Efficient Drone Hijacking Detection using Onboard Motion Sensors

Zhiwei Feng[1], Nan Guan[2], Mingsong Lv[1], Weichen Liu[3], Qingxu Deng[1], Xue Liu[4] and Wang Yi[5,1]

[1]Northeastern University, China [2]Hong Kong Polytechnic University, Hong Kong
[3]Chongqing University, China [4]McGill University, Canada [5]Uppsala University, Sweden

***Abstract*—The fast growth of civil drones raises significant security challenges. A legitimate drone may be hijacked by GPS spoofing for illegal activities, such as terrorist attacks. The target of this paper is to develop techniques to let drones detect whether they have been hijacked using onboard motion sensors (accelerometers and gyroscopes).**

Ideally, the linear acceleration and angular velocity measured by motion sensors can be used to estimate the position of a drone, which can be compared with the position reported by GPS to detect whether the drone has been hijacked. However, the position estimation by motion sensors is very inaccurate due to the significant error accumulation over time. In this paper, we propose a novel method to detect hijacking based on motion sensors measurements and GPS, which overcomes the accumulative error problem. The computational complexity of our method is very low, and thus is suitable to be implemented in the micro-controllers of drones. Experiments with a quad-rotor drone are conducted to show the effectiveness of the proposed method.

I. INTRODUCTION

Drones (or UAV, unmanned aerial vehicles) are aircrafts without human pilots on board. They have been increasingly used for various civil purposes, including surveillance, journalism, disaster management, environmental protection and various leisure activities. It is widely believed that they will be applied to more areas in the future. The drone industry has experienced an exponential growth in recent years. For example, the global sales of DJI, a young drone manufacturer, has increased for 80 times in the last three years [1].

The fast growth of drones raises significant security challenges. For example, a civil drone can be easily equipped with weapons or explosive materials to launch a terrorism attack. Currently many countries are working on laws to strictly control the whole life cycle of drones, including production, sales and use. However, there are still many challenges raised by drones that are difficult to solve even under very strict policies. One of the major problems is that a legal drone may be hijacked by others for illegal usage.

Drone Hijacking by GPS spoofing. Drones rely on GPS navigation in middle- or long-distance flights. The attacker can deceive the GPS receiver on a drone by broadcasting counterfeit GPS signals (called GPS spoofing [2]) and navigate the drone to fly following the attacker's intention. The most famous example of GPS attack to drones could be the RQ-170 incident. In December 2011, an Lockheed Martin RQ-170 Sentinel drone was captured by Iranian forces and guided to land in northeastern Iran by GPS spoofing[3]. Compared with military drones, civil drones are more vulnerable to GPS spoofing attacks since the civil GPS signals are not encrypted. In June 2012, researchers from University of Texas at Austin demonstrated to the U.S. Department of Homeland Security how can they hijack a civil drone with an inexpensive device (about $1000). Although researchers have proposed several methods to detect or prevent the GPS spoofing (see Section II), these methods all require extra hardware devices, which considerably increase both the cost and the weight of drones and thus may not be acceptable to the market of lightweight civil drones.

This work aims to provide a lightweight approach to let a drone detect whether itself has been hijacked. Our approach does not require any extra device, but only uses the onboard motions sensors that are mandatory on all drones for flight control. The linear acceleration and angular velocity measured by the motion sensors can be integrated over time to calculate the position of the drone. Therefore, ideally, we can detect drone hijacking by comparing the position computed according to the motion sensors and the position reported by GPS. Unfortunately, this approach does not work in practice due to the significant error accumulation over time. In Section III-B, we will discuss this inaccuracy problem in details.

Contribution of This Paper. This paper presents a novel hijacking detection method based on motion sensors and GPS, but can overcome the above mentioned error accumulation problem. The main idea is, instead of integrating the angular velocity and acceleration to estimate the position and compare it with that reported by GPS, we will combine the angular velocity by the gyroscopes and the position reported by GPS to estimate the linear acceleration, and compare it with the measurement of the accelerometers. In this way, we can avoid error accumulation due to the double-integrating of linear acceleration, and greatly improve the hijacking detection accuracy. Experiments show that our method is very effective to precisely detect hijacking. On the other hand, the complexity of our method is very low, and is suitable to be implemented on the micro-controllers of common civil drones.

*Corresponding author: Nan Guan, email: nan.guan@polyu.edu.hk

978-3-9815370-9-3/DATE17/©2017 EDAA

II. Related Works

Several GPS anti-spoofing techniques have been proposed in recent years. Reference [4] introduced a spoofing-aware receiver architecture that is able to detect spoofing attacks, classify the spoofing and authentic signals and mitigate the harmful effect of counterfeit spoofing signals. Reference [5] developed a single antenna anti-jam/anti-spoofing method called MAGIC. It applied a reduced-rank MMSE based C/A code correlator for single antenna GPS receivers that replaces a standard C/A code correlator for enhanced anti-jam/anti-spoofing capability. Reference [6] developed an INS batch RAIM monitor to detect GPS spoofing attacks. The common drawback of the above methods is that they require special hardware devices, which significantly increase the weight and cost of the drone. Therefore, these methods are not applicable to lightweight civil drones.

Reference [7] exploited the Doppler effect to monitor the behavior and integrity of the GPS signals to detect GPS spoofing. When a GPS receiver is able to precisely determine the Doppler shift of the carrier frequency, the difference from the actually measured shift is the cause of the receiver's own movement and then can be calculated. Reference [8] presented a method for spoofing mitigation in acquisition by joint detection of code Doppler and carrier Doppler. This method not only detects but also mitigates spoofing signal. Reference [9] used a special GPS receiver to detect sophisticated spoofing that cannot be detected using receiver autonomous integrity monitoring techniques. Reference [10] proposed a method based on sequential probability ratio test in acquisition. Reference [11] characterized spoofing detection methods and extracted the causal relation between measurement validity and signal integrity, then proposed an approach to derive signal integrity while capturing its uncertainty in a natural way. Reference [12] proposed the array processing method that used a predespreading approach to extract the spatial characteristics of spoofing signals without acquiring and tracking all the spoofing and authentic PRNs separately. All the above techniques involve very expensive signal processing algorithms, and thus are not suitable to lightweight civil drones, which use micro-controllers with limited computation capacity.

III. Preliminary

A. GPS Spoofing Attack

A GPS spoofing attack attempts to deceive a GPS receiver by broadcasting counterfeit GPS signals, structured to resemble a set of normal GPS signals, or by rebroadcasting genuine signals captured elsewhere or at a different time. These spoofed signals may be modified in such a way as to cause the receiver to estimate its position to be somewhere other than where it actually is, as determined by the attacker.

Fig. 1 illustrates GPS-spoofing-based hijacking of a drone. The drone originally plans to fly from the start point to the planned destination. The GPS spoofing starts when the drone flies to point p_0, after that the counterfeit GPS signal reports the fake positions along the red dash line and the drone will deviate from its planned route. For example, when the drone flies to the point p_2, and counterfeit GPS signal reports a fake position p_1, the drone will fly to the same direction as how it should fly from p_1 to the planned destination.

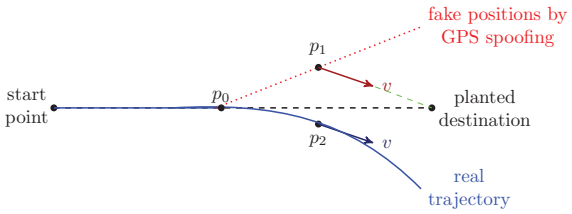


Fig. 1. An example of drone hijacking by GPS spoofing attacks.

B. INS

The motion sensors (accelerometers and gyroscopes) can be used as Inertial Navigation System (INS) [13] to compute the position and navigate the flight. Different from GPS, INS does not rely on any external signal. The left part (in the box) of Fig. 2 illustrates the basic principle of INS. The accelerometers measure the linear acceleration A in different directions in the body-fixed coordinates. The gyroscopes measure the angle velocities (in the yaw, pitch and row axis respectively) of the drone. Integrating the angle velocities over time gives the absolute angles of the drone, which are used to derive the transforming matrix M. Using M, the accelerations in the body-fixed coordinates are transformed into the linear accelerations A^* in the geographic coordinate. The linear accelerations A^* are integrated over time once to get the linear speeds V^*, and then integrated over time again to get the estimated position P^*. Note that since the drones fly in low speed, the influences of earth rotation is neglected.

C. Hijacking Detection by Comparing INS and GPS

A straightforward approach to detect whether a drone has been hijacked is to compare two positions estimated by GPS and INS at the same time, as shown in Fig. 2. However, this approach does not work in practise due to the poor accuracy of the position calculated by INS. The accelerometers and gyroscopes may introduce errors in their instantaneous measurement results. Although the instantaneous errors are typically very small, the accumulated error over time could be very large and thus the estimated position by INS may significantly deviate from reality. Fig. 3 shows the experimental result with a civil drone (Section V introduces the hardware platform of the drone), where the estimated trajectory grossly deviates from the real trajectory.

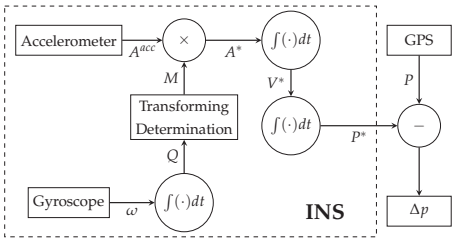


Fig. 2. Detection by comparing the positions reported by GPS and INS.

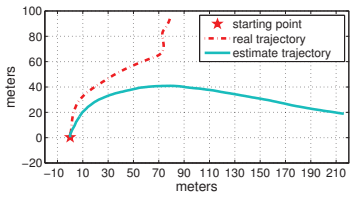


Fig. 3. The estimated positions using INS are far away from the GPS's

IV. Our New Method

The main idea of our method is shown in Fig. 4. We use the position information reported by the GPS to estimate the speeds V and then the accelerations A in the geographic coordinate. The angular velocities measured by the gyroscopes will be used to compute the transform matrix M, by which the accelerations A in the geographic coordinate are transformed to A^* accelerations in the body-fixed coordinate. Then we can detect drone hijacking if the difference between A^* and the measured accelerations A^{acc} by the accelerometers exceed certain threshold.

Compared with the method introduced in Section III-C, our method avoids the accumulated errors caused by the two integration operations with the accelerations measurements, which is the main reason why the INS-estimated position is very inaccurate. Therefore, our method can achieve a much higher detection precision. Notice that there is still an integration operation in Fig. 4, which calculates the angular information of the drone from the angular speeds measured by gyroscopes, in order to compute the coordinate transform matrix M. One may wonder if it is possible to develop a method to avoid this integration operation as well, e.g., by using the positions reported by GPS and the accelerations measured by the accelerometers to estimate the angular speed, and compare it with the measurement of gyroscopes, as shown in Fig. 5. In this way, the integration of ω is also avoided, and hopefully a higher detection precision can be achieved. However, this approach requires to solve a multivariate quadratic equation system, which is computationally expensive. We exclude this approach since this paper aims to develop an *efficient* detection method that is easy to be implemented on lightweight civil drones. On the other hand, as will be shown in Section V, the accumulative error due to the integration of the angular speed ω is not significant, and our detection method is sufficiently precise for practical usage.

In the following we introduce the details of our proposed hijacking detection method. Section IV-A focuses on how to compute the estimated linear accelerations A^* from the GPS and gyroscopes output data. Section IV-B presents algorithms to decide whether hijacking has happened based on the difference between A^* and A^{acc}, the linear accelerations measured by the accelerometers.

A. Calculating A^*

The GPS reports the position $P(t) = [P_x(t), P_y(t), P_h(t)]^T$ of the drone in the geographic coordinate at every sampling time point t. Since a sampling interval Δt is sufficiently

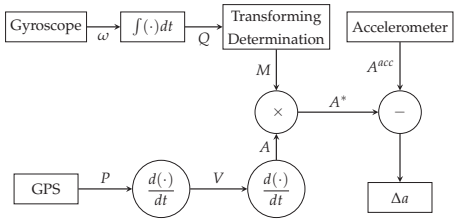


Fig. 4. The block diagram of our method

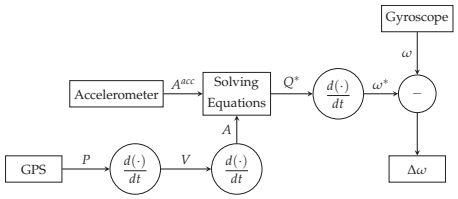


Fig. 5. The detection method by comparing the angular speeds, which involves heavy computation to solve multivariate quadratic equation systems.

small, we assume that a drone moves linearly with a constant acceleration between two successive sampling time points. Under this assumption, we can easily get the linear speed vector $V(t) = [V_x(t), V_y(t), V_h(t)]^T$ in the geographic at each time point t by:

$$V(t) = \frac{P(t) - P(t-1)}{\Delta t}$$

where $P(t-1)$ is the position vector reported by the GPS at the previous sampling time point.

The acceleration vector $A(t) = [A_x(t), A_y(t), A_h(t)]^T$ in the geographic coordinate at each time point t:

$$A(t) = \frac{V(t) - V(t-1)}{\Delta t} \tag{1}$$

where $V(t-1)$ is the linear speed vector obtained at the previous sampling time point.

Then $A(t)$ will be transformed into $A^*(t)$ in the body-fixed coordinate with the transformation matrix $M(t)$ at time t [13]:

$$A^*(t) = M^{-1}(t) \times A(t) \tag{2}$$

where $M^{-1}(t)$ is the inverse matrix of $M(t)$, and $M(t)$ is computed by:

$$M(t) = \begin{bmatrix} M_{11}(t) & M_{12}(t) & M_{13}(t) \\ M_{21}(t) & M_{22}(t) & M_{23}(t) \\ M_{31}(t) & M_{32}(t) & M_{33}(t) \end{bmatrix} \tag{3}$$

$$\begin{cases} M_{11}(t) = q_0(t)^2 + q_1(t)^2 - q_2(t)^2 - q_3(t)^2 \\ M_{12}(t) = 2(q_1(t)q_2(t) - q_0(t)q_3(t)) \\ M_{13}(t) = 2(q_1(t)q_3(t) + q_0(t)q_2(t)) \\ M_{21}(t) = 2(q_1(t)q_2(t) + q_0(t)q_3(t)) \\ M_{22}(t) = q_0(t)^2 - q_1(t)^2 + q_2(t)^2 - q_3(t)^2 \\ M_{23}(t) = 2(q_2(t)q_3(t) - q_0(t)q_1(t)) \\ M_{31}(t) = 2(q_1(t)q_3(t) - q_0(t)q_2(t)) \\ M_{32}(t) = 2(q_2(t)q_3(t) + q_0(t)q_1(t)) \\ M_{33}(t) = q_0(t)^2 - q_1(t)^2 - q_2(t)^2 + q_3(t)^2 \end{cases} \tag{4}$$

$Q(t) = [q_0(t), q_1(t), q_2(t), q_3(t)]^T$ is iteratively computed according to the angular speed vector

$$\omega = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$$

reported by the gyroscopes:

$$\begin{bmatrix}\Delta q_0(t)\\ \Delta q_1(t)\\ \Delta q_2(t)\\ \Delta q_3(t)\end{bmatrix}=\frac{1}{2}\begin{bmatrix}0 & -\omega_x(t) & -\omega_y(t) & -\omega_h(t)\\ \omega_x(t) & 0 & \omega_h(t) & -\omega_y(t)\\ \omega_y(t) & -\omega_h(t) & 0 & \omega_x(t)\\ \omega_h(t) & \omega_y(t) & -\omega_x(t) & 0\end{bmatrix}\begin{bmatrix}q_0(t-1)\\ q_1(t-1)\\ q_2(t-1)\\ q_3(t-1)\end{bmatrix}\Delta t \quad (5)$$

where $\Delta q_i(t)$ is the variation of $q_i(t)$ and we can get Q for the current moment t by

$$q_i(t) = q_i(t-1) + \Delta q_i(t).$$

Before being applied to (4) to compute M, Q can be normalized to be more robust to measurement errors:

$$q_i(t) \leftarrow \frac{q_i(t)}{\sqrt{q_0^2(t)+q_1^2(t)+q_2^2(t)+q_3^2(t)}}, \quad i=0,1,2,3 \quad (6)$$

such that it satisfies:

$$q_0(t)^2 + q_1(t)^2 + q_2(t)^2 + q_3(t)^2 = 1 \quad (7)$$

The iterative computation of $\overline{Q}$ starts with the initial values obtained as follows. The initial attitude angles of the drone are θ (pitch), φ (yaw) and γ (roll), then we compute the initial value of M, denoted by $\overline{M}$ by:

$$\begin{cases}\overline{M}_{11} = cos\gamma\cdot cos\varphi - sin\gamma\cdot sin\theta\cdot sin\varphi\\ \overline{M}_{12} = -cos\theta\cdot sin\varphi\\ \overline{M}_{13} = sin\gamma\cdot cos\varphi + cos\gamma\cdot sin\theta\cdot sin\varphi\\ \overline{M}_{21} = cos\gamma\cdot sin\varphi + sin\gamma\cdot sin\theta\cdot cos\varphi\\ \overline{M}_{22} = cos\theta\cdot cos\varphi\\ \overline{M}_{23} = sin\gamma\cdot cos\varphi - cos\gamma\cdot sin\theta\cdot sin\varphi\\ \overline{M}_{31} = -sin\gamma\cdot cos\theta\\ \overline{M}_{32} = sin\theta\\ \overline{M}_{33} = cos\gamma\cdot cos\theta\end{cases} \quad (8)$$

When a drone starts to fly, we assume that $\theta = \gamma = 0$ and φ can be calculated by the first two successive GPS points. Then according to (9) we get Q's initial absolute value.

$$\begin{cases}|\overline{q_1}| = \frac{1}{2}\sqrt{1+\overline{M}_{11}-\overline{M}_{22}-\overline{M}_{33}}\\ |\overline{q_2}| = \frac{1}{2}\sqrt{1-\overline{M}_{11}+\overline{M}_{22}-\overline{M}_{33}}\\ |\overline{q_3}| = \frac{1}{2}\sqrt{1-\overline{M}_{11}-\overline{M}_{22}+\overline{M}_{33}}\\ |\overline{q_0}| = \sqrt{1-\overline{q_1}^2-\overline{q_2}^2-\overline{q_3}^2}\end{cases} \quad (9)$$

The sign of $\overline{q_i}$ is decided as follows:

$$\begin{cases}\text{sign}(\overline{q_1}) = +\\ \text{sign}(\overline{q_1}) = \text{sign}(M_{32}-M_{23})\\ \text{sign}(\overline{q_2}) = \text{sign}(M_{13}-M_{31})\\ \text{sign}(\overline{q_3}) = \text{sign}(M_{21}-M_{12})\end{cases} \quad (10)$$

Note that the computation of the transformation matrix M is the same as in the standard position estimation in INS [13] (in both Fig. 2 and Fig. 4, we use ω to compute Q, and then compute M, and finally transfer accelerations from the body-fixed coordinates to the geographic coordinates).

The time complexity of the computation at each sampling time point is constant. As we mentioned before, one may think of avoiding the accumulative error of the gyroscope outputs using the approach in Fig. 5, which estimates the angular speeds using the GPS and accelerometers and compare them with the measurements of the gyroscopes. However, this requires us to compute q_0, q_1, q_2, q_3 by solving the equation system of (2). There is no general closed-form method to solve this multivariate quadratic equation systems, and one has to rely on numerical analysis methods (e.g., the Newton iteration method), which are computationally expensive.

Algorithm 1 Pseudo-code of calculating A^*

```
Compute M by (8)
Compute Q by (9) and (10)
for each sampling point i do
    Compute A(i) by (1)
    Compute Q(i) with Q(i − 1) by (5), (6)
    Compute M(i) with Q(i) by (4)
    Compute M^{-1}(i) from M(i)
    A*(i) ← M^{-1} × A(i)
end for
```

B. Hijacking Detection

Now we present how to decide whether the drone has been hijacked by comparing A^* and A. We compute the difference $\Delta a'(t) = [a'_x(t), a'_y(t), a'_h(t)]^T$ at each sampling time point t:

$$\Delta a'(t) = A(t) - A^*(t)$$

We first apply a simple median filtering algorithm to the $\Delta a(t)$ sequence by:

$$\Delta a(t) = \sum_{i=t-n}^{t} \Delta a'(i)/n$$

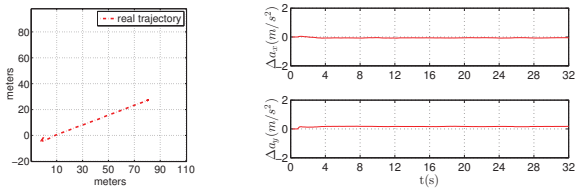
where n is the window size of the median filtering. This step is mainly to eliminate fluctuation noises, and the window size of n is relatively small. In the following we use $\Delta a(t)$ for hijacking detection.

Before presenting the hijacking detection rules, we first look into some data collected from experiments with a realistic drone (the parameters of the drone and the experiment methodology will be introduced in Section V), and use these examples to motivate the design of our hijacking detection rules. Since the height of the drone in GPS-spoofing-based hijacking is irrelevant, the following discussions we only consider the x and y axis.

In the experiment of Fig. 6, the drone flies normally along a straight line, where $\Delta a_x(t)$ and $\Delta a_y(t)$ are continuously close to 0. In Fig. 7, the drone is hijacked, where $\Delta a_x(t)$ and $\Delta a_y(t)$ significantly deviate from 0. These experiments imply that $|\Delta a_x(t)|$ and $|\Delta a_y(t)|$ are important for hijacking detection.

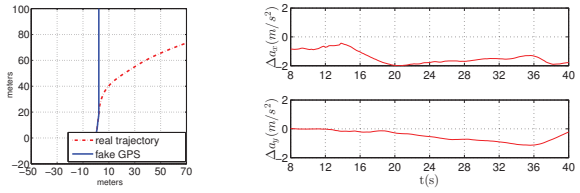
However, only relying on $|\Delta a_x(t)|$ and $|\Delta a_y(t)|$ may give wrong results. In the experiment of Fig. 8, the drone is not hijacked, but flies following a zigzag route. This time, Δa_y may deviate from 0 significantly during a certain period of time. However, its average over a period of time is sufficiently close to 0. This phenomenon suggests that when we see a large $|\Delta a_x(t)|$ or $|\Delta a_y(t)|$, it does not necessarily mean that a hijacking actually has happened. We should further evaluate its average over a period of time, and does not report hijacking if the average is close to 0.

On the other hand, $|\Delta a_y(t)|$ could be small in some hijacking cases. Fig. 7 shows such an example, where $|\Delta a_y(t)|$



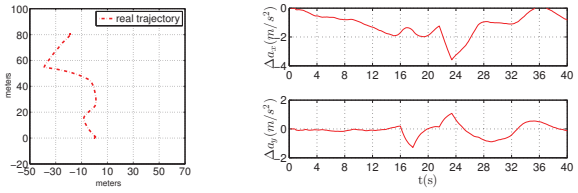
(a) Trajectory (b) Δa_x and Δa_y

Fig. 6. Detection rule motivating example 1.



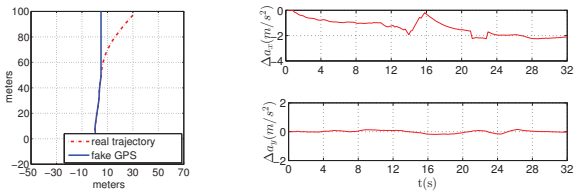
(a) Trajectory (b) Δa_x and Δa_y

Fig. 7. Detection rule motivating example 2.



(a) Trajectory (b) Δa_x and Δa_y

Fig. 8. Detection rule motivating example 3.



(a) Trajectory (b) Δa_x and Δa_y

Fig. 9. Detection rule motivating example 4.

is continuously close to 0, but $|\Delta a_x(t)|$ is large. We observed from our experiments that a typical behavior in hijacking is that the drone's acceleration along its heading direction varies dramatically, which is mainly due to the specific flight control algorithm of the drone used in our experiments. In this case, we will use the variance of $\Delta a_x(t)$ over a period of time. Note that the reason why we do not use the average, but the variance of $\Delta a_x(t)$ is that we observe that in many non-hijacked cases, $\Delta a_x(t)$ may deviate from 0 consistently, but vary much slower compared with the hijacked cases, as shown in Fig. 6.

By the above observations, we design the detection procedure as shown in Algorithm 2. Note that the window size N used to compute the average or the variance is subject to the designer's choice, but is typically much larger than n. In all the experiments in this paper we set $N = 100$. Moreover, the design of hijacking detection rules heavily depends on the flight control algorithm of the drone. The detection algorithm in this paper is designed with observations from experiments with a particular drone. In the future work we may conduct experiments to different drones and study detection algorithms that are more general to different flight control algorithms.

Algorithm 2 Pseudo-code of the Detecting procedure

```
1:  while not hijacked do
2:  calculate Δa(i) in Algorithm 1
3:      if |Δa_y(i)| > ε then
4:          calculate avg(Δa_y(i)) from i − N to i
5:          if avg(Δa_y(i)) > μ then
6:              claim hijacked; break
7:          end if
8:      else
9:          if |Δa_x(i)| > ρ then
10:             calculate var(Δa_x) from i − N to i
11:             if var(Δa_x) > δ then
12:                 claim hijacked; break
13:             end if
14:         end if
15:     end if
16: end while
```

V. Experiments

Experiments are conducted with a Quadrotor drone using the Pixhawk™flight control system [14]. It uses L3GD20H gyroscopes [15] and LSM303D accelerometers [16], for both of which we set the sampling rate of 50HZ (inter-sampling separation of 0.2 seconds). The drone uses the NEO-M8N GPS system [17], for which we set the data rate to 5Hz(inter-sampling separation of 0.2 second). Therefore, the GPS outputs are updated every 10 sampling points. The onboard micro-controller is PX4FMU [14], a 168MHz Cortex-M4F processor with 192KB SRAM and 1024 KB Flash.

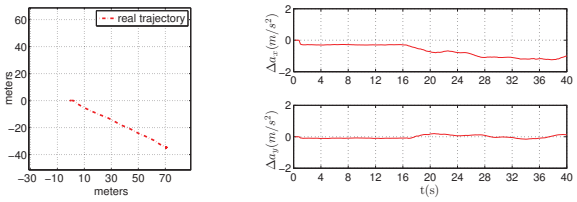
We let the drone fly in an open space for 100 times, among which the hijacked and non-hijacked cases are half-half. For the non-hijacked case, half of the flight routes are straight lines and half are randomly generated curves. The hijacked case is implemented as follows. We use an array to store the fake GPS signals on the micro-controller, and set a timer to trigger the hijacked mode, in which the GPS signal processing program reads inputs from this array instead of from the GPS receiver. We implement our hijacking detection method on the micro-controller and modify the flight control algorithm so that the drone will land as soon as it detects hijacking.

The precision is evaluated with the correctness ratio α:

$$\alpha = \frac{succ}{total},$$

where $total$ is the total number of experiments, and $succ$ is the number of experiments that our method correctly judges whether the hijacking has happened. In other words, in the hijacked case $1 - \alpha$ is the false-positive ratio while in the non-hijacked case $1 - \alpha$ is the false-negative ratio.

The thresholds ε, μ, ρ, and δ greatly affect the correctness ratio. If ε, μ, ρ, and δ are too tight, the correctness ratio in



(a) Trajectory (b) Δa_x and Δa_y

Fig. 10. Detection rule motivating example 5.

TABLE I
CORRECTNESS RATIO COMPARISON OF DIFFERENT METHODS

	no hijacked cases	hijacked cases
our method	96%	100%
GPS vs. INS method	0%	100%

the hijacked case will be higher but it will be lower in the non-hijacked case, and the other way around if ε, μ, ρ, and δ are too loose. Therefore, in the following we evaluate the correctness ratio in both hijacked and non-hijacked cases with varying ε, μ, ρ, and δ.

Fig. 11 to Fig. 14 show how the correctness ratio changes (for both the hijacked cases and non-hijacked cases) with one of the four thresholds, while keeping the other three constant. In all these experiments, the success ratio in hijacked case is very high, while the success ratio in the non-hijacked case improves as the thresholds increase.

According to the above experiment results, the following thresholds appear to be a good choice for our drone: $\varepsilon = 0.5$, $\mu = 0.1$, $\rho = 1.5$ and $\delta = 0.02$. We also implement the straightforward approach by comparing the position calculated by INS and that reported by GPS, and compare it with our method, as shown in Table I. It turns out that the straightforward approach always reports hijacking in both cases, due to the extremely poor position estimation accuracy by INS.

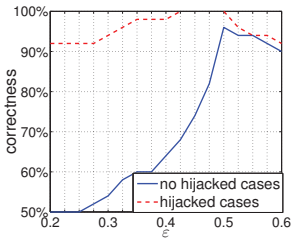


Fig. 11. The correctness vs. ε when $\mu = 0.1$, $\rho = 0.5$, $\delta = 0.02$

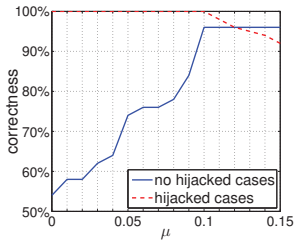


Fig. 12. The correctness vs. μ when $\varepsilon = 0.5$, $\rho = 0.5$, $\delta = 0.02$

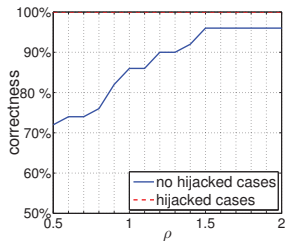


Fig. 13. The correctness vs. ρ when $\varepsilon = 0.5$, $\mu = 0.1$, $\delta = 0.02$

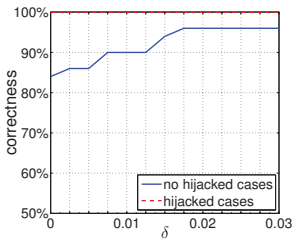


Fig. 14. The correctness cases vs. δ when $\varepsilon = 0.5$, $\mu = 0.1$, $\rho = 0.5$

VI. CONCLUSIONS

This paper presents an efficient method to let a drone detect whether it has been hijacked. This method does not require any extra hardware device, but only using data from the GPS and motion sensors (gyroscopes and accelerometers). The motion sensors can be used as INS (inertial navigation system) to estimate the position of the drone. However, the precision of INS is very low, due to the accumulative error problem. Therefore, it is not feasible to detect hijacking by comparing the positions reported by INS and GPS. The method proposed in this paper solves the accumulative error problem by a novel approach to use the motion sensors and GPS data. On the other hand, our method does not require any expensive computation, and thus is easy to be implemented in any drone.

ACKNOWLEDGMENT

This work is partially supported by National Natural and Science Foundation of China under grant no. 61672140, 61528202 and 61472072.

REFERENCES

[1] https://en.wikipedia.org/wiki/DJI_(company).
[2] D. M. Akos, "Who's afraid of the spoofer? gps/gnss spoofing detection via automatic gain control (agc)," *Journal of The Institute of Navigation*, vol. 59, no. 4, pp. 281–290, 2012.
[3] https://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident.
[4] A. Broumandan, A. Jafarnia-Jahromi, and G. Lachapelle, "Spoofing detection, classification and cancelation (sdcc) receiver architecture for a moving gnss receiver," *GPS Solutions*, 2015.
[5] W. L. Myrick, M. Picciolo, J. S. Goldstein, and V. Joyner, "Multistage anti-spoof gps interference correlator (magic)," in *IEEE Military Communications Conference*. IEEE, 2015, pp. 1497–1502.
[6] S. Khanafseh, N. Roshan, S. Langel, F. C. Chan, M. Joerger, and B. Pervan, "Gps spoofing detection using raim with ins coupling," in *2014 IEEE/ION Position, Location and Navigation Symposium*, May 2014, pp. 1232–1239.
[7] L. van Mastrigt, A. van der Wal, and P. Oonincx, "Exploiting the doppler effect in gps to monitor signal integrity and to detect spoofing," Piscataway, NJ, USA, 2015, pp. 8 pp. –. [Online]. Available: http://dx.doi.org/10.1109/IAIN.2015.7352259
[8] D. Yuan, H. Li, and M. Lu, "Gnss spoofing mitigation based on joint detection of code doppler and carrier doppler in acquisition," in *China Satellite Navigation Conference (CSNC) 2014 Proceedings: Volume I*. Springer, 2014, pp. 763–774.
[9] M. L. Psiaki, S. P. Powell, and B. W. O'Hanlon, "Gnss spoofing detection using high-frequency antenna motion and carrier-phase data," *Proceedings of International Technical Meeting of the Satellite Division of the Institute of Navigation*, pp. 2949–2991, 2013.
[10] D. Yuan, H. Li, and M. Lu, "A method for gnss spoofing detection based on sequential probability ratio test," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*. IEEE, 2014, pp. 351–358.
[11] X. Chen, G. Lenzini, M. Martins, S. Mauw, and J. Pang, "A trust framework for evaluating gnss signal integrity," in *2013 IEEE 26th Computer Security Foundations Symposium*, June 2013, pp. 179–192.
[12] S. D. et al., "A gnss structural interference mitigation technique using antenna array processing," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop*, June 2014, pp. 109–112.
[13] G. H. Elkaim, F. A. P. Lie, and D. Gebre-Egziabher, *Handbook of Unmanned Aerial Vehicles*. Springer Netherlands, 2015, ch. Principles of Guidance, Navigation, and Control of UAVs.
[14] "Pixhawk project," *https://pixhawk.org/*.
[15] "L3gd20h gyroscopes." *http://www.st.com/web/catalog/sense-power/FM89/SC1288/PF254039*.
[16] "Lsm303d accelerometers." *https://www.pololu.com/product/2127*.
[17] "Neo-m8n gps system." *https://www.u-blox.com/en/product/neo-m8qm8m-series*.