HopSAC: Frequency Hopping Parameter Estimation Based on Random Sample Consensus for Counter-Unmanned Aircraft Systems

Brandon F. Lo, Scott Torborg, and Chun Kin Au Yeung
SkySafe, Inc.
San Diego, CA 92104
{brandon, scott, chunkin}@skysafe.io

***Abstract*—Small unmanned aircraft systems (UAS), commonly known as drones and widely used in recreational and commercial applications, have caused alarming concerns of public safety and homeland security due to frequently reported unauthorized UAS incidents in recent years. To effectively disable potential threats from frequency hopping drones and controllers, the counter attack of Counter-UAS (CUAS) systems typically require parameter estimation of the frequency hopping signals with high precision and low complexity for real-time responses. Therefore, a model parameter estimation method to meet all these requirements becomes a challenge for CUAS systems. In this paper, a novel hopping parameter estimation method based on random sample consensus called HopSAC is proposed to conquer this challenge. Given a small set of samples, HopSAC estimates the parameters of linear frequency hopping sequence and achieves high multiple target detection performance with low implementation complexity that can be realized in real time. Simulation results show that the proposed HopSAC significantly outperforms linear Least Squares method in achieving exceptional accuracy of model parameter estimation under the impact of gross errors, timing errors, and multiple UAS targets.**

I. Introduction

In recent years, small unmanned aircraft systems (UAS), commonly known as drones, have gained a lot of popularity due to their affordability and versatility. They have been widely used in many applications from recreational flying such as drone racing to commercial uses such as package delivery and real estate photography. According to FAA Forecast [1], the non-model (commercial) UAS will grow three-fold from 2018 to 2023 whereas the number of model (recreational) UAS will increase from 1.25 to 1.39 million units in 5 years. However, unauthorized UAS activities and incidents have been reported more and more frequently near airports, stadiums, and borders, which has caused growing concerns about public safety and homeland security. Therefore, an effective Counter-UAS (CUAS) becomes an indispensable mechanism for law enforcement and military to detect, identify, and disable any potential and imminent threats caused by the improper and unauthorized uses of drones.

A CUAS is typically capable of detecting any wireless signal transmitted from drones and remote controllers (RCs) when they are in the detection range, and determining when in time and where in frequency the signals from drones are detected. Nevertheless, the time and frequency information is subject to errors due to sensing and measurement errors, channel impairments such as fading and interference, and hardware limitations such as carrier frequency offsets and timing jitters. One of the main challenges is to identify the frequency hopping parameters from these noisy time-frequency samples. The estimation of these parameters is further complicated by multi-target scenarios where multiple drones are present.

There are many frequency hopping parameter estimation methods based on a variety of techniques: dynamic programming and low-rank trilinear decomposition [2], sparse linear regression [3], [4], sparse Bayesian method [5], and orthogonal matching pursuit [4]. The implementation complexity of these methods is generally high. Thus, they are not suitable for real-time applications that have critical response time constraints.

In this paper, we propose a novel frequency hopping parameter estimation method called HopSAC based on the random sample consensus algorithm. Random sample consensus (RANSAC) [6] is known as a paradigm for estimating parameters of a model by fitting the model to observation data. Similar to RANSAC, HopSAC outperforms linear Least Squares (LS) method for its excellent capability of rejecting gross errors with only a small subset of samples. Gross errors, also known as outliers, are data samples significantly deviating from the majority of observations. The ability of HopSAC to discard gross errors allows the detector to tolerate higher false alarm rate for detection at a lower signal-to-noise ratio (SNR). Moreover, HopSAC is very robust to timing and frequency errors, enabling the jammer to transmit much more precise pulses in time, and thus improving power efficiency and reducing collateral damage. To the best of our knowledge, HopSAC is the first work to address the challenges of multi-target frequency hopping parameter estimation using the paradigm of RANSAC. Our simulation results show that the proposed HopSAC provides an effective way for CUAS systems to achieve high performance of estimation accuracy with low complexity in the real-time parameter estimation of linear frequency hopping drones and controllers.

In the remainder of this paper, Section II describes the system model of a CUAS system and the linear frequency hopping model for fitting the time-frequency samples. Section III discusses the proposed HopSAC algorithm for multi-target hopping parameter estimation. Section IV demonstrates

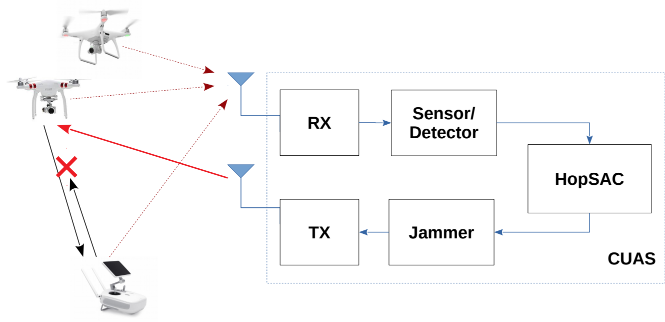

Fig. 1. Simplified CUAS System Model.

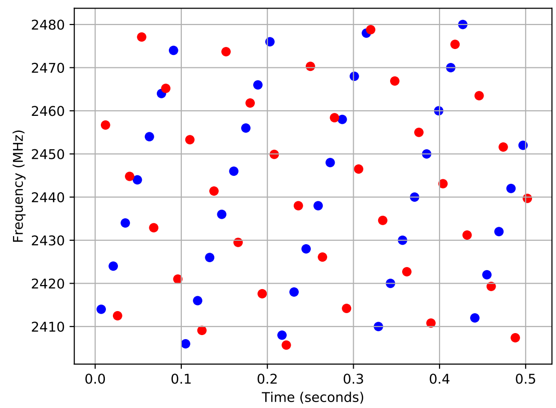

Fig. 2. Example of Frequency Hopping Sequences.

the performance of HopSAC under the impact of gross errors, timing errors, and multiple targets. Finally, Section V concludes the paper.

II. System Model

Fig. 1 shows the simplified system model of a CUAS system. The CUAS receiver (RX) receives wideband frequency-hopping signals from all drones or RCs in the detection range, and stores signal parameters in pulse descriptor words (PDW) after preprocessing to clusters by emitter type or angle of arrival, level control with power squelch, and correlation by matched filtering. A detector then determines the presence of the signal at time $\tilde{t}_n$ and frequency $\tilde{f}_n$, where n is the index of the time-frequency samples, and sends a set of time-frequency samples $\{(\tilde{t}_n, \tilde{f}_n)\}$ every T_s seconds to the HopSAC estimator for parameter estimation. HopSAC estimator then sends model parameters to configure the jammer. The transmitter (TX) sends interference signals for counter attack.

Let $\mathbf{x} = \{x_n\}$ be a sequence of 2-tuples $x_n = (\tilde{t}_n, \tilde{f}_n)$ representing the time-frequency observations of frequency-hopping signals. If t_n and f_n are the time and the frequency, respectively, of UAS transmissions, we have timing errors $t_{\delta,n} = \tilde{t}_n - t_n$ and frequency errors $f_{\delta,n} = \tilde{f}_n - f_n$. We assume that $t_{\delta,n}$ and $f_{\delta,n}$ are independent and identically distributed (i.i.d.) zero mean Gaussian variables with variance σ_t^2 and σ_f^2, respectively. That is, $t_{\delta,n} \sim \mathcal{N}(0, \sigma_t^2)$ and $f_{\delta,n} \sim \mathcal{N}(0, \sigma_f^2)$. In addition to measurement errors, we assume that the occurrence of gross errors follows a Poisson distribution with parameter λ where λ is the average number of gross errors per time interval T_s known as outlier arrival rate. Therefore, each sample x_n could be the transmission of a real target or a false alarm caused by a spurious source or hardware impairments.

In this paper, we consider the parameter estimation of linear frequency hopping models. Although the proposed HopSAC can be applied to arbitrary linear models, certain linear hopping sequences such as linear feedback shift register (LFSR)-based pseudo random hopping sequences require a large number of samples for accurate fit due to a large number of hopping states. Thus, in this work, based on the linear hopping behaviors of UAS systems such as [7]–[9], we model a linear hopping sequence f_m as

$$f_m = (f_0 - f_b + m \cdot f_d \mod B) + f_b, \quad m = 0, 1, \ldots \quad (1)$$

where f_0 is the start frequency of the first hop after timestamp 0, f_b is the center frequency of the lowest channel, f_d is frequency difference or frequency spacing between two neighboring hops, $B = N_c \cdot b$ is the total bandwidth of all channels, N_c is the number of channels used in the hopping sequence, and b is the channel bandwidth, which is assumed to be known. Note that not all hops can be observed and detected by the system. If we map channel center frequencies to channel numbers starting from channel 0, (1) can be simplified to

$$c_m = c_0 + m \cdot c_d \mod N_c, \quad m = 0, 1, \ldots \quad (2)$$

where $c_m = (f_m - f_b)/b$, $c_0 = (f_0 - f_b)/b$, and $c_d = f_d/b$. c_d is also known as hop number or hop count. The corresponding timestamp t_m of each hop is given by

$$t_m = t_0 + m \cdot t_d, \quad m = 0, 1, \ldots \quad (3)$$

where t_0 is the start time of the first hop and t_d is the time interval between two neighboring hops known as hop period. Hence, the frequency hopping parameter model contains five parameters: i) time start t_0, ii) frequency start f_0 or channel start c_0, iii) hop period t_d, iv) frequency spacing f_d or hop count c_d, and v) number of channels N_c.

The operating frequencies of small model drones are usually known and published in FCC test reports. For example, in [7], 39 channels are defined from 2404 MHz to 2480 MHz in 2.4 GHz band for DJI GL300A remote controller. The differences of operating frequencies can be derived from all possible center frequency of channels. Fig. 2 shows an example of frequency hopping sequences from DJI remote controllers GL300A [7] (blue dots: $f_b = 2404$ MHz, $t_0 = 0.007$ s, $f_0 = 2414$ MHz, $t_d = 0.014$ s, $f_d = 10$ MHz, $N_c = 39$, and $b = 2$ MHz), and GL300F [8] (red dots: $f_b = 2404$ MHz, $t_0 = 0.012$ s, $f_0 = 2456.7$ MHz, $t_d = 0.014$ s, $f_d = 32.3$ MHz, $N_c = 45$, and $b = 1.7$ MHz).

Algorithm 1 HopSAC Model Parameter Estimation

procedure HopSAC($\mathbf{x}, \mathcal{F}, N_r, n_s, \tilde{t}_d, \bar{\delta}_t, \bar{\delta}_f, \bar{\gamma}, s_f$)
$\mathcal{T} \leftarrow \emptyset$
repeat
    $\mathcal{M}, \mathcal{I}, \mathcal{O} \leftarrow \emptyset$
    for $r = 1$ **to** N_r **do**
        for each $\mathcal{F}_i \in \mathcal{F}$ **do**
            $\mathcal{M}' \leftarrow$ Estimate($\mathbf{x}, n_s, \tilde{t}_d, \mathcal{F}_i, \bar{\delta}_t, \bar{\delta}_f, s_f$)
            if $\mathcal{M}' \neq \emptyset$ **then**
                $\mathcal{I}', \mathcal{O}' \leftarrow$ Classify($\mathbf{x}, \mathcal{M}', \bar{\gamma}, s_f$)
                if $|\mathcal{I}'| > |\mathcal{I}|$ **then**
                    $\mathcal{M} \leftarrow \mathcal{M}', \mathcal{I} \leftarrow \mathcal{I}', \mathcal{O} \leftarrow \mathcal{O}'$
                end if
            end if
        end for
    end for
    if $\mathcal{M} \neq \emptyset$ **then**
        $\mathcal{T} \leftarrow \mathcal{T} \bigcup \mathcal{M}$, $\mathbf{x} \leftarrow \mathcal{O}$
    end if
until $\mathcal{M} \equiv \emptyset$ or $\|\mathbf{x}\| \leq n_s$
end procedure

III. Multi-Target Hopping Parameter Estimation Based on RANSAC (HopSAC)

The proposed HopSAC parameter estimation method estimates the parameters of frequency-hopping sequences used by remote controllers and/or drones. It receives a vector of time-frequency samples $\mathbf{x}$ of length $\|\mathbf{x}\| = N_s$ from the detector each time interval T_s, and performs the fitting of the parameter model to these samples by using the RANSAC paradigm. The HopSAC algorithm starts with randomly selected n_s samples from $\mathbf{x}$, uses them to estimate the parameters, constructs the model using these parameters, and classifies the samples as inliers or outliers by evaluating their fitting errors. Here inliers refer to the samples closely matching the model with negligible or no fitting errors whereas outliers are samples resulting in large errors (gross errors) from the fitting. This process is repeated for N_r trials. The best model, $\mathcal{M}$, is the one with the largest number of inliers $|\mathcal{I}|$ as the largest group of samples, $\mathcal{I}$, reaches the consensus. After $\mathcal{M}$ is obtained for the first target and stored in the target set $\mathcal{T}$, the set of inliers is subtracted from the set of samples because these inliers already formed the consensus group associated with the first target. As a result, the set of outliers $\mathcal{O}$ is used as input samples $\mathbf{x}$ to find the best model of the second target. This process is repeated until no new target is found or the number of outliers remaining in the sample set is not enough. At the end, $\mathcal{T}$ contains the models of all detected targets. Algorithm 1 outlines the proposed HopSAC algorithm.

Let $\mathcal{F} = \bigcup_i \mathcal{F}_i$ be a set of publicly known and well defined channel center frequencies $\mathcal{F}_i$ for frequency hopping device type i. The frequency differences of any two defined channel frequencies and the total number of channels for a given device type i are also known based on the prior knowledge of $\mathcal{F}_i$.

Algorithm 2 HopSAC Estimate and Classify Procedures

procedure Estimate($\mathbf{x}, n_s, \tilde{t}_d, \mathcal{F}_i, \bar{\delta}_t, \bar{\delta}_f, s_f$)
$\{(\tilde{t}_j, \tilde{f}_j)\} \leftarrow$ RandomSelect($\mathbf{x}, n_s$), $\mathcal{M}' \leftarrow \emptyset$
$\mathcal{D}_i \leftarrow \{|f_j - f_k|, j \neq k, f_j, f_k, \in \mathcal{F}_i\}$
$\mathcal{D}_t \leftarrow \{d_{t,j,k} \mid |\tilde{t}_j - \tilde{t}_k|, \tilde{t}_j < \tilde{t}_k, j < k\}$
$\mathcal{D}_f \leftarrow \{d_{f,j,k} \mid |\tilde{f}_j - \tilde{f}_k|, \tilde{t}_j < \tilde{t}_k, j < k\}$
for each $d_{t,j,k} \in \mathcal{D}_t$ **do**
    $n_{h,j,k} \leftarrow \lfloor d_{t,j,k}/\tilde{t}_d \rceil$
    $\hat{t}_{d,j,k} \leftarrow d_{t,j,k}/n_{h,j,k}$
    $\hat{t}_{0,j,k} \leftarrow ((s_f \cdot t_j) \mod (s_f \cdot \hat{t}_{d,j,k}))/s_f$
    if $|d_{f,j,k} - d_i| < \bar{\delta}_f, d_i \in \mathcal{D}_i$ **then**
        $\hat{f}_{d,j,k} \leftarrow f_{d,j,k}$ that satisfies (7)
        $n_{0,j,k} = \lfloor (t_j - \hat{t}_{0,j,k})/\hat{t}_{d,j,k} \rceil$
        $\hat{f}_{0,j,k} = ((f_j - f_b - n_{0,j,k} \cdot \hat{f}_{d,j,k}) \mod B) + f_b$
        if (10), (11), (12), (13) are all true **then**
            $\mathcal{M}' \leftarrow \{\hat{t}_d, \hat{t}_0, \hat{f}_d, \hat{f}_0, \hat{N}_c\}$, $\hat{N}_c \leftarrow |\mathcal{F}_i|$
        end if
    end if
end for
end procedure
procedure Classify($\mathbf{x}, \mathcal{M}', \bar{\gamma}, s_f$)
$\mathcal{I}', \mathcal{O}' \leftarrow \emptyset$, $\{\hat{t}_d, \hat{t}_0, \hat{f}_d, \hat{f}_0\} \leftarrow \mathcal{M}'$
for $k = 1$ **to** $\|\mathbf{x}\|$ **do**
    $(\tilde{t}_k, \tilde{f}_k) \leftarrow \mathbf{x}$
    $\epsilon_{t,k} \leftarrow (\lfloor \tilde{t}_k \cdot s_f \rceil \mod \lfloor \hat{t}_d \cdot s_f \rceil)/(s_f - \hat{t}_0)$
    $h_{p,k} \leftarrow \lfloor \tilde{t}_k - \hat{t}_0 \rceil / \hat{t}_d$
    $\hat{f}_k \leftarrow ((\hat{f}_0 - f_b + h_{p,k} \cdot \hat{f}_d) \mod B) + f_b$
    $\epsilon_{f,k} \leftarrow \hat{f}_k - \tilde{f}_k$
    $\epsilon_k \leftarrow \sqrt{\epsilon_{t,k}^2 + \epsilon_{f,k}^2}$
    if $\epsilon_k < \bar{\gamma}$ **then**
        $\mathcal{I}' \leftarrow \mathcal{I}' \bigcup \{x_k\}$
    else
        $\mathcal{O}' \leftarrow \mathcal{O}' \bigcup \{x_k\}$
    end if
end for
end procedure

In each trial r, we estimate the model parameters of each device type i under consideration using the prior knowledge $\mathcal{F}_i$, and obtain a parameter model $\mathcal{M}'$ of this device type. By using the model parameters in $\mathcal{M}'$, we calculate the fitting errors of all samples and classify each sample as inliers or outliers depending on their fitting errors. This model $\mathcal{M}'$ will be compared to the best known model $\mathcal{M}$ if $\mathcal{M}$ is found in previous trials. $\mathcal{M}$ will be updated with $\mathcal{M}'$ if the number of inliers, $|\mathcal{I}'|$, of $\mathcal{M}'$ is greater. This means that model $\mathcal{M}'$ is more likely to be the best estimate due to a larger number of inliers reaching the consensus. At the end of all trials, the best model remaining in $\mathcal{M}$ will be the model of a detected target with a specific set of hopping parameters. Those two important processing steps in each trial: model parameter estimation and inliers/outliers classification are described in detail as follows. Algorithm 2 outlines these two procedures.

A. Model Parameter Estimation

The model parameter estimation procedure estimates model parameters using randomly selected samples and the knowledge of $\mathcal{F}_i$. It starts with n_s randomly selecting samples from the observations in time interval T_s. For linear hopping model defined in Section II, it is sufficient to have only two samples ($n_s = 2$) randomly selected each time for parameter estimation. In general, n_s samples are first randomly selected: $(\tilde{t}_j,\ \tilde{f}_j)$, $j = 1, 2, \ldots, n_s$. The sets of time differences and frequency differences between any of two samples are $\mathcal{D}_t = \{d_{t,j,k} \mid |\tilde{t}_j - \tilde{t}_k|, j < k\}$ and $\mathcal{D}_f = \{d_{f,j,k} \mid |\tilde{f}_j - \tilde{f}_k|, \tilde{t}_j < \tilde{t}_k, j < k\}$, respectively. There are $n_c = \binom{n_s}{2}$ elements in $\mathcal{D}_t$ and in $\mathcal{D}_f$.

We first find the new estimates of t_d and t_0. Let the coarse estimate of t_d from the sensor/detector be $\tilde{t}_d$. For each $d_{t,j,k} \in \mathcal{D}_t$, the number of hop intervals between samples $(\tilde{t}_j, \tilde{f}_j)$ and $(\tilde{t}_k, \tilde{f}_k)$ is given by

$$n_{h,j,k} = \lfloor d_{t,j,k}/\tilde{t}_d \rceil \tag{4}$$

where $\lfloor\cdot\rceil$ denotes a rounding function. The hop period estimate for each $d_{t,j,k}$ is obtained as

$$\hat{t}_{d,j,k} = d_{t,j,k}/n_{h,j,k}. \tag{5}$$

The start time of the first hop can then be estimated as

$$\hat{t}_{0,j,k} = ((s_f \cdot t_j) \mod (s_f \cdot\ \hat{t}_{d,j,k}))/s_f \tag{6}$$

where s_f is the stability factor for numerical stability.

We next find the new estimates of f_0 and f_d. From $\mathcal{F}_i$, the set of differences of operating frequencies in $\mathcal{F}_i$ is $\mathcal{D}_i = \{|f_j - f_k|, j \neq k,\ f_j, f_k, \in \mathcal{F}_i\}$. For each $d_{f,j,k} \in \mathcal{D}_f$, if $d_{f,j,k}$ matches any elements in $\mathcal{D}_i$ within a small error $\bar{\delta}_f$, the estimate of frequency difference between hops, $\hat{f}_{d,j,k}$, is the $f_{d,j,k}$ that satisfies

$$f_k = ((f_j - f_b + n_{h,j,k} \cdot f_{d,j,k}) \mod B) + f_b. \tag{7}$$

To find the start frequency $f_{0,j,k}$, we first find the number of hops from $(\tilde{t}_j, \tilde{f}_j)$ to the first sample, $n_{0,j,k}$, given by

$$n_{0,j,k} = \lfloor(\tilde{t}_j - \hat{t}_{0,j,k})/\hat{t}_{d,j,k}\rceil \tag{8}$$

The start frequency $f_{0,j,k}$ can be estimated as

$$\hat{f}_{0,j,k} = ((f_j - f_b - n_{0,j,k} \cdot \hat{f}_{d,j,k}) \mod B) + f_b. \tag{9}$$

If the estimates in the same group are the same or only differ in a negligible value, they are valid estimates and a model is found. That is,

$$\hat{t}_0 = \Sigma_{j,k}\hat{t}_{0,j,k}/n_c \quad \text{if} \quad |\hat{t}_{0,j,k} - \hat{t}_{0,m,n}| < \bar{\delta}_t \tag{10}$$

$$\hat{t}_d = \Sigma_{j,k}\hat{t}_{d,j,k}/n_c \quad \text{if} \quad |\hat{t}_{d,j,k} - \hat{t}_{d,m,n}| < \bar{\delta}_t \tag{11}$$

$$\hat{f}_0 = \Sigma_{j,k}\hat{f}_{0,j,k}/n_c \quad \text{if} \quad |\hat{f}_{0,j,k} - \hat{f}_{0,m,n}| < \bar{\delta}_f \tag{12}$$

$$\hat{f}_d = \Sigma_{j,k}\hat{f}_{d,j,k}/n_c \quad \text{if} \quad |\hat{f}_{d,j,k} - \hat{f}_{d,m,n}| < \bar{\delta}_f \tag{13}$$

$j \neq m,\ k \neq n\ \forall j, k, n, m$. Clearly, $\hat{t}_0 = \hat{t}_{0,j,k}$, $\hat{t}_d = \hat{t}_{d,j,k}$, $\hat{f}_0 = \hat{f}_{0,j,k}$, $\hat{f}_d = \hat{f}_{d,j,k}$ if $n_c = 1$ ($n_s = 2$). The number of channels $\hat{N}_c$ is simply $|\mathcal{F}_i|$. Finally, we obtain the new model $\mathcal{M}' = \{\hat{t}_d, \hat{t}_0, \hat{f}_d, \hat{f}_0, \hat{N}_c\}$.

TABLE I
SIMULATION PARAMETERS

Symbols	Values	Description
T_s	0.5 s	Sample acquiring time interval
N_s	35	Number of input samples in T_d
n_s	2	Number of randomly selected samples
N_t	1-10	Number of targets
$\tilde{t}_d$	14 ± 0.1 ms	Coarse hop period
$\bar{\delta}_t$	0.001 ms	Hop period threshold
$\bar{\delta}_f$	100 Hz	Frequency difference threshold
$\bar{\gamma}$	0.01	Fitting error threshold
N_r	100	Number of trials
N_{iter}	10,000	Number of Monte Carlo iterations
s_f	1000	Stability factor
$\mathcal{F}_i$	See [7]–[9]	Channel definition of device type i
N_c	See [7]–[9]	Number of channels

B. Classification of Inliers and Outliers

The classification process starts with the fitting error evaluation, which calculates the fitting errors of all samples based on the current model parameters in $\mathcal{M}'$. For each sample k, we find time fitting error $\epsilon_{t,k}$ and frequency fitting error $\epsilon_{f,k}$ separately. The time fitting error $\epsilon_{t,k}$ can be calculated with the assistance of s_f for numerical stability as

$$\epsilon_{t,k} = (\lfloor\tilde{t}_k \cdot s_f\rceil \mod \lfloor\hat{t}_d \cdot s_f\rceil)/(s_f - \hat{t}_0). \tag{14}$$

To find frequency fitting error $\epsilon_{f,k}$, we first determine hop phase h_p, which is the number of hop intervals from the beginning of the sequence to the current sample $(\tilde{t}_k, \tilde{f}_k)$ given by

$$h_{p,k} = \lfloor\tilde{t}_k - \hat{t}_0\rceil/\hat{t}_d \tag{15}$$

We next estimate the frequency of the current sample based on model parameters as

$$\hat{f}_k = ((\hat{f}_0 - f_b + h_{p,k} \cdot \hat{f}_d) \mod B) + f_b \tag{16}$$

The frequency fitting error is then $\epsilon_{f,k} = \hat{f}_k - \tilde{f}_k$. The combined fitting error ϵ is hence determined by the square root of the sum of time error squared and frequency error squared. That is, $\epsilon_k = \sqrt{\epsilon_{t,k}^2 + \epsilon_{f,k}^2}$. The fitting error of each sample, ϵ_k, is compared to a fitting error threshold $\bar{\gamma}$. If ϵ_k is smaller than $\bar{\gamma}$, sample x_k is an inlier and is included in the set of inliers $\mathcal{I}'$, otherwise, it is an outlier, which is included in the set of outliers $\mathcal{O}'$.

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed HopSAC algorithm for hopping parameter estimation using Monte Carlo simulation with $N_{\text{iter}} = 10,000$ iterations, and compare the performance of HopSAC to that of linear LS estimation under the influence of gross errors, timing errors, and multiple targets. The parameter settings in simulations are tabulated in Table I.

We consider three linear frequency hopping devices [7]–[9] in simulations. The spectrogram in 2.4 GHz band shows that the coarse time period $\tilde{t}_d$ between hops is roughly 14 ms. In each iteration, we generate N_t targets by randomly selecting N_t devices from these three devices under consideration. The frequency hopping sequence of each target is configured by parameters $t_0 \in [0, 0.014)$, $t_d \in [0.01399, 0.0141]$, $f_0 \in \mathcal{F}_i$, $f_d \in \mathcal{F}_i$, $N_c = |\mathcal{F}_i|$ where $\mathcal{F}_i$ is determined by the selected device i. Note that the same device may be selected multiple times as multiple targets with different hopping parameters and hopping behavior. We combine hopping samples from all targets, add N_o outliers where N_o is determined by Poisson process with outlier arrival rate λ, randomly place them in time interval T_s and in frequency range defined in $\mathcal{F}$, and adjust t_k with timing error following Gaussian distribution $\mathcal{N}(0, \sigma_t^2)$. Finally, all time-frequency samples are sorted in time domain to form the input samples.

We evaluate the performance of HopSAC and LS estimators by using the metric: target detection probability P_d defined as

$$P_d = \frac{1}{N_{\text{iter}} \cdot N_t} \Sigma_{j=1}^{N_{\text{iter}}} \Sigma_{k=1}^{N_t} \Pi_{\ell=1}^{5} \mathbf{I}_{j,k,\ell} \tag{17}$$

where $\mathbf{I}_{j,k,\ell}\{x\}$ is an indicator function of iteration j for target k and test ℓ that returns 1 when x is true and 0 otherwise, $\mathbf{I}_{j,k,1} = \{|t_0^k - \hat{t}_0^j| < \bar{\delta}_t\}$, $\mathbf{I}_{j,k,2} = \{|t_d^k - \hat{t}_d^j| < \bar{\delta}_t\}$, $\mathbf{I}_{j,k,3} = \{c_0^k \equiv \hat{c}_0^j\}$, $\mathbf{I}_{j,k,4} = \{c_d^k \equiv \hat{c}_d^j\}$, $\mathbf{I}_{j,k,5} = \{N_c^k \equiv \hat{N}_c^j\}$ are the five test criteria, t_0^k, t_d^k, c_0^k, c_d^k, and N_c^k are true hopping parameters of target k, and $\hat{t}_0^j$, $\hat{t}_d^j$, $\hat{c}_0^j$, $\hat{c}_d^j$, and $\hat{N}_c^j$ are best hopping parameter estimates of iteration j.

We start with finding out the number of input samples required for satisfying target detection performance in Section IV-A. We then evaluate the performance of single target detection ($N_t = 1$) under the impact of gross errors in Section IV-B and timing errors in Section IV-C. Finally, we evaluate the parameter estimation performance in multiple target scenarios in Section IV-D.

A. Number of Input Time-Frequency Samples

We first aim to find out the number of samples required from the detector in each sensing/acquiring period T_s to achieve a certain level of model parameter estimation accuracy and target detection probability P_d. Fig. 3 shows detection probability P_d for different numbers of input samples (N_s) for HopSAC and LS estimators with no gross errors ($\lambda = 0$) or timing errors ($\sigma_t^2 = 0$). We see that HopSAC requires slightly smaller number of samples to achieve $P_d = 0.9$ than Least Squares. To achieve close to 100% accuracy, both methods require approximately 35 samples. For fair comparison, we set acquiring period $T_s = 0.5\,\text{s}$ for the rest of the paper unless stated otherwise, which generates about 35 time-frequency samples for coarse hop period $\tilde{t}_d = 0.014\,\text{s}$.

B. Impact of Gross Errors

Next we evaluate the impact of gross errors on the accuracy of model parameter estimation. Fig. 4 shows the detection probability under the impact of gross errors. As shown in the figure, the detection probability for HopSAC and LS estimators with different degrees of outlier arrival rates λ in the input samples. For the same acquiring period ($T_s = 0.5\,\text{s}$), the performance of HopSAC is virtually unaffected by increasing outlier arrival rates whereas the performance of LS degrades significantly and drops to less than 50% even with a single outlier. This result is expected because HopSAC, like RANSAC, is capable of rejecting gross errors. LS, on the other hand, uses all samples including outliers for estimation. As a result, any outlier of large gross errors can considerably influence the accuracy of fitting. Since the target detection probabilities of HopSAC estimator are all above 0.996 for different levels of gross errors, this excellent ability to discard gross errors means that the impact of false positives is lower and a lower detection threshold can be used (e.g. a lower Constant False Alarm Rate (CFAR) threshold). Therefore, the detector is able to achieve better detection probability at a lower SNR.

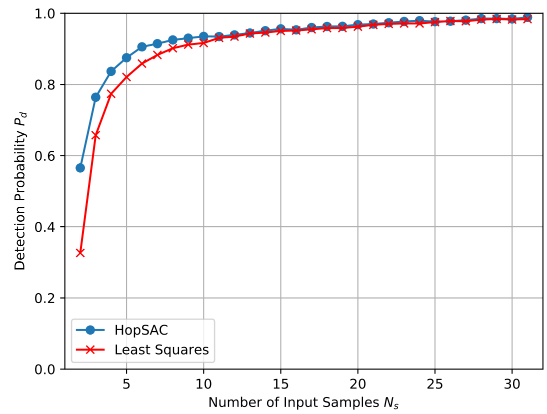

Fig. 3. Target Detection performance with Numbers of Input Samples.

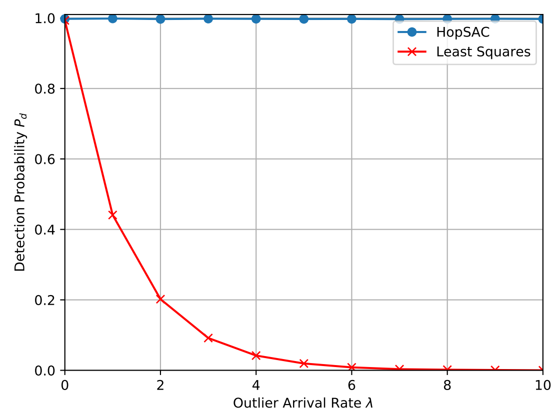

Fig. 4. Target Detection Performance under the Impact of Gross Errors.

C. Impact of Timing Errors

In this section, we consider the effect of timing errors on the performance of hopping parameter estimation. Fig. 5 shows the detection probability of HopSAC and LS estimators under the influence of timing errors in increasing variances of timing errors. As shown in the figure, HopSAC outperforms LS dramatically with the same number of samples. Unlike LS, HopSAC is very robust to small timing errors with variance

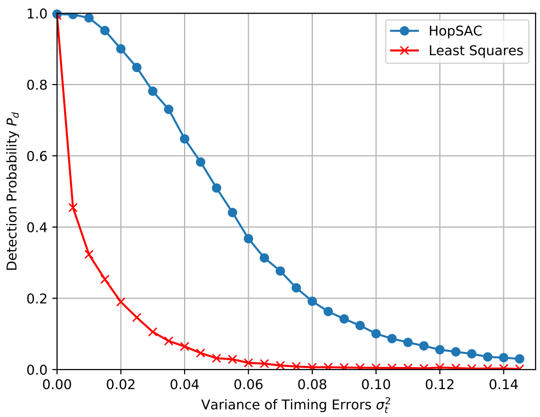

Fig. 5. Target Detection performance under the Impact of Timing Errors.

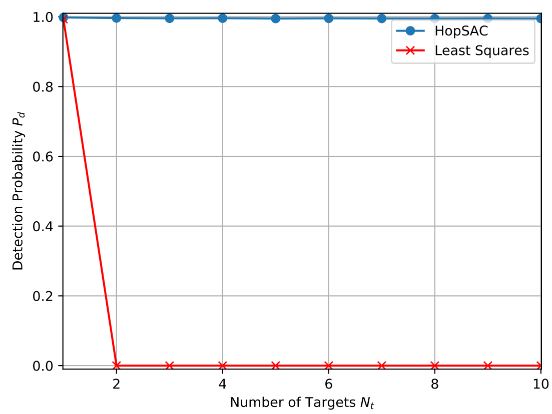

Fig. 6. Target Detection Performance for Multiple Target Detection.

$\sigma_t^2 < 0.01$. Since it is unlikely to observe huge timing errors with functional hardware (e.g. timing jitters are typically on the order of picoseconds), HopSAC is very robust to timing errors caused by timing jitters. The vastly improved robustness to timing errors also means that the pulses transmitted by the CUAS jammer can be much more precise in time. This improves the power efficiency of CUAS transmitters, which reduces collateral damage and unwanted interference to other legitimate UAS systems.

D. Multiple Target Scenarios

We now evaluate the performance of HopSAC and LS estimators for multiple target detection. Fig. 6 shows the target detection probability for different numbers of targets appearing in the input samples. The number of input samples N_s increases proportionally with the number of targets N_t in the same acquiring interval $T_s = 0.5\,\text{s}$. In the figure, we see that HopSAC achieves target detection probability greater than 0.994 for detecting up to 10 targets, confirming its consistency of good performance in multiple target detection, whereas LS is unable to detect anything in multiple target scenarios even with only two targets. The HopSAC's capability of detecting multiple targets lies in its ability to classify the outliers and inliers, associate each set of inlier samples to a specific target one by one, and extract outliers from all samples for the estimation of other targets. For this reason, the complexity of HopSAC grows linearly with the number of targets for detection. On the contrary, LS estimator uses all samples for estimation regardless of the number of targets. The samples from transmissions of one target are outliers of other targets. Thus, LS estimator is greatly affected by multiple target transmissions and is unable to accurately estimate the parameters in any multiple target scenarios.

V. Conclusions

In this paper, we propose a novel HopSAC frequency hopping parameter estimation method based on the RANSAC paradigm for CUAS systems. We show that HopSAC outperforms LS method, and achieves virtually 100% estimation accuracy under the impact of gross errors, small timing errors, and the presence of multiple targets. Its exceptional gross error rejection capability allows the detector to tolerate higher false positives with good detection performance in low SNR regions. Moreover, its robustness to timing errors significantly reduces response time and collateral damage while increasing the precision of transmissions and power efficiency. Finally, HopSAC excels in multiple target detection with exceptionally high detection probability at the expense of linearly increasing complexity.

References

[1] Federal Aviation Administration, "FAA aerospace forecasts fiscal years 2019-2039," 2019. [Online]. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/

[2] X. Liu, N. D. Sidiropoulos, and A. Swami, "Joint hop timing and frequency estimation for collision resolution in FH networks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 6, pp. 3063–3074, Nov. 2005.

[3] D. Angelosante, G. B. Giannakis, and N. D. Sidiropoulos, "Estimating multiple frequency-hopping signal parameters via sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5044–5056, Oct. 2010.

[4] Y. Wang, C. Zhang, and F. Jing, "Frequency-hopping signal parameters estimation based on orthogonal matching pursuit and sparse linear regression," *IEEE Access*, vol. 6, pp. 54 310–54 319, Sep. 2018.

[5] L. Zhao, L. Wang, G. Bi, L. Zhang, and H. Zhang, "Robust frequency-hopping spectrum estimation based on sparse bayesian method," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 781–793, Feb. 2015.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[7] "FCC PART 15.247 Test Report for GL300A (FCC ID: SS3-GL3001501)," Tech. Rep., Apr 2015. [Online]. Available: https://fccid.io/SS3-GL3001501/Test-Report/Test-Report-Rev-2599009

[8] "FCC PART 15.247 Test Report for GL300F (FCC ID: SS3-GL300F1609)," Tech. Rep., Sept 2016. [Online]. Available: https://fccid.io/SS3-GL300F1609/Test-Report/Test-Report-3155813

[9] "FCC PART 15.247 Test Report for GL300C (FCC ID: SS3-GL3001510)," Tech. Rep., Oct 2015. [Online]. Available: https://fccid.io/SS3-GL3001510/RF-Exposure-Info/SAR-Test-Report-2805129