A Crowdsensing-based Cyber-physical System for Drone Surveillance Using Random Finite Set Theory

CHAOQUN YANG, Zhejiang University, China
LI FENG, Macau University of Science and Technology, China
ZHIGUO SHI, Zhejiang University and Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China
RONGXING LU, University of New Brunswick, Canada
KIM-KWANG RAYMOND CHOO, University of Texas at San Antonio, USA

Given the popularity of drones for leisure, commercial, and government (e.g., military) usage, there is increasing focus on drone regulation. For example, how can the city council or some government agency detect and track drones more efficiently and effectively, say, in a city, to ensure that the drones are not engaged in unauthorized activities? Therefore, in this article, we propose a crowdsensing-based cyber-physical system for drone surveillance. The proposed system, CSDrone, utilizes surveillance data captured and sent from citizens' mobile devices (e.g., Android and iOS devices, as well as other image or video capturing devices) to facilitate jointly drone detection and tracking. Our system uses random finite set (RFS) theory and RFS-based Bayesian filter. We also evaluate CSDrone's effectiveness in drone detection and tracking. The findings demonstrate that in comparison to existing drone surveillance systems, CSDrone has a lower cost, and is more flexible and scalable.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: Drone surveillance system, Bayesian filter, crowdsensing, random finite set

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61772467, No. 61872452, and No. 61872451, in part by National Key Research and Development Program of China under Grant No. 2018YFB1702101, in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LR16F010002, and in part by the Macao FDCT under Grant No. 0098/2018/A3.

Authors' addresses: C. Yang, College of Information Science and Electronic Engineering, Zhejiang University, 38 Zheda Road, Hangzhou, Zhejiang, 310027, China; email: chaoqunyang@zju.edu.cn; L. Feng, Faculty of Information Technology, Macau University of Science and Technology, Macau, China; email: lfeng@must.edu.mo; Z. Shi, College of Information Science and Electronic Engineering, Zhejiang University, Alibaba-Zhejiang University Joint Institute of Frontier Technologies, 1818-2 Westwenyi Road, Hangzhou, Zhejiang, China; email: shizg@zju.edu.cn; R. Lu, Faculty of Computer Science, University of New Brunswick, 550 Windsor Street, Fredericton, Canada; email: rlu1@unb.ca; K.-K. R. Choo, Department of Information Systems and Cyber Security, University of Texas at San Antonio, 1 UTSA Cir, San Antonio, TX; email: raymond.choo@fulbrightmail.org.

2378-962X/2019/08-ART42 $15.00
https://doi.org/10.1145/3342049

ACM Reference format:
Chaoqun Yang, Li Feng, Zhiguo Shi, Rongxing Lu, and Kim-Kwang Raymond Choo. 2019. A Crowdsensing-based Cyber-physical System for Drone Surveillance Using Random Finite Set Theory. *ACM Trans. Cyber-Phys. Syst.* 3, 4, Article 42 (August 2019), 22 pages.
https://doi.org/10.1145/3342049

1 INTRODUCTION

In recent years, the use of drones (also referred to as unmanned aerial vehicles (UAVs)) has increased exponentially, as the applications of drones increase (e.g., commercial deliveries [4, 25], agriculture [18, 33], search and rescue [24, 32], and traffic monitoring [34–36]). Similar to other consumer technologies, such as driverless vehicles [13, 26], drones are a double-edged sword, in the sense that it can be also abused for nefarious purposes, such as infringing of privacy [8, 22], being used as an improvised explosive device [1, 15, 19], and smuggling of contraband, such as illicit drugs [7, 11, 15]. There have also been real-world incidents involving drones in a number of countries [1, 7, 19].

Hence, it is not surprising that both public and private sector organizations as well as the research community are paying closer attention to drones from a number of perspectives [5, 10] (e.g., security, privacy, performance, regulation, and surveillance). For example, the French-German Research Institute of Saint-Louis presented a system that combines acoustic arrays and optical senors to enhance drone tracking performance [6]. By integrating multiple passive sensors, Shi et al. designed a drone surveillance system (ADS-ZJU) for drone detection and localization [23]. Fu et al. developed a software defined radio (SDR)-based prototype for drone detection [9]. DedDrone, a German company, developed a platform (DroneTracker) to detect amateur drones via acoustic, RF, and optimal sensors [3]. Combining both radar target detection and electro-optical classification, Plextek Limited (an organization) proposed a counter-drone system (AUDS) to remotely detect, track, and classify drones [2].

Existing systems, however, have a number of limitations, such as the following:

- **High cost of equipment in terms of purchase, deployment, and maintenance.** For example, a large number of sensors generally need to be purchased and deployed in a specified surveillance region in advance. There are also ongoing maintenance costs.
- **Limited scalability.** Once a drone surveillance system is deployed, the scalability of the surveillance region is limited unless more sensors are purchased and deployed.
- **Limited generalization capability.** Most existing drone surveillance systems are designed for specific drones or scenarios [8]. Thus, this limits their applications to other types of drones or scenarios.
- **High radiant power.** Drone surveillance systems that use radars may not be permitted to be deployed in urban areas due to high radiant power.

Therefore, to mitigate the above limitations, in this article, we propose a drone surveillance system designed to be low cost, flexible, and scalable. Unlike existing drone surveillance systems that are mostly physical system-based, the proposed system (hereafter referred to as CSDrone) combines physical, information, and communication entities. In other words, CSDrone is a typical cyber-physical system. Specifically, in CSDrone, we utilize crowdsensing to efficiently collect data via mobile devices such as Android and iOS devices [16] see Figure 1. We also use the random finite set (RFS) theory [17], a powerful technique for data analytics and data fusion, in CSDrone to analyze the massive data in the system (acquired by and sent from citizen's mobile devices and other image/video capturing devices) to detect and track drones.

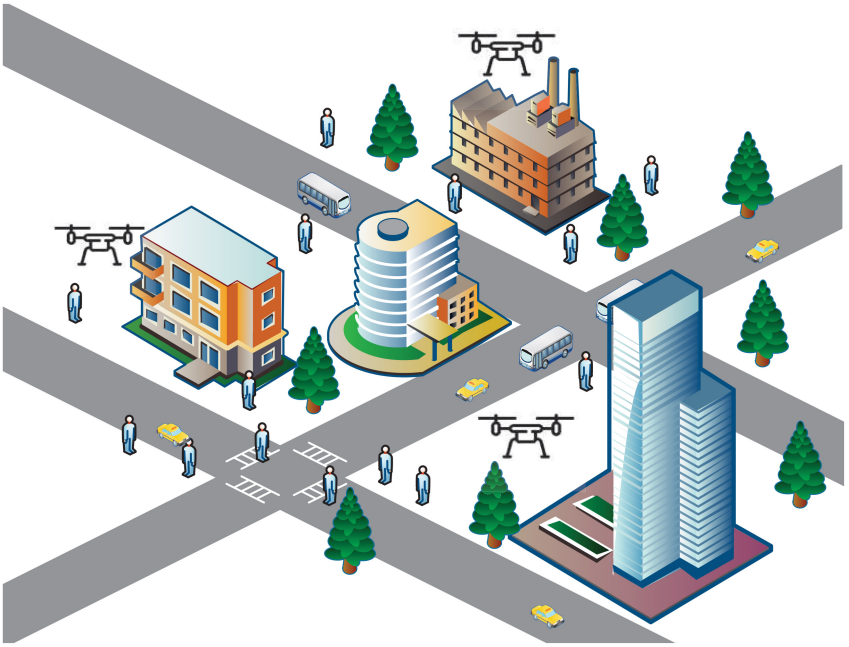

Fig. 1. Detecting drones via smartphones in an urban area: A potential scenario.

A crowdsourcing approach allows us to tap into previously non-utilized sources, such as users' mobile devices, without incurring expensive equipment setup, deployment, and maintenance. Such devices are capable of capturing images and videos that can be used to facilitate drone detection and tracking, for example, due to their in-built GPS receivers, acoustic sensors, and WiFi modules. For example, by using the fast Fourier transform algorithm [29] to analyze the received acoustic signals from the acoustic sensors and using the generalized cross-correlation algorithm [12] to calculate signals' time difference of arrival, mobile devices can obtain the detection and the direction of arrival (DOA) results of drones.

The question then is, "How can we motivate or encourage citizen participation, for example, to contribute their acquired images or videos, particularly in a privacy-sensitive society?" Incentive mechanism to encourage mobile users is not necessarily financial, as citizens have a vested interest for the city they live or work in to remain safe and free from terrorist activities. The data contributed by these users can then be fused with other information to more efficiently and effectively detect and track drones. One such example scenario is presented in Figure 2. Specifically, this scenario is in an urban area setting, where a group of mobile device users are in the area of interest who can then opportunistically contribute to the drone detection and tracking efforts.

In summary, the key contribution of this article is the proposed three-layer cyber-physical system utilizing crowdsensing to facilitate drone surveillance. At the time of this research, this is the first crowdsensing-based drone surveillance system presented in the literature. Specifically, in the system, we utilize RFS theory, where RFS-based Bayesian filter and its approximation is used to help us achieve drone detection and tracking by using the RFS-based Bayesian filter and its approximation.

Section 2 briefly describe the relevant background materials. In Sections 3 and 4, we present the proposed architecture and the underpinning RFS-based formulation. In Section 5, we present the iterative RFS-based Bayesian filter and its particle approximation, which are used for drone

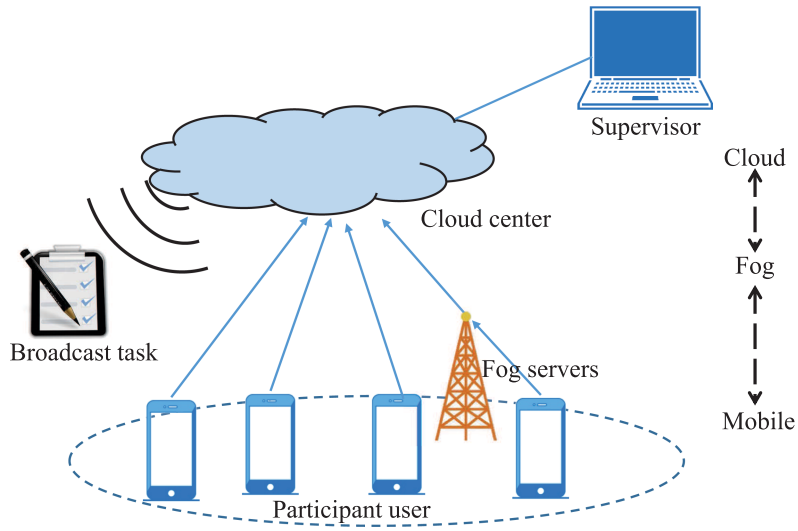

Fig. 2. CSDrone architecture.

detection and tracking. Section 6 presents the evaluation findings, and the last section concludes the article.

2 PRELIMINARIES

Prior to introducing the basic concepts in RFS theory, we will now describe the notations and definitions used in the remainder of this article.

Lower case letters denote scalars, and bold lower case letters denote vectors. We use bold capital letters to represent matrices, and bold Greek letters to represent RFSs. Also, $\boldsymbol{X}^T$ denotes the transpose of matrix $\boldsymbol{X}$, $\mathcal{N}(a, b)$ is the Gaussian distribution with mean a and covariance b, and $\mathcal{U}(a, b)$ is the uniform distributed between a and b.

A random finite set can be taken as a random variable, whose values are taken as unordered finite sets [20]. The obvious differences between a random finite set and a random vector [27, 30] are as follows:

(1) The number of elements in a random finite set is random; and
(2) The elements themselves are random and unordered.

Let us take the detection set of a radar's detector as an example, where z_i and Z denote detection and the detection set, respectively. Then, Z may be taken from $\{\emptyset\}, \{z_1\}, \{z_1, z_2\}, \ldots, \{z_1, \ldots, z_m\}, \ldots$. If Z is a RFS, then both m and z_i are random.

For RFS $\Omega = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, its set p.d.f. is defined as [17]

$$\begin{aligned} f(\Omega) &= f(\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}) \\ &= n!\rho(n)p_n(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n), \end{aligned} \tag{1}$$

where $|\Omega| = n$ is the cardinality (the number of elements) of the RFS, $\rho(n)$ is the distribution of the number of elements, and $p_n(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ is the family of symmetric joint distributions of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ [20]. Taking a Poisson RFS Ω as an example, the distribution of the number of elements is also Poisson; that is,

$$\rho(n) = \frac{e^{-\lambda}\lambda^n}{n!}, n = 0, 1, 2, \ldots. \tag{2}$$

Meanwhile, the elements of Ω are independent identically distributed according to p.d.f. $p(\boldsymbol{x})$. Then, we have

$$p_n(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n) = \prod_{\boldsymbol{x} \in \Omega} p(\boldsymbol{x}). \tag{3}$$

According to Equation (1), it follows that

$$f(\Omega) = n! \frac{e^{-\lambda} \lambda^n}{n!} \prod_{\boldsymbol{x} \in \Omega} p(\boldsymbol{x}) \tag{4}$$

$$= e^{-\lambda} \prod_{\boldsymbol{x} \in \Omega} \lambda p(\boldsymbol{x}). \tag{5}$$

For a continuous random variable, the integral of its p.d.f. is its cumulative distribution function (c.d.f.). Similarity, for a RFS, the integral of the set p.d.f. $f(\Omega)$ also exists, which is defined as

$$\int f(\Omega) \delta\Omega = f(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int f(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}) d\boldsymbol{x}_1, \ldots, d\boldsymbol{x}_n. \tag{6}$$

The belief function of a RFS Ω, $\beta_\Omega(\Psi)$, is similar to the c.d.f. of a random variable. $\beta_\Omega(\Psi)$ is defined as

$$\begin{aligned} \beta_\Omega(\Psi) &= Pr(\Omega \in \Psi) \\ &= \int_\Psi f(\Omega) \delta\Omega. \end{aligned} \tag{7}$$

For a RFS, its belief function is as important as its set p.d.f., because both totally capture the statistical characteristic of the RFS. In Section 5, we will present how to obtain the set p.d.f. of an RFS according to Equations (6) and (7).

3 PROPOSED CSDRONE SYSTEM ARCHITECTURE

In this section, we will introduce the architecture of our proposed CSDrone.

As illustrated in Figure 2, CSDrone consists of a three-layer mobile-fog-cloud hierarchy. The mobile layer consists of a (large) number of geo-distributed mobile users, but are physically within the region of interest. In other words, these users will contribute and regularly receive the broadcast tasks of drone detection from the cloud center. To encourage user participation and ensure the authenticity/quality of user reports, we need to sufficiently reward these users. While typical rewards are usually financially related (e.g., cash and credit), rewards can also be in other forms such as peer recognition (similar to the Publons[1] website used to encourage scholars to participate in peer review of manuscripts and showcase their peer review record).

After performing a cost-benefit analysis (i.e., analyzing the tradeoff between the reward and his/her current agenda), the user will decide to accept or decline the tasks. Usually, the user's agenda involves numerous subjective or objective factors, such as his/her emotion, location, working state and so on.

Once the user accepts the task, he/she will run the CSDrone mobile application (app). There are four main functions of the CSDrone app:

(1) Continuously scan the sampled signals from the built-in sensors, such as acoustic sensors and cameras on the user's device;
(2) Analyze the sampled signals, obtain detection results and calculate DOA results (if detected);

[1] https://publons.com/home/.

(3) Reports detection and DOA results to the cloud center; and
(4) Invoke the user's GPS data in the user's device.

The fog layer consists of several fog servers, which are deployed in the region of interest (i.e., surveillance region). The aim of the fog layer is to build a bridge between the cloud center and the user devices that do not have sufficient computational capacity to compute the detection and DOA results.

The cloud center is the core of CSDrone, which has the following functions:

(1) Broadcast tasks of drone detection;
(2) Design appropriate incentive mechanisms and privacy-preserving mechanisms;
(3) Integrate and fuse reported results from mobile users and fog servers;
(4) Obtain global detection and operate drone tracking (if required); and
(5) Output detection and tracking results to the relevant stakeholder.

Compared with existing surveillance systems (e.g., ADS-ZJU [23], DroneTracker [3], AUDS [2]), CSDrone has the following advantages:

- **Lower cost.** Due to the bring your own device (BYOD) setting, there is no cost required to acquire, install and maintain the sensors to collect data. However, there will be a need to acquire, install and maintain fog servers should they be deployed. Such servers are still a cheaper option.
- **More flexible, extensive, and ubiquitous surveillance coverage.** By encouraging more users to participate in the tasks, the surveillance coverage will be more flexible and more extensive. Meanwhile, the proliferation of mobile devices also makes the surveillance coverage ubiquitous and scalable.
- **More surveillance data.** Massive data can be user-contributed, particularly at crowded places and during large scale events such as a concert, a sport event, and so on [8]. There may also be duplicated data, and hence this also raises the need for data deduplication efforts.
- **More resilience and robust to sensor fault.** The redundancy of many more sources of data significantly increases the resilience and robustness to false data injection and sensor failure.
- **No radiant power.** Safer for human health.

However, due to the utilization of crowdsensing technique, CSDrone also faces the following open challenges:

- Formulation for user reporting scheme. Due to the randomness of human mobility, the varying number and the time-varying location of user device, it may be challenging to accurately formulate a reporting scheme.
- Data fusion for a large source of data can be challenging, particularly if we also require the analysis results to be available in real-time.
- A trade-off between incentive and privacy-preserving mechanisms. When accepting the tasks, participants consume their resources (e.g., time and device battery life) and their privacy (e.g., GPS data) may also be compromised. Therefore, it is necessary to design appropriate incentive mechanisms to compensate for participation, without compromising their privacy.
- Real-time implementation of detection and DOA estimation algorithms on mobile platforms can be challenging due to the large amount of data and constant changing nature of events.

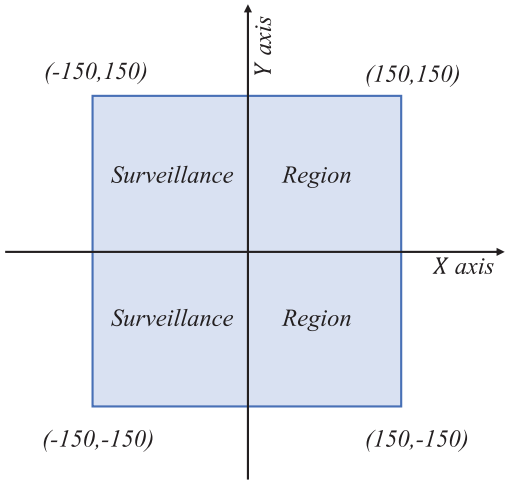

Fig. 3. An example of the surveillance region.

In the following, we focus on the first two challenges. Specifically, we utilize the RFS theory to accurately formulate CSDrone including drone dynamics, participant dynamics and device reporting scheme. Further, we focus on dealing with the challenge in drone detection and tracking by fusing all devices' reports.

4 RFS-BASED SYSTEM FORMULATION

In this section, we use the RFS theory to formulate three main elements in CSDrone, namely, drone dynamics, participant dynamics, and device reporting scheme (see Sections 4.1–4.3).

4.1 Drone Dynamics

For simplicity, let us consider a scenario where no more than one drone exists in a two-dimensional surveillance region. The state vector of the drone at time step k is denoted as $\boldsymbol{x}_k = [x_k,\ \dot{x}_k,\ y_k,\ \dot{y}_k]^T$, where (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$ represent the drone's position and velocity at time step k, respectively. The following assumptions about the drone dynamics are made:

A1: The Markov transition density of the drone state is denoted as $f_{k+1|k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$. For instance, if

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{8}$$

holds true, where $\boldsymbol{v}_k$ is spatially and temporally white Gaussian noise that follows $\mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$, then $f_{k+1|k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{F}\boldsymbol{x}_k, \boldsymbol{R})$.

A2: If the drone exists in the surveillance region at time step k, then it has a probability p_s of remaining in the surveillance region at time step $k+1$.

A3: If the drone is out of the surveillance region at time step k, then it has a probability p_r of appearing in the surveillance region. The initial state vector $\boldsymbol{b}$ when it appears in the surveillance region is assumed to follow a p.d.f. $r(\boldsymbol{b})$. In other words, $r(b)$ is the p.d.f. of the drone's state when the drone first appears in the surveillance region. For example, consider the surveillance region as shown in Figure 3. Let us assume that $r(\boldsymbol{b}) = \mathcal{U}(\boldsymbol{a}, \boldsymbol{c})$ where $\boldsymbol{a} = [-150, 2, -150, 2]^T, \boldsymbol{c} = [150, 3, 150, 3]^T$. Then, both initial x and y positions of the drone follow the uniform distribution between -150 and 150. Thus, both initial x velocity and y velocity of the drone follow the uniform distribution between 2 and 3.

Now, we formulate the drone dynamics at time step $k-1$ by an RFS Σ_{k-1}. It is straightforward that Σ_{k-1} only exists in two possible cases, i.e., $\Sigma_{k-1} = \{\boldsymbol{x}_{k-1}\}$ or $\Sigma_{k-1} = \{\emptyset\}$. Accordingly, given Σ_{k-1} at time step k, the RFS-based drone dynamics at time step k can be formulated as

$$\Sigma_k = \begin{cases} \{\boldsymbol{x}_k\} \cap \emptyset^{p_s}, & \text{if } \Sigma_{k-1} = \{\boldsymbol{x}_{k-1}\}, \\ \{\boldsymbol{b}\} \cap \emptyset^{p_r}, & \text{if } \Sigma_{k-1} = \emptyset, \end{cases} \tag{9}$$

where $\emptyset^p$ is a discrete random set whose probability follows [17]

$$Pr(\emptyset^p = \Omega) = \begin{cases} 1-p, & \text{if } \Omega \text{ is } \emptyset, \\ p, & \text{if } \Omega \text{ is a singleton set}, \\ 0, & \text{if otherwise.} \end{cases} \tag{10}$$

Let $m(k)$ denotes the cardinality of Σ_k, and it is straightforward to know that $m(k)$ can only take values from $\{0, 1\}$.

4.2 Participant Dynamics

To achieve drone tracking, the cloud center needs to invoke each participant's GPS data. Let $(\xi_{i,k}, \zeta_{i,k})$ and $(\dot{\xi}_{i,k}, \dot{\zeta}_{i,k})$ represent the GPS location and velocity of the ith participant user, respectively, and let $\boldsymbol{y}_{i,k} = [\xi_{i,k},\ \dot{\xi}_{i,k},\ \zeta_{i,k},\ \dot{\zeta}_{i,k}]^T$ denote the GPS-based dynamics of the ith participant user. We should formulate the dynamics of the device carried by participants. However, since the devices are always with the users, we assume that they share the same dynamics for simplicity.

Then, we use the following RFS to denote the collected GPS data from all participants at time step k,

$$\Theta_k = \{\boldsymbol{y}_{i,k}, \boldsymbol{y}_{2,k}, \ldots, \boldsymbol{y}_{n(k),k}\}, \tag{11}$$

where $n(k)$ is the number of participants at time step k.

4.3 Device Reporting Scheme

If the user accepts the tasks, then he/she will run the CSDrone app. Upon confirming drone detection, the CSDrone app will run the DOA estimation algorithm and report the estimated DOA to the cloud center during each time step. Here, we emphasize that the drone detection may originate from a drone or a false alarm. If nothing is detected, then the CSDrone app will also report this to the cloud center during this time step. Therefore, each report either contains one estimated DOA, or is the report of "no detection."

Let an RFS $\Phi_{i,k}$ represents the report of the ith participant user's smartphone at time step k, then $\Phi_{i,k} = \{\theta_{i,k}\}$ means that the reported DOA estimated by this device, while $\Phi_{i,k} = \{\emptyset\}$ means the report of "no detection." Note that the reports of all participants' devices are mutually independent at time step k, then these reports can be formulated by the following RFS:

$$\Phi_k = \Phi_{1,k} \cup \Phi_{2,k} \cup \cdots \cup \Phi_{n(k),k}. \tag{12}$$

Remark: From the cloud center's perspective, we seek to detect the drone and estimate its state $\boldsymbol{x}_k$ (if detected), from a sequence of reports $\Phi_{1:k}$ and GPS data $\Theta_{1:k}$. Here, $\Phi_{1:k} \equiv \Phi_1, \ldots, \Phi_k$ is the sequence of all reports until time step k, and $\Theta_{1:k} \equiv \Theta_1, \ldots, \Theta_k$ is the sequence of all participants' GPS data until time step k. Since Equation (9) faithfully encapsulates the information in the drone dynamics including the number of drones and the drone's states (if detected), our goal equates to estimating Σ_k from a sequence of reports $\Phi_{1:k}$ and GPS data $\Theta_{1:k}$.

5 DRONE DETECTION AND TRACKING

5.1 RFS-based Bayesian Filter

Let $\sigma_{k|k-1}(\Sigma_k|\Sigma_{k-1})$ and $\phi_k(\Phi_k|\Sigma_k,\Theta_k)$ denote the set p.d.f. of Equations (9) and (12), respectively. Then, the posterior set p.d.f of Σ_k, i.e., $\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k})$, can be iteratively estimated by the RFS-based Bayesian filter, which contains the following two steps:

Prediction step: Given the posterior set p.d.f. of Σ_{k-1}, i.e., $\pi_{k-1|k-1}(\Sigma_{k-1}|\Phi_{1:k-1},\Theta_{1:k-1})$ at time step $k-1$, the predicted set p.d.f. at time step k is given as

$$\pi_{k|k-1}(\Sigma_k|\Phi_{1:k-1},\Theta_{1:k-1}) = \int \sigma_{k|k-1}(\Sigma_k|\Sigma_{k-1})\pi_{k-1|k-1}(\Sigma_{k-1}|\Phi_{1:k-1},\Theta_{1:k-1})\delta\Sigma_{k-1}. \tag{13}$$

Update step: On receipt of the new reports set Φ_k and corresponding Θ_k, the posterior p.d.f. at time step k is given as

$$\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k}) = \frac{\phi_k(\Phi_k|\Sigma_k,\Theta_k)\pi_{k|k-1}(\Sigma_k|\Phi_{1:k-1},\Theta_{1:k-1})}{\int \phi_k(\Phi_k|\Sigma,\Theta_k)\pi_{k|k-1}(\Sigma|\Phi_{1:k-1},\Theta_{1:k-1})\delta\Sigma}. \tag{14}$$

According to Equations (13) and (14), once we obtain $\sigma_{k|k-1}(\Sigma_k|\Sigma_{k-1})$ and $\phi_k(\Phi_k|\Sigma_k,\Theta_k)$, we can iteratively estimate the posterior set p.d.f. $\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k})$ over time. Therefore, the main challenge lies in the derivation of $\sigma_{k|k-1}(\Sigma_k|\Sigma_{k-1})$ and $\phi_k(\Phi_k|\Sigma_k,\Theta_k)$.

5.2 Set p.d.f. of Drone Dynamics

Now, we will present the derivation of $\sigma_{k|k-1}(\Sigma_k|\Sigma_{k-1})$. Let us begin with $\Sigma_{k-1}=\emptyset$, and the belief function of Equation (9) can be written as

$$\begin{aligned}\beta_{k|k-1}(\Omega|\emptyset) &= Pr\{\Sigma_k \subseteq \Omega|\emptyset\} \\ &= (1-p_r+p_r\cdot Pr\{\{\boldsymbol{b}\}\subseteq\Omega|\emptyset\} \\ &= 1-p_r+p_r\int_\Omega r(\boldsymbol{b})d_{\boldsymbol{b}}.\end{aligned} \tag{15}$$

It follows that

$$\sigma_{k|k-1}(\Sigma_k|\emptyset) = \begin{cases} 1-p_r, & \text{if } \Sigma_k=\emptyset, \\ p_r\cdot r(\boldsymbol{b}), & \text{if } \Sigma_k\neq\emptyset, \\ 0, & \text{if otherwise.}\end{cases} \tag{16}$$

Similarly, if $\Sigma_{k-1}\neq\emptyset$, then the belief function and set p.d.f. of Equation (9) are as follows:

$$\begin{aligned}\beta_{k|k-1}(\Omega|\{\boldsymbol{x}_{k-1}\}) &= Pr\{\Sigma_k\subseteq\Omega|\{\boldsymbol{x}_{k-1}\}\} \\ &= (1-p_s)+p_s\cdot Pr\{\{\boldsymbol{x}_k\}\subseteq\Omega|\{\boldsymbol{x}_{k-1}\}\} \\ &= 1-p_s+p_s\int_\Omega f_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})d_{\boldsymbol{x}_k},\end{aligned} \tag{17}$$

$$\sigma_{k|k-1}(\Sigma_k|\{\boldsymbol{x}_{k-1}\}) = \begin{cases} 1-p_s, & \text{if } \Sigma_k=\emptyset, \\ p_s\cdot f_{k|k-1}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), & \text{if } \Sigma_k\neq\emptyset, \\ 0, & \text{if otherwise.}\end{cases} \tag{18}$$

5.3 Set p.d.f. of Smartphones' Report Scheme

Now, we will present the derivation of $\phi_k(\Phi_k|\Sigma_k,\Theta_k)$. Let us begin with the investigation of $\Phi_{i,k}$. Note that $\Phi_{i,k}=\{\emptyset\}$ when both the following two conditions are simultaneously satisfied:

(1) No drone is detected (if the drone exists); and
(2) No false alarm happens.

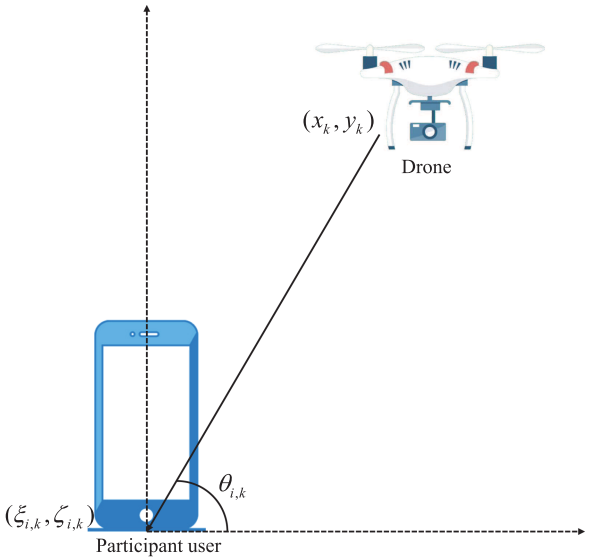

Fig. 4. Bearing measurement model of the ith participant user.

Thus, we have

$$Pr(\mathbf{\Phi}_{i,k} = \emptyset) = 1 - q_i = (1 - p_{i,f})(1 - p_{i,d}), \tag{19}$$

where $p_{i,f}$ and $p_{i,d}$ are the false alarm probability and the detection probability of the ith participant user, respectively. For simplicity, assume that $p_{i,f} = p_f$ and $p_{i,d} = p_d$, then $q_i = q$.

First, consider a special case in which $\Sigma_k = \emptyset$, then $q = p_f$ and

$$\mathbf{\Phi}_{i,k} = \{\vartheta\} \cap \emptyset^{p_f}, \tag{20}$$

where $\vartheta \in [-\pi/2, \pi/2]$ is a false alarm angle that follows a p.d.f. $\kappa(\vartheta)$. Similar to Equations (15) and (16), the belief function and set p.d.f. of Equation (20) are as follows:

$$\begin{aligned} \beta_{i,k}(\Omega|\emptyset, \mathbf{\Theta}_k) &= Pr\{\mathbf{\Phi}_{i,k} \subseteq \Omega|\emptyset, \mathbf{\Theta}_k\} \\ &= 1 - p_f + p_f \int_{\Omega} \kappa(\vartheta) d_{\vartheta}, \end{aligned} \tag{21}$$

$$\phi_{i,k}(\mathbf{\Phi}_{i,k}|\emptyset, \mathbf{\Theta}_k) = \begin{cases} 1 - p_f, & \text{if } \mathbf{\Phi}_{i,k} = \emptyset, \\ p_f \kappa(\vartheta), & \text{if } \mathbf{\Phi}_{i,k} \neq \emptyset, \\ 0, & \text{if otherwise.} \end{cases} \tag{22}$$

Second, consider a more general case in which $\Sigma_k \neq \emptyset$, then

$$Pr(\mathbf{\Phi}_{i,k} = \Omega) = \begin{cases} 1 - q, & \text{if } \Omega = \emptyset, \\ p_f, & \text{if } \Omega = \{\vartheta\}, \\ p_d(1 - p_f), & \text{if } \Omega = \{\theta_{i,k}\}. \end{cases} \tag{23}$$

As shown in Figure 4, $\theta_{i,k}$ can be obtained from the following bearing measurement model of the ith participant user

$$\theta_{i,k} = \arctan\left(\frac{y_k - \zeta_{i,k}}{x_k - \xi_{i,k}}\right) + w_{i,k}, \tag{24}$$

where $w_{i,k}$ is spatially and temporally white Gaussian noise that follows $\mathcal{N}(0, Q_i)$. For simplicity, we denote Equation (24) by $l_{i,k}(\theta_{i,k}|\boldsymbol{x}_k, \boldsymbol{y}_{i,k})$.

The belief function of $\Phi_{i,k}$ under the case $\Sigma_k \neq \emptyset$ is

$$\begin{aligned}\beta_{i,k}(\Omega|\{\boldsymbol{x}_k\},\Theta_k) &= Pr(\Phi_{i,k} \subseteq \Omega|\{\boldsymbol{x}_k\},\Theta_k) \\ &= 1-q+p_f\int_{\Omega}\kappa(\vartheta)d_{\vartheta} + (1-p_f)p_d\int_{\Omega} l_{i,k}(\theta_{i,k}|\boldsymbol{x}_k,\boldsymbol{y}_k)d_{\theta_{i,k}}. \end{aligned} \tag{25}$$

Therefore, the set p.d.f. of $\Phi_{i,k}$ under the case $\Sigma_k \neq \emptyset$ can be expressed as

$$\phi_{i,k}(\Phi_{i,k}|\{\boldsymbol{x}_k\},\Theta_k) = \begin{cases} 1-q, & \text{if } \Phi_{i,k}=\emptyset, \\ \lambda(\vartheta,\theta_{i,k}), & \text{if } \Phi_{i,k}\neq\emptyset, \\ 0, & \text{if otherwise,} \end{cases} \tag{26}$$

where

$$\lambda(\vartheta,\theta_{i,k}) = p_f\kappa(\vartheta) + (1-p_f)p_d l_{i,k}(\theta_{i,k}|\boldsymbol{x}_k,\boldsymbol{y}_k). \tag{27}$$

According to Equations (22) and (26), (27), if $\Sigma_k = \emptyset$ and $|\Phi_k| = j$, then the set p.d.f. of ϕ_k can be described as

$$\phi_k(\Phi_k|\Sigma_k,\Theta_k) = \frac{n_k!}{(n_k-j)!}(p_f\kappa(\vartheta))^j(1-p_f)^{(n_k-j)}, \tag{28}$$

where $j = 0, 1, \ldots, n_k$. However, if $|\Phi_k| = j$, and $\Sigma_k \neq \emptyset$, then

$$\phi_k(\Phi_k|\Sigma_k,\Theta_k) = q^j(1-q)^{(n_k-j)} \sum_{1\leq i_1\neq\cdots\neq i_j\leq n_k} \lambda(\vartheta,\theta_{i_1,k})\cdots\lambda(\vartheta,\theta_{i_j,k}). \tag{29}$$

We refer readers to the Appendix for the proofs of Equations (28) and (29).

5.4 Particle Approximation

By substituting Equations (16), (18), (22) and (28), (29) into the RFS-based Bayesian filter (see Equations (13) and (14)), the posterior p.d.f. $\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k})$ can be recursively updated in time. Unfortunately, due to the involved multiple integrals, the above RFS-based Bayesian filter usually suffers high computational complexity. As an alternative way, it can be approximated by a particle method.

Assume that the posterior set p.d.f. $\pi_{k-1|k-1}(\Sigma_{k-1}|\Phi_{1:k-1},\Theta_{1:k})$ can be approximated by a set of particles as

$$\pi_{k-1|k-1}(\Sigma_{k-1}|\Phi_{1:k-1},\Theta_{1:k}) \approx \sum_{i=1}^{L} w_{k-1}^{(i)}\delta_{\Sigma_{k-1}^{(i)}}(\Sigma_{k-1}), \tag{30}$$

where $\Sigma_{k-1}^{(i)}$ is the ith particle, $w_{k-1}^{(i)}$ is the weight corresponding to $\Sigma_{k-1}^{(i)}$ and $w_{k-1}^{(i)} \geq 0$, $\sum_1^L w_{k-1}^{(i)} = 1$, L is the number of particles, $\delta_{\Sigma_{k-1}^{(i)}}(\Sigma_{k-1})$ is the set-valued version of the Dirac delta function [20]. Then, the particles $\{\Sigma_k^{(i)}\}_1^L$ can be randomly generated as

$$\Sigma_k^{(i)} \sim \sigma_{k|k-1}\big(\cdot|\Sigma_{k-1}^{(i)}\big). \tag{31}$$

Using importance sampling [14] and Equation (14), the weight associated with the ith particle $\Sigma_k^{(i)}$ is updated as

$$w_k^{(i)} = \frac{\phi_k(\Phi_k|\Sigma_k^{(i)},\Theta_k)\sigma_{k|k-1}(\Sigma_k^{(i)}|\Sigma_{k-1}^{(i)})}{\eta_k(\Sigma_k^{(i)}|\Sigma_{k-1}^{(i)},\Phi_k)} w_{k-1}^{(i)}, \tag{32}$$

where $\eta_k(\cdot|\Sigma_{k-1}^{(i)},\Phi_k)$ is the importance sampling density [14]. For the ease of calculation, a naive choice of importance sampling density is $\eta_k(\cdot|\Sigma_{k-1}^{(i)},\Phi_k) = \sigma_{k|k-1}(\cdot|\Sigma_{k-1}^{(i)})$ [28]. Then, Equation (32) is simplified to

$$w_k^{(i)} = \phi_k\big(\Phi_k|\Sigma_k^{(i)},\Theta_k\big)w_{k-1}^{(i)}. \tag{33}$$

Then, we have

$$\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k}) \approx \sum_{i=1}^{L} w_k^{(i)} \delta_{\Sigma_k^{(i)}}(\Sigma_k), \tag{34}$$

and

$$\pi_{k|k}(m(k)|\Phi_{1:k},\Theta_{1:k}) \approx \sum_{i:|\Sigma_k^{(i)}|=m(k)} w_k^{(i)}, \tag{35}$$

where $\pi_{k|k}(m(k)|\Phi_{1:k},\Theta_k)$ is the cardinality distribution of $\pi_{k|k}(\Sigma_k|\Phi_{1:k},\Theta_{1:k})$.

Remark: Based on Equation (35), drone detection can be realized by calculating the expected a posteriori (EAP) or maximum a posteriori (MLE) estimator of $m(k)$. For example, the fact that the EAP estimator of $m(k)$ equals to 1 implies that CSDrone detects the drone. Then, the drone's state can be extracted by using cluster technique on these particles.

Algorithm 1 summarizes the particle approximation of the above RFS-based Bayesian filter. To reduce the problem of particle degeneracy, the step of resampling is added.

ALGORITHM 1: Particle approximation of the Bayesian RFS filter

Initialize: L, $\{\Sigma_0^{(i)}, w_0^{(i)}\}_{i=1}^{L}$;
for $k = 1, 2, \ldots$ **do**
  Procedure 1: Sampling;
  for $i = 1, \ldots, L$ **do**
    Generate $\Sigma_k^{(i)} \sim \sigma_{k|k-1}(\cdot|\Sigma_{k-1}^{(i)})$;
    Compute $w_k^{(i)} = \phi_k(\Phi_k|\Sigma_k^{(i)},\Theta_k)w_{k-1}^{(i)}$;
  end
  Procedure 2: Resampling;
  Compute $\sum_{j=1}^{L} w_k^{(i)}$;
  for $i = 1, \ldots, L$ **do**
    Normalize weight $w_k^{(i)} = w_k^{(i)} / \sum_{j=1}^{L} w_k^{(i)}$;
  end
  Obtain $\{\tilde{\Sigma}_L^{(i)}, \tilde{w}_k^{(i)}\}_{i=1}^{k}$ by applying a resampling algorithm on $\{\Sigma_k^{(i)}, w_k^{(i)}\}_{i=1}^{L}$;
end

6 EVALUATION

In this section, we present extensive numerical experiments to evaluate CSDrone's drone detection and tracking performance.

6.1 Linear Case with Static Users

Consider a two-dimensional case with no more than one drone observed by three mobile users over the region $[-150, 150] \times [-150, 150]$. The drone dynamics follow Equation (8), and

$$F = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{36}$$

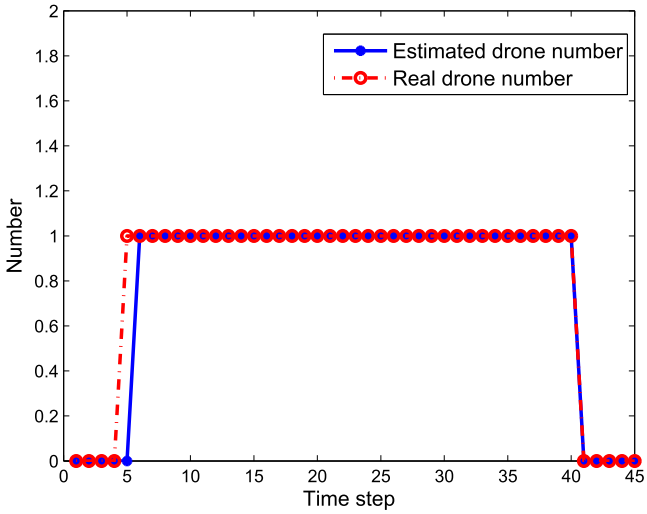

Fig. 5. Drone detection results (linear case with static users).

$$\boldsymbol{R} = 0.25\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{37}$$

Assume that the drone enters the region at time step 5 and escapes from this region at time step 40. The probability of survival and the probability of appearing are set to $p_s = 0.98$ and $p_r = 0.01$, respectively. The initial state vector $\boldsymbol{b}$ follows $r(\boldsymbol{b}) = \mathcal{U}(\boldsymbol{a}, \boldsymbol{c})$, where $\boldsymbol{a} = [-150, 2, -150, 2]^T$, $\boldsymbol{c} = [150, 2, 150, 2]^T$. Let us begin with a simple case where three users keep static and participate the tasks all the time. Their initial positions are $[50, 0]^T$, $[0, 50]^T$, and $[60, 100]^T$, respectively, and their detection rate and false alarm rate are $p_d = 0.98$ and $p_f = 0.05$, respectively. The false alarm ϑ follows $\kappa(\vartheta) = \mathcal{U}(-\pi/2, \pi/2)$. The measurement noises of each users are $Q_1 = Q_2 = Q_3 = \pi/180$.

To jointly evaluate the detection and estimation performance, we introduced optimal subpattern assignment (OSPA) distance. For $\boldsymbol{x}, \boldsymbol{y}$, let ${d_p}^{(c)}(\boldsymbol{x}, \boldsymbol{y}) = \min(c, ||\boldsymbol{x} - \boldsymbol{y}||)$, and Π_k denote the set of permutations on $\{1, 2, \ldots, k\}$ for any positive integer k. Then, for $p \geq 1, c > 0$, $\boldsymbol{\Phi} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ and $\boldsymbol{\Psi} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, the OSPA distance between $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ is defined as follows [21, 31]:

$$\bar{d_p}^{(c)}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \left(\frac{1}{n}\left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c)}(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})^p + c^p(n-m)\right)\right)^{\frac{1}{p}}, \tag{38}$$

and if $m > n$, $\bar{d_p}^{(c)}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \bar{d_p}^{(c)}(\boldsymbol{\Psi}, \boldsymbol{\Phi})$, and $\bar{d_p}^{(c)}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \bar{d_p}^{(c)}(\boldsymbol{\Psi}, \boldsymbol{\Phi}) = 0$ if $m = n = 0$. In the following simulations, $c = 300$ and $p = 1$ are set, the total number of particles is set to 2,048, and 100 Monte Carlo trials are operated.

In Figure 5, we present the drone detection results. Except for the detection result at time step 5, drone number has been accurately estimated at each time step. Figure 6 presents the estimation results of drone trajectories. Triangle symbols represent the coordinates of participant users. It can be seen that the trajectories of the drone are accurately estimated. As Figure 7 shows, except for the OSPA distance at time step 5, the OSPA distances during all time steps are very small. The reason for the over-high OSPA distance at time step 5 is the incorrect estimated drone number.

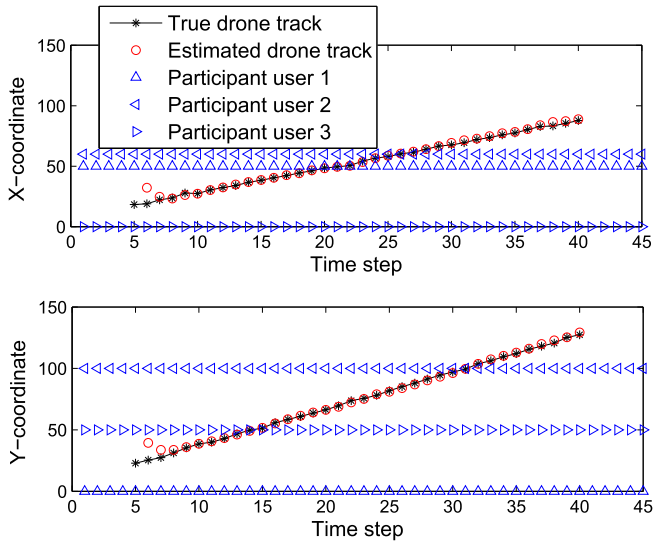

Fig. 6. Estimation results of drone trajectories (linear case with static users).

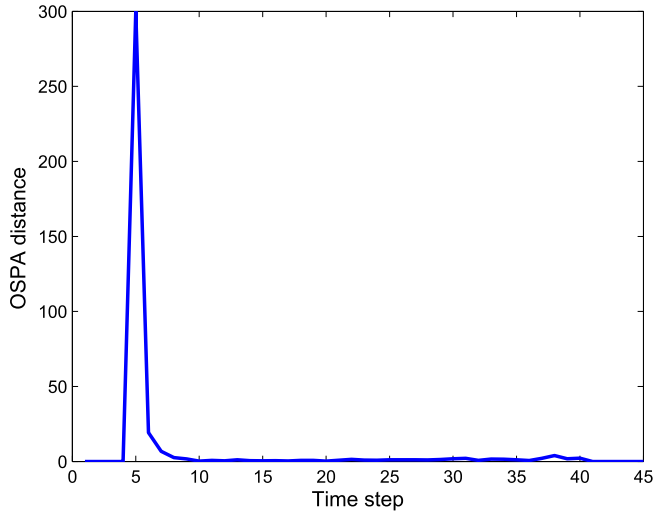

Fig. 7. OSPA distance (linear case with static users).

6.2 Linear Case with Mobile Users

Let consider a more complex case where each user has dynamics as Section 4.2 states. For $i = 1, 2, 3$, assume that each user's dynamics follow

$$\boldsymbol{y}_{i,k} = \boldsymbol{A}\boldsymbol{y}_{i,k-1} + \boldsymbol{v}_k, \tag{39}$$

where $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, G)$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{40}$$

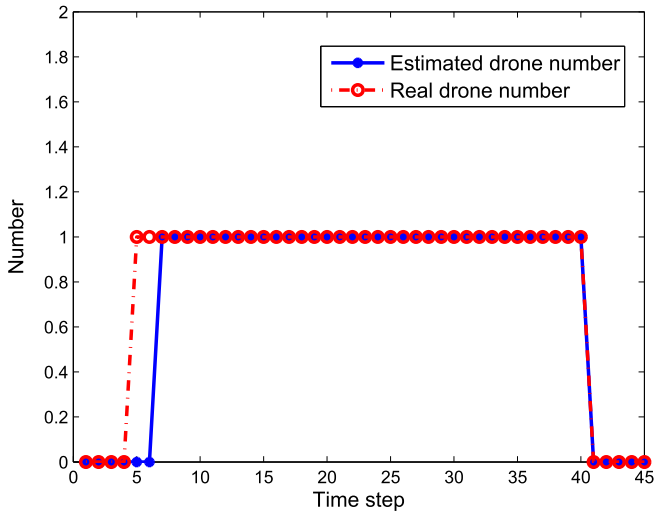

Fig. 8. Drone detection results (linear case with mobile users).

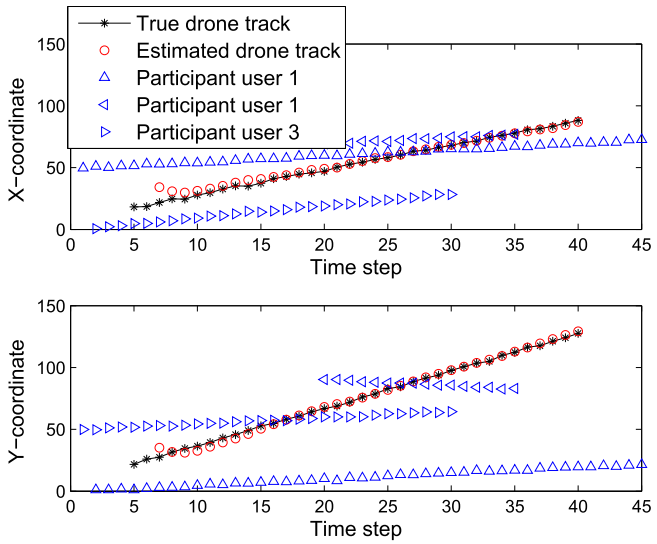

Fig. 9. Estimation results of drone trajectories (linear case with mobile users).

and

$$G = 0.25\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{41}$$

Their initial states are $[50, 0.5, 0, 0.5]^T$, $[0, 1, 50, 0.5]^T$, and $[60, 0.5, 100, -0.5]^T$, respectively. The first user participants the tasks all the time, while the second user and the third user participate the tasks during time steps 1–30 and 20–35, respectively.

The jointly detection and tracking results are presented in Figures 8–10.

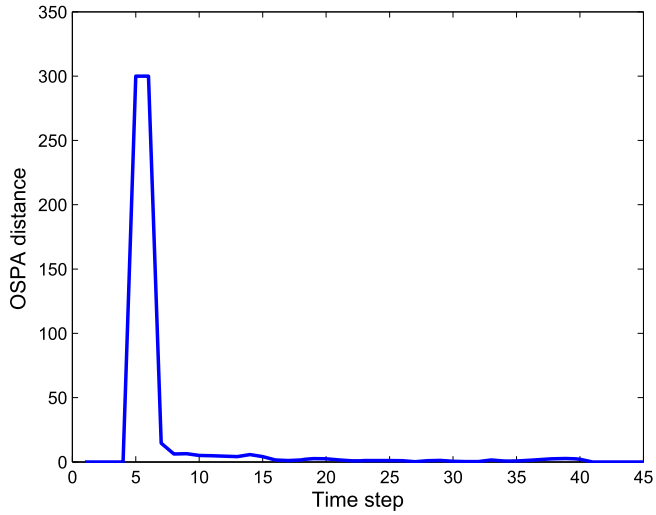

Fig. 10. OSPA distance (linear case with mobile users).

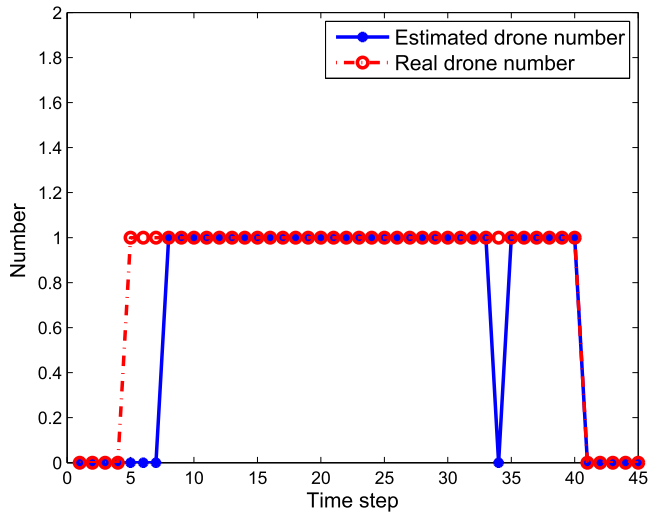

Fig. 11. The drone detection results (nonlinear case with mobile users).

Similar to the results of the linear case with static users, both the number and trajectories of the drone are simultaneously estimated with high accuracy. Note that there only exists one user who participates the tasks during time steps 36–45, while the results are still acceptable.

6.3 Nonlinear Case with Mobile Users

Consider the drone dynamics follow the nonlinear equations

$$\boldsymbol{x}_k = \boldsymbol{F}(\omega_{k-1})\boldsymbol{x}_{k-1} + \boldsymbol{v}_k, \tag{42}$$

$$\omega_k = \omega_{k-1} + \mu_k, \tag{43}$$

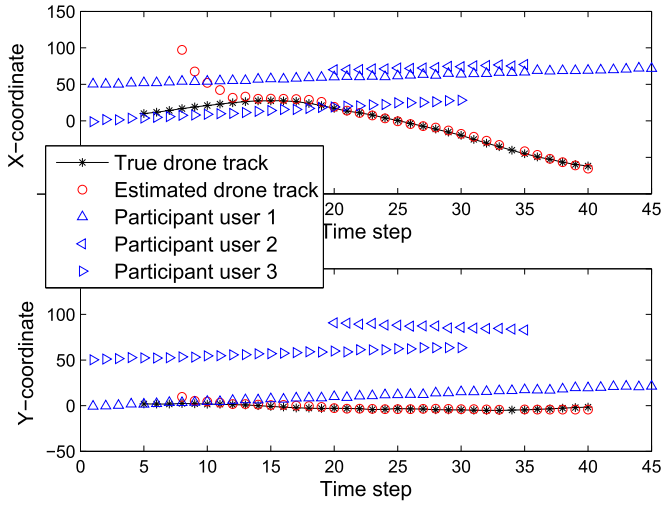

Fig. 12. The estimation results of drone trajectories (nonlinear case with mobile users).

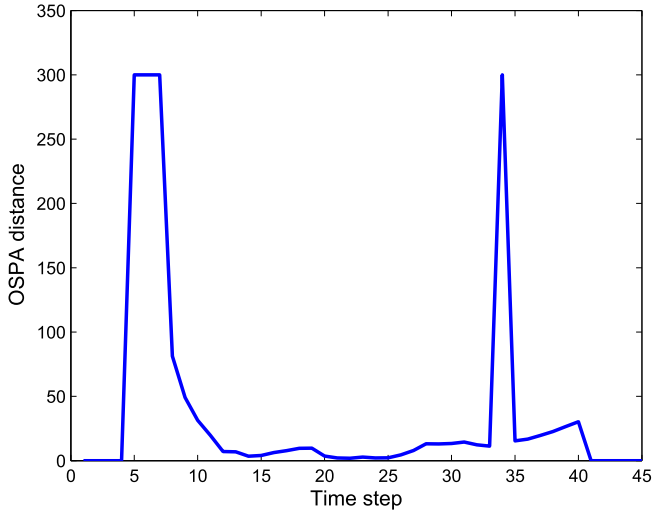

Fig. 13. The OSPA distance (nonlinear case with mobile users).

where $\boldsymbol{v}_k \sim \mathcal{N}(\mathbf{0}, G)$, $\mu_k \sim \mathcal{N}(0, (\pi/180)^2)$ and

$$\boldsymbol{F}(\omega) = \begin{bmatrix} 1 & \frac{\sin(\omega)}{\omega} & 0 & -\frac{1-\cos\omega}{\omega} \\ 0 & \cos\omega & 0 & -\sin\omega \\ 0 & \frac{1-\cos\omega}{\omega} & 1 & \frac{\sin(\omega)}{\omega} \\ 0 & \sin\omega & 0 & \cos\omega \end{bmatrix}. \tag{44}$$

The remaining settings are the same as those in the above cases.

The results are presented in Figures 11–13. In comparison with the OSPA distances in Figures 7 and 10, the OSPA distance in Figure 13 becomes much larger, which implies that the results of this case are worse than that of the above linear cases. However, from Figures 11 and 12, it can be seen

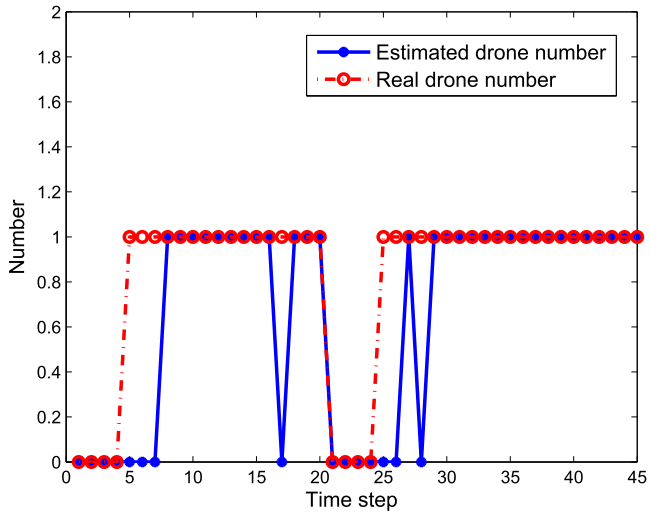

Fig. 14. Drone detection results (linear case with two drones and mobile users).

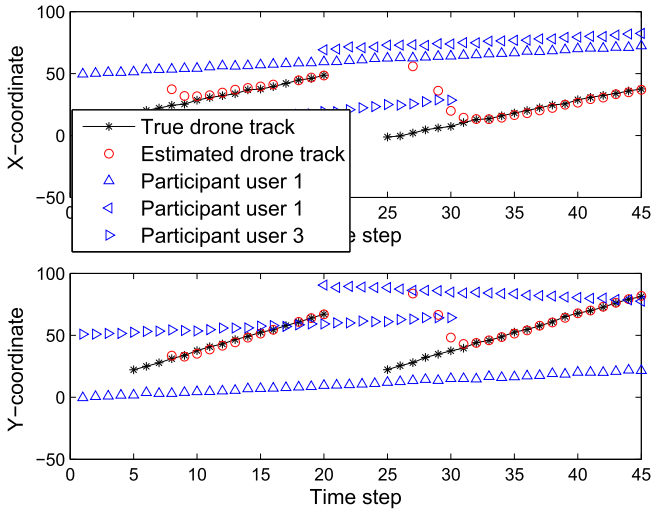

Fig. 15. Estimation results of drone trajectories (linear case with two drones and mobile users).

that both the number and the trajectories of the drone are still accurately estimated most of the time. Therefore, the proposed system is suitable for the nonlinear case.

6.4 Linear Case with Two Drones and Mobile Users

Let us assume that the first drone enters the region at time step 5 and escapes from the region at time step 20, and the second drone enters the region at time step 25 and escapes from the region at time step 45. The above assumptions still satisfy drone dynamics in Section 4.1, since there is no more than one drone that exists simultaneously. The initial state vectors of both drones both follow $r(\boldsymbol{b}) = \mathcal{U}(\boldsymbol{a}, \boldsymbol{c})$, where $\boldsymbol{a} = [-150, 2, -150, 2]^T$, $\boldsymbol{c} = [150, 2, 150, 2]^T$. The dynamics of both drones follow Equations (8), (36), and (37). The first user participates in the tasks all the time, while the

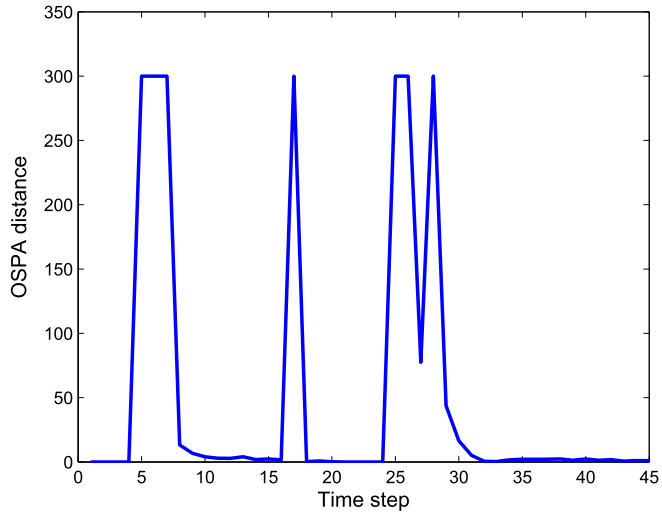

Fig. 16. OSPA distance (linear case with two drones and mobile users).

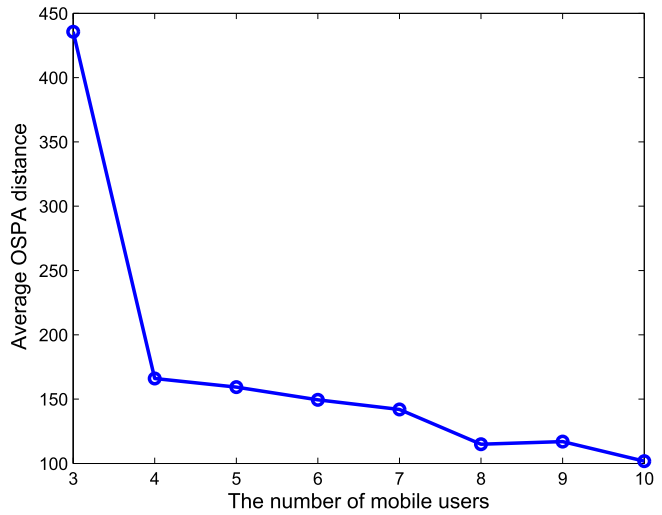

Fig. 17. The OSPA distance (accuracy versus the number of mobile users).

second user and the third user participate in the tasks during time steps 1–30 and steps 20–45, respectively. The remaining settings are the same as those in Section 6.2.

The results are presented in Figures 14–16. It can be observed that CSDrone needs several time steps to discover drones entering the region. However, both the number and the trajectories are accurately estimated after those drones are detected.

6.5 Accuracy versus the Number of Mobile Users

In this case, we investigate the accuracy of joint detection and estimation when the number of mobile users is different. We assume that mobile users remain static and participate in the tasks all the time. To maximally alleviate the effect caused by their positions, let us assume that their

positions are randomly generated in the region. The number of mobile users increases from 3 to 10. The remaining settings are the same as those in Section 6.1.

One hundred Monte Carlo trials are run and the average OSPA distance is presented in Figure 17. One can easily observe that as the number of mobile users increases, so does the accuracy of joint detection and estimation.

7 CONCLUSION

Drones (UAVs) are likely to be more widely used in our society, both autocratic and liberal, for the foreseeable future. This is evidenced by the increasing interest in drone surveillance systems.

In this article, we proposed a cyber-physical system for drone surveillance. Using crowdsourced user-contributed data, the proposed system significantly reduces the operation costs, particularly in comparison to existing competing systems. To ensure efficient and effective drone detection and tracking, the proposed system utilized the RFS theory and RFS-based Bayesian filter. Using extensive numerical results, we demonstrated the utility of the proposed system.

Future research includes collaborating with an organization, such as the authors' institutions police department (e.g., UTSA Police Department) or a city council to deploy the system. This will allow us to evaluate its utility, performance, and scalability in a real-world environment (e.g., its efficiency and effectiveness in detecting and tracking multiple drones in real-time), as well as identifying any limitations that can be addressed in subsequent versions.

APPENDIX

A THE PROOF OF EQUATIONS (28) AND (29)

Proof. Based on the set p.d.f. of $\mathbf{\Phi}_{i,k}$, i.e., $\phi_{i,k}(\mathbf{\Phi}_{i,k}|\Sigma_k, \mathbf{\Theta}_k)$, and

$$\mathbf{\Phi}_k = \mathbf{\Phi}_{1,k} \cup \mathbf{\Phi}_{2,k} \cup \cdots \cup \mathbf{\Phi}_{n(k),k}, \tag{45}$$

$\phi_k(\mathbf{\Phi}_k|\Sigma_k, \mathbf{\Theta}_k)$ can be obtained by using RFS convolution formula (see chapter 11 in Reference [17]). However, here we present a more straightforward way to deduce $\phi_k(\mathbf{\Phi}_k|\Sigma_k, \mathbf{\Theta}_k)$.

Let us begin with the case in which Σ_k is empty. If $\mathbf{\Phi}_k$ also is empty, then it is straightforward that

$$\mathbf{\Phi}_{i,k} = \emptyset, \quad i = 1, 2, \ldots, n(k). \tag{46}$$

Therefore,

$$\phi_k(\emptyset|\emptyset, \mathbf{\Theta}_k) = (1 - p_f)^{n(k)}. \tag{47}$$

If $|\mathbf{\Phi}_k| = j$, then we need to assign j of $n(k)$ reports to false alarms and $n(k) - j$ reports to "no detection." Thus,

$$\phi_k(\mathbf{\Phi}_k|\emptyset, \mathbf{\Theta}_k) = \frac{n_k!}{(n_k - j)!}(p_f \kappa(\vartheta))^j (1 - p_f)^{(n_k - j)}. \tag{48}$$

Now, we are in a position to deduce $\phi_k(\mathbf{\Phi}_k|\Sigma_k, \mathbf{\Theta}_k)$ when Σ_k is not an empty set. If $\mathbf{\Phi}_k$ still is empty, then clearly Equation (46) holds true, and

$$\phi_k(\emptyset|\Sigma_k, \mathbf{\Theta}_k) = (1 - q)^{n(k)}.$$

If $|\mathbf{\Phi}_k| = 1$, then we need to assign 1 of $n(k)$ reports to detection (true detection or false alarms) and $n(k) - 1$ reports to "no detection." Similar to Equation (48),

$$\phi_k(\mathbf{\Phi}_k|\Sigma_k, \mathbf{\Theta}_k) = q(1 - q)^{n_k - 1} \left(\lambda(\vartheta, \theta_{1,k}) + \cdots + \lambda(\vartheta, \theta_{n_k,k})\right).$$

If $|\mathbf{\Phi}_k| = 2$, then we need to assign 2 of $n(k)$ reports to detection (true detection and false alarms) and $n(k) - 2$ reports to no detection. In a similar manner, we have

$$\phi_k(\mathbf{\Phi}_k|\Sigma_k, \mathbf{\Theta}_k) = q^2(1 - q)^{n_k - 2} \sum_{1 \le i_1 \ne \cdots \ne i_j \le n_k} \lambda(\vartheta, \theta_{i_1,k})\lambda(\vartheta, \theta_{i_2,k}).$$

Thus, if $|\Phi_k| = j$, then we have

$$\phi_k(\Phi_k|\Sigma_k, \Theta_k) = q^j(1-q)^{(n_k-j)} \sum_{1 \le i_1 \ne \cdots \ne i_j \le n_k} \lambda(\vartheta, \theta_{i_1,k}) \cdots \lambda(\vartheta, \theta_{i_j,k}).$$

This completes the proof. □

REFERENCES

[1] en.people.cn. 2017. Drone flying over SW China airport affects flights. Retrieved from http://en.people.cn/n3/2017/0502/c90000-9209828.html.

[2] Blighter. 2017. AUDS: Anti-UAV Defence System. Retrieved from http://www.blighter.com/products/auds-anti-uav-defence-system.html.

[3] sUAS News. 2017. DJI Drones Cannot Evade Drone Detection-DroneTracker. Retrieved from https://www.suasnews.com/2017/04/dronetracker-dji-drones-cannot-evade-drone-detection/.

[4] Amazon. 2018. Amazon prime air. Retrieved from https://www.amazon.com/Amazon-Prime-Airl.

[5] M. M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin. Jan. 2018. Key technologies and system trade-offs for detection and localization of amateur drones. *IEEE Commun. Mag.* 56, 1 (Jan. 2018), 51–57.

[6] F. Christnacher, S. Hengy, M. Laurenzis, A. Matwyschuk, P. Naz, S. Schertzer, and G. Schmitt. 2016. Optical and acoustical UAV detection. In *Proceedings of the International Society for Optics and Photonics (SPIE'16)*. 1–13.

[7] S. Dinan. [n.d.]. Drones become latest tool drug cartels use to smuggle drugs into U.S. Retrieved from https://www.washingtontimes.com/news/2017/aug/20/mexican-drug-cartels-using-drones-to-smuggle-heroi/.

[8] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y. Yao. 2018. An amateur drone surveillance system based on the cognitive Internet of Things. *IEEE Commun. Mag.* 56, 1 (Jan. 2018), 29–35.

[9] H. Fu, S. Abeywickrama, L. Zhang, and C. Yuen. 2018. Low-complexity portable passive drone surveillance via SDR-based signal processing. *IEEE Commun. Mag.* 56, 4 (Apr. 2018), 112–118.

[10] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak. 2018. Detection, tracking, and interdiction for amateur drones. *IEEE Commun. Mag.* 56, 4 (Apr. 2018), 75–81.

[11] T. Humphreys. 2015. Statement on the security threat posed by unmanned aerial systems and possible countermeasures. *Subcommittee on Oversight and Management Efficiency of the House Committee on Homeland Security*. 1–9.

[12] C. Knapp and G. Carter. 1976. The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process.* 24, 4 (Aug. 1976), 320–327.

[13] N. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, and K.-K. R. Choo. 2018. Smart vehicle forensics: Challenges and case study. *Future Gen. Comput. Syst.* (2018). DOI: 10.1016/j.future.2018.05.081

[14] T. Li, M. Bolic, and P. M. Djuric. 2015. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* 32, 3 (May 2015), 70–86.

[15] C. Lin, D. He, N. Kumar, K.-K. R. Choo, A. V. Vinel, and X. Huang. 2018. Security and privacy for the Internet of drones: Challenges and solutions. *IEEE Commun. Mag.* 56, 1 (Jan. 2018), 64–69.

[16] H. Ma, D. Zhao, and P. Yuan. 2014. Opportunities in mobile crowd sensing. *IEEE Commun. Mag.* 52, 8 (Aug. 2014), 29–35.

[17] R. P. S. Mahler. 2007. *Statistical Multisource-Multitarget Information Fusion.* Artech House, Norwood, MA.

[18] K. Moskvitch. 2015. Take off: Are drones the future of farming? *Engineering and Technology* 10, 7 (Aug. 2015), 162–166.

[19] W. Ripley. [n.d.]. Drone with radioactive material found on Japanese Prime Minister's roof. Retrieved from https://edition.cnn.com/2015/04/22/asia/japan-prime-minister-rooftop-drone/index.html.

[20] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina. 2013. A tutorial on Bernoulli filters: Theory, implementation, and applications. *IEEE Trans. Signal Process.* 61, 13 (July 2013), 3406–3430.

[21] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. 2008. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* 56, 8 (Aug. 2008), 3447–3457.

[22] V. Sharma, D. N. K. Jayakody, I. You, R. Kumar, and J. Li. 2018. Secure and efficient context-aware localization of drones in urban scenarios. *IEEE Commun. Mag.* 56, 4 (Apr. 2018), 120–128.

[23] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen. 2018. Anti-drone system with multiple surveillance technologies: Architecture, implementation and challenges. *IEEE Commun. Mag.* 56, 4 (Apr. 2018), 68–74.

[24] Z. Shi, C. Zhou, Y. Gu, N. A. Goodman, and F. Qu. 2018. Source estimation using coprime array: A sparse reconstruction perspective. *IEEE Sensors J.* 17, 13 (Feb. 2018), 755–765.

[25] D. Solomitckii, M. Gapeyenko, V. Semkin, S. Andreev, and Y. Koucheryavy. 2018. Technologies for efficient amateur drone detection in 5G millimeter-wave cellular infrastructure. *IEEE Commun. Mag.* 56, 1 (Jan. 2018), 43–50.

[26] G. D. L. Torre, P. Rad, and K.-K. R. Choo. 2018. Driverless vehicle security: Challenges and future research opportunities. *Future Gen. Comput. Syst.* (2018). DOI: 10.1016/j.future.2017.12.041

[27] B.-T. Vo and B.-N. Vo. 2013. Labeled random finite sets and multi-object conjugate priors. *IEEE Trans. Signal Process.* 61, 13 (July 2013), 3460–3475.
[28] B.-T. Vo, B.-N. Vo, and A. Cantoni. 2009. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Process.* 57, 2 (Feb. 2009), 409–423.
[29] P. Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.* 15, 2 (June. 1967), 70–73.
[30] C. Yang, Z. Shi, K. Han, J. J. Zhang, Y. Gu, and Z. Qin. 2018. Optimization of particle CBMeMBer filters for hardware implementation. *IEEE Trans. Vehic. Technol.* 67, 9 (Sept. 2018), 9027–9031.
[31] C. Yang, Z. Shi, H. Zhang, J. Wu, and X. Shi. 2019. Multiple attacks detection in cyber-physical systems using random finite set theory. *IEEE Trans. Cybernet.* (2019). DOI: 10.1109/TCYB.2019.2912939
[32] S. P. Yeong, L. M. King, and S. S. Dol. 2015. A review on marine search and rescue operations using unmanned aerial vehicles. *Int. Scholar. Sci. Res. Innovat.* 9, 2 (2015), 396–399.
[33] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang. 2018. Direction-of-arrival estimation for coprime array via virtual array interpolation. *IEEE Trans. Signal Process.* 66, 22 (Nov. 2018), 5956–5971.
[34] C. Zhou, Y. Gu, S. He, and Z. Shi. 2018. A robust and efficient algorithm for coprime array adaptive beamforming. *IEEE Trans. Vehic. Technol.* 67, 2 (Feb. 2018), 1099–1112.
[35] C. Zhou, Y. Gu, Z. Shi, and Y. M. Zhang. 2018. Off-grid direction-of-arrival estimation using coprime array interpolation. *IEEE Signal Process. Lett.* 25, 11 (Nov. 2018), 1710–1714.
[36] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi. 2015. Efficient road detection and tracking for unmanned aerial vehicle. *IEEE Trans. Intell. Transport. Syst.* 16, 1 (Feb. 2015), 297–309.

Received September 2018; revised January 2019; accepted May 2019