K. Ding, "A survey on GPU-based implementation of swarm intelligence algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2028–2041, Sep. 2016.

[9] F. Yang, P. Wang, Y. Zhang, L. Zheng, and J. Lu, "Survey of swarm intelligence optimization algorithms," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, 2017, pp. 544–549.

[10] Z. Zhang, K. Long, J. Wang, and F. Dressler, "On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 513–537, 1st Quart., 2014.

[11] S. R. Jondhale and R. S. Deshpande, "Kalman filtering framework-based real time target tracking in wireless sensor networks using generalized regression neural networks," *IEEE Sensors J.*, vol. 19, no. 1, pp. 224–233, Jan. 2019.

[12] S. He, H. Shin, and A. Tsourdos, "Trajectory optimization for target localization with bearing-only measurement," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 653–668, Jun. 2019.

[13] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered uav communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, Jun. 2019.

[14] N. Zhao *et al.*, "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[15] H. Pei, "Finite-time consensus tracking of nonlinear heterogeneous multi-agent systems with disturbances and directed graphs," *IEEE Access*, vol. 8, pp. 96316–96321, 2020.

[16] J. Moon, S. Papaioannou, C. Laoudias, P. Kolios, and S. Kim, "Deep reinforcement learning multi-UAV trajectory control for target tracking," *IEEE Internet Things J.*, early access, Apr. 19, 2021, doi: 10.1109/JIOT.2021.3073973.

[17] Q. Shi, X. Cui, S. Zhao, and M. Lu, "Sequential TOA-based moving target localization in multi-agent networks," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1719–1723, Aug. 2020.

[18] D. Yan, Y. Wang, J. Wang, H. Wang, and Z. Li, "K-nearest neighbor search by random projection forests," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 147–157, Mar. 2021.

[19] S. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao, "Development of UAV-based target tracking and recognition systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3409–3422, Aug. 2020.

[20] J. Guo, D. Li, and B. He, "Intelligent collaborative navigation and control for AUV tracking," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1732–1741, Mar. 2021.

[21] M. N. Anjum and H. Wang, "Mobility modeling and stochastic property analysis of airborne network," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1282–1294, Jul.-Sep. 2020.

[22] T. Ahmed B., M. S. Krishnan, and A. K. Anil, "A predictive analysis on the influence of WiFi 6 in fog computing with OFDMA and MU-MIMO," in *Proc. 4th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, 2020, pp. 716–719.

[23] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[24] J. M. Kim, Y. G. Kim, and S. W. Chung, "Stabilizing CPU frequency and voltage for temperature-aware DVFS in mobile devices," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 286–292, Jan. 2015.

[25] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[26] J. Lee, H.-J. Noh, and J. Lim, "Dynamic cooperative retransmission scheme for TDMA systems," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2000–2003, Dec. 2012.

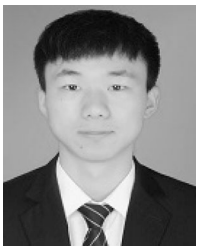
[27] V. Balakrishnan, "Lyapunov functionals in complex /spl mu/ analysis," *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1466–1479, Sep. 2002.

[28] Z. Qin, H. Yao, and T. Mai, "Traffic optimization in satellites communications: A multi-agent reinforcement learning approach," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2020, pp. 269–273.

[29] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[30] U. Stumper, "Uncertainties of VNA S-parameter measurements applying the TAN self-calibration method," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 2, pp. 597–600, Apr. 2007.

[31] S. Park, Y. Min, J.-S. Ha, D. Cho, and H.-L. Choi, "A distributed ADMM approach to non-myopic path planning for multi-target tracking," *IEEE Access*, vol. 7, pp. 163589–163603, 2019.



Longyu Zhou is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China.

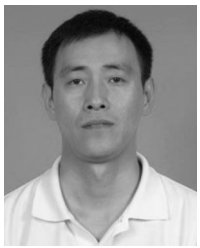
His research interests include Internet of Things, edge intelligence, resource scheduling, and wireless sensor networks.

Mr. Zhou was a recipient of the Best Paper Award in 20th IEEE Conference on Communications and Technology. He serves as a reviewer for the IEEE Global Communications Conference.

Supeng Leng (Member, IEEE) received the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2005.

He is currently a Full Professor and the Vice Dean with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. He is also the Leader with Research Group, Ubiquitous Wireless Networks. He has been working as a Research Fellow with Network Technology Research Center, NTU. He has published over 150 research articles in recent years. His research interests include spectrum, energy, routing and networking in the Internet of Things, vehicular networks, broadband wireless access networks, smart grid, and the next-generation mobile networks.

Prof. Leng serves as an organizing committee chair and a TPC member for many international conferences, and a reviewer for over ten international research journals.



Qiang Liu (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1996, 2000, and 2012, respectively.

After graduating from M.S. study in 2000, he has worked with the School of Communication and Information Engineering, UESTC, where he is currently an Associate Professor. He had worked with the University of Essex, Colchester, U.K. from December 2012 to December 2013 and the University of California at Davis, Davis, CA, USA, from August 2017 to August 2018, as a Visitor Scholar. His researches focus on wireless sensor networks, Internet of Things, broadband wireless networks, and molecular communication.

Dr. Liu is a reviewer of IEEE TRANSACTIONS ON NANOBIOSCIENCE and *International Journal of Communication Systems*. He is the member of IEEE ComSoc.

Qing Wang (Senior Member, IEEE) received the Ph.D. degree from UC3M, Getafe, Spain, and IMDEA Networks Institute, Madrid, Spain, in 2016.

He is an Assistant Professor with Embedded and Networked Systems Group, Delft University of Technology, Delft, The Netherlands. His research interests include visible light communication and sensing systems, and the Internet of Things.

Dr. Wang has received several awards, including the MobiCom Honourable Mention Award in 2020, the COMSNETS Best Paper Award in 2019, the Accenture Innovation Award in 2017, and the Best Paper Runner Up at CoNEXT in 2016. He is the co-founder of OpenVLC, an open-source and low-cost platform for VLC research. His research outcomes on active/passive visible light communication and sensing systems have been published at IEEE/ACM conferences and journals, such as ACM MobiCom, CoNEXT, SenSys, IEEE INFOCOM, IEEE/ACM TRANSACTIONS ON NETWORKING, and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.