2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Stealthy Perception-based Attacks on Unmanned Aerial Vehicles*

Amir Khazraei, Haocheng Meng, and Miroslav Pajic

***Abstract*— In this work, we study vulnerability of unmanned aerial vehicles (UAVs) to stealthy attacks on perception-based control. To guide our analysis, we consider two specific missions: (i) ground vehicle tracking (GVT), and (ii) vertical take-off and landing (VTOL) of a quadcopter on a moving ground vehicle. Specifically, we introduce a method to consistently attack both the sensors measurements and camera images over time, in order to cause control performance degradation (e.g., by failing the mission) while remaining stealthy (i.e., undetected by the deployed anomaly detector). Unlike existing attacks that mainly rely on vulnerability of deep neural networks to small input perturbations (e.g., by adding small patches and/or noise to the images), we show that stealthy yet effective attacks can be designed by changing images of the ground vehicle's landing markers as well as suitably falsifying sensing data. We illustrate the effectiveness of our attacks in Gazebo 3D robotics simulator.**

I. Introduction

Recent years have witnessed significant research attention focused on unmanned aerial vehicles (UAVs) in a variety of military and civilian applications [1]. Also, recent progress in computer vision has resulted in a new generation of controllers that utilize visual data for decision making and control. Popular control tasks that incorporate visual data into the UAV's control-loop are vertical take-off and landing (VTOL) and ground vehicle tracking (GVT). These tasks have been widespread in the areas of rescue and reconnaissance, package delivery, inspection in hazardous environments, etc. Yet, despite their wide application, vulnerability of such systems to cyber attacks has not been well studied.

Quadcopter UAV's have been shown vulnerable to adversarial attacks, illustrating security challenges in these systems (e.g., [2]–[4]). For example, [2] has shown that an attacker could perform a Man-in-the-Middle attack, and inject false data and control commands to the compromised UAV, with catastrophic impact on the system and even human life. Thus, in this work, our goal is to evaluate the physical impact of these attacks (i.e., on the UAV's behavior), focusing on the false data-injection attacks that compromise both camera images and sensor measurements used for control.

Starting from [5], most existing adversarial attacks on images leverage vulnerability of deep neural networks to small changes in their input. The idea is to add imperceptible noise to the original image, causing the deep neural network's output to deviate from its original output [6]. This idea has been applied to UAVs in tracking [7] and control tasks [6], [8]. For example, [7] proposes adversarial attacks that add noise to the images to cause tracking drift in the bounding box around the target objects; in [8], noise is added to the image to change the steering angle and collision probabilities, degrading the UAV navigation performance.

Yet, despite their success in disrupting the controller performance, these attacks are not designed to be stealthy. Most existing UAVs are equipped with an anomaly detector (e.g., χ^2 detector based on the extended Kalman filter) that use the raw sensor measurements (and possibly the output of a vision module like position/pose estimation) to detect the presence of abnormal behaviors; thus, effectively limiting impact of the attacks. For non-perception control systems, stealthy attacks have been well-defined in e.g., [9]–[20], including replay [9], covert [13], zero-dynamic [11], and false data injection attacks [10], [12], [19]. However, in addition to not covering perception, these works only focus on LTI systems and controllers, limiting their use on UAVs.

On the other hand, control performance degradation caused by an attack on camera images used for control will be reflected on the employed physical sensor measurements (e.g., GPS and IMU). Hence, as we will show in the paper, if the sensor data is not suitably falsified, attacks on camera images will likely be detected. Consequently, it is unclear how vulnerable are the UAV controllers that employ both visual and 'traditional' sensing data, and whether it is possible to launch stealthy and effective attacks that significantly degrade control performance while staying undetected.

To answer this question, in this work, we introduce a general framework to design false data injection attacks that consistently falsify both camera images and physical sensors data, causing significant deviations from the trajectory of the non-compromised drone, while being stealthy from the system's intrusion detector. Specifically, we investigate attacks on two different mission tasks – VTOL and GVT, where a vision-based controller is used to navigate the drone. For both tasks, we consider a squared fiducial marker ArUco [21] on a moving vehicle, with a correlation filter being used to detect the marker and estimate the relative position and the heading of the drone with respect to the marker.

To show stealthiness of our attacks, in our experiments, we consider two widely-used anomaly detectors – χ^2 and deep learning-based detectors. We show that instead of adding noise to the image as commonly done today for non-control applications, the attacker should manipulate the scene geometry of the current image in a way that is governed by the UAVs dynamics. For example, for a moving vehicle tracking,

*This work is sponsored in part by the ONR under the agreementS N00014-20-1-2745 and N00014-23-1-2206, AFOSR award number FA9550-19-1-0169, NSF CNS-1652544 award as well as the National AI Institute for Edge Computing Leveraging Next Generation Wireless Networks, Grant CNS-2112562.

The authors are with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: amir.khazraei@duke.edu, haocheng.meng@duke.edu, miroslav.pajic@duke.edu).

our attack strategy starts with a desired false relative position. Then, the attacker constructs a false image that renders the falsified relative position of the marker. The desired false relative position is not chosen arbitrarily; instead, it is governed by the system dynamics to be stealthy from any intrusion detector. We will show that while having a physical impact on the drone (deviating the drone from its desired trajectory in GVT and landing in a wrong place in VTOL), the resulting attacks are stealthy from any considered anomaly detector.

This paper is organized as follows. Section II introduces the system model, before we present a formal attack model, capturing the required attack impact and stealthiness constraints, as well as a procedure to design such attacks (Section III). Section IV demonstrates effectiveness of the develop attacks, before concluding remarks in Section V.

II. System Model

We consider the common system architecture illustrated in Fig. 1 where raw sensor measurements from the IMU and GPS are used by a sensor fusion module to estimate the systems state, whereas the relative position of the ground vehicle with respect to the camera is obtained by a vision module. Depending on current state of the system, the Finite State Machine (FSM) switches between the drone control tasks, for VTOL and GVT as in e.g., [22], [23].

We also assume that the system is equipped with an anomaly detector that utilizes the sensing information, as well as the controller and FSM outputs, to detect presence of abnormal behaviours. Further, we assume that an attacker can exploit the vulnerabilities of the employed communication network to compromise the sensing information delivered to the controller. In this work, we show that under such assumption, the attacker can design a sequence of false image and sensor values that can degrade the control performance while remaining stealthy from the employed anomaly detector, independent of the type of the utilized detector.

In what follows, we first describe the UAV's dynamical model that the attacker uses to generate the false images and sensor values at runtime. Then, we present the deployed vision-based controller and anomaly detector. Finally, the notion of stealthy and effective attacks is introduced as well as a methodology to design such attacks.

A. Vehicle Dynamical Model

The quadcopter's dynamical model can be described using the standard Newton-Euler equations as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v}, \qquad m\dot{\mathbf{v}} = \mathbf{R}\mathbf{f}^{\mathcal{B}} - mge_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\omega}}^{\mathcal{B}}, \quad \boldsymbol{\tau}^{\mathcal{B}} = \boldsymbol{\mathcal{I}}\dot{\boldsymbol{\omega}}^{\mathcal{B}} + \boldsymbol{\omega}^{\mathcal{B}} \times \boldsymbol{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{B}}; \end{aligned} \tag{1}$$

here, $\mathbf{p} = \left[x, y, z\right]^T$ and $\mathbf{v}$ denote the position and velocity of the vehicle in the earth frame $\mathcal{E}$, m is the vehicle's mass, $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix from the body frame to the earth frame, g is the gravity acceleration, $\mathbf{f}^{\mathcal{B}} = \left[0, 0, F_z\right]^T$ is the force vector in the body frame, $e_3 = \left[0, 0, 1\right]^T$ is the unit vector, $\boldsymbol{\omega}^{\mathcal{B}} = \left[p, q, r\right]^T$ is the angular velocity in body frame, $\hat{\cdot}$ denotes the operator that maps a vector in

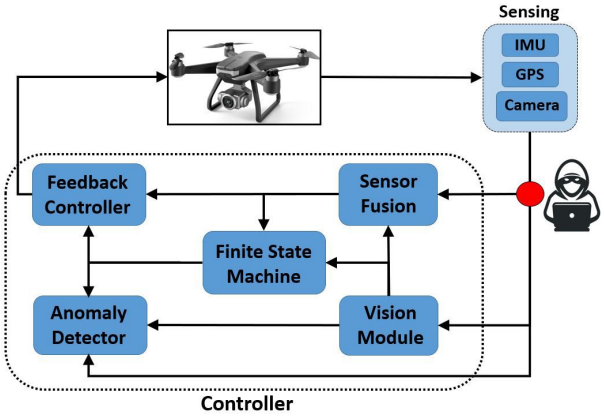

Fig. 1: The architecture of the perception-based UAV control system under attack on system sensing, including perception. Independently of the way attacks are actually implemented (e.g., directly compromising a sensor or modifying the measurements delivered over the network to the controller), the same impact on the control performance is obtained.

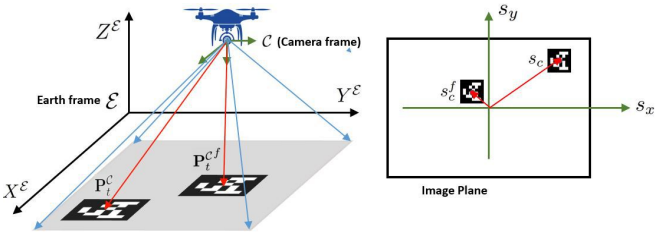

Fig. 2: Illustration of the drone with the landing mark, camera frame $\mathcal{C}$, earth frame $\mathcal{E}$, and image plane.

$\mathbb{R}^3$ to a skew-symmetric matrix, $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{3\times3}$ is the inertia matrix, and $\boldsymbol{\tau}^{\mathcal{B}} = \left[\tau_x, \tau_y, \tau_z\right]^T$ is the total torque vector in the body frame. The inputs to the system are the squared angular velocity of the four rotors $\mathbf{u} = \left[w_1^2, w_2^2, w_3^2, w_4^2\right]^T$. The torques and thrust are obtained using the angular velocity of the four rotors as

$$\begin{bmatrix} F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix}, \tag{2}$$

where l is the distance from the motor to the center of gravity, and b and d are the thrust and drag coefficients, respectively.

By defining $\mathbf{x} = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, p, q, r\right]^T$, after discretization the system (1) can be captured in the state-space form

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t, \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{v}_t, \quad \mathbf{z}_t = G(\mathbb{X}_t^{\mathcal{C}}), \end{aligned} \tag{3}$$

where $\mathbf{w}_t$ is the system disturbance, $\mathbf{y}_t$ is the vector of the raw sensor measurements with Gaussian noise $\mathbf{v}_t$, G is the perspective camera projection model that maps the points represented in the camera frame $\mathcal{C}$ at time t, denoted by $\mathbb{X}_t^{\mathcal{C}}$, to the pixel images $\mathbf{z}_t$. Since the 3D points captured by camera $\mathbb{X}_t^{\mathcal{C}}$ are a function of the position of the camera and the UAV in the earth frame $\mathcal{E}$, $\mathbf{z}_t$ is explicitly a function of state $\mathbf{x}_t$; the corresponding frames are illustrated in Fig. 2.

B. Vision-Based Controller

The objective of the vision-based controller is to use the camera-feed (i.e., images) and sensor measurements to compute the desired control inputs. Here, we utilize the widely-employed vision-based control framework as in e.g., [22], [24], [25]. We consider a squared fiducial marker ArUco [21] as a landing mark. Specifically, a pin-hole model is used to find the relative position of the landmark center with respect to the camera frame (Fig. 2, on the right) – i.e.,

$$s_c = \begin{bmatrix} s_{c_x} \\ s_{c_y} \end{bmatrix} = \frac{f_c}{Z^{\mathcal{C}}} \begin{bmatrix} X^{\mathcal{C}} \\ Y^{\mathcal{C}} \end{bmatrix}, \tag{4}$$

where $\left[X^{\mathcal{C}}, Y^{\mathcal{C}}, Z^{\mathcal{C}}\right]^T \in \mathbb{X}_t^{\mathcal{C}}$ is a point on a 3D object visible to the pinhole camera in the camera frame $\mathcal{C}$ and $s_c = \begin{bmatrix} s_{c_x} & s_{c_y} \end{bmatrix}^T$ is the projected point on the image plane with the focal distance f_c (see Fig. 2).

After detecting the landing mark position s_c in the current image (Fig. 2), using (4) and the fact that the target (i.e., marker) size is known (which enables computing the $Z^{\mathcal{C}}$ in (4)), the relative position of the landing mark *with respect to the camera reference frame*, denoted by $\mathbf{P}_t^{\mathcal{C}}$, can be estimated from the image $\mathbf{z}_t$; i.e., we can define

$$V(\mathbf{z}_t) = \hat{\mathbf{P}}_t^{\mathcal{C}} = \mathbf{P}_t^{\mathcal{C}} + \mathbf{e}(\mathbf{P}_t^{\mathcal{C}}), \tag{5}$$

where V is the described mapping from an image $\mathbf{z}_t$ into $\hat{\mathbf{P}}_t^{\mathcal{C}}$, the estimated value of the relative position of the landing mark in the camera frame (i.e., $\mathbf{P}_t^{\mathcal{C}}$). In addition, $\mathbf{e}(\mathbf{P}_t^{\mathcal{C}})$ is the error of relative position estimation caused by the vision module and it is a function of the actual value of relative position $\mathbf{P}_t^{\mathcal{C}}$. Here, we use a common assumption for perception-based control (e.g., [26]) that the norm of the camera-based localization error is bounded by some $\gamma > 0$; i.e., $\|\mathbf{e}(\mathbf{P}_t^{\mathcal{C}})\| \leq \gamma$.

To connect the target position $\mathbf{P}_t^{\mathcal{C}}$ in the camera frame to the UAV state, in particular the drone position $\mathbf{p}_t$ captured in the earth (i.e., global) frame (see (1)), $\mathbf{P}_t^{\mathcal{C}}$ can be represented using the coordinates in the global frame as

$$\mathbf{P}_t^{\mathcal{C}} = \mathbf{R}_{\mathcal{E}}^{\mathcal{C}}(\mathbf{x}_t)\Big(\mathbf{P}_t^{\mathcal{E}} - \mathbf{p}_t\Big); \tag{6}$$

here, $\mathbf{R}_{\mathcal{E}}^{\mathcal{C}}(\mathbf{x}_t)$ is the rotation matrix from the earth frame $\mathcal{E}$ to the camera frame $\mathcal{C}$ and $\mathbf{P}_t^{\mathcal{E}}$ is the landing mark position in the earth frame $\mathcal{E}$.

The UAVs control goals are determined by the FSM unit. Specifically, there are two states for the GVT; the first is to take off to the highest altitude such that the target vehicle is visible and can be detected. Then, the second is to follow the ground target vehicle at a certain altitude. For the VTOL task, there are three states. The first is similar to the first state of GVT. However, in the second state, the goal is to minimize the relative distance to the landing mark. Finally, if the relative distance from the target is less than a predefined threshold, the drone switches to the *landing-mode*, which is the final state for VTOL. For each VTOL and GVT task, a cascade PID controller uses $\hat{\mathbf{P}}_t^{\mathcal{C}}$ and the sensor fusion output to control the position and the attitude of the drone. Thus, we assume that the closed-loop control system is stable.

C. Anomaly Detector

Physics-based anomaly detectors including (but not limited to) χ^2, cumulative sum (CUSUM), sequential probability ratio test (SPRT) have been widely deployed in robotics systems like UAVs (e.g., [19], [27]). Recently, learning-based anomaly detectors have been employed for detecting the presence of attacks or any system anomalies (e.g., [28]). Without loss of generality, assume that the attack starts at time $t = 0$, and let us use $\mathbf{O}_t = \{\mathbf{z}_t, \mathbf{y}_t\}$ and $\mathbf{O}_t^a = \{\mathbf{z}_t^a, \mathbf{y}_t^a\}$ to respectively denote the real and compromised (i.e., false) sensing measurements at time t, which includes the sensor values and the camera image. Now, given a random sequence $\bar{\mathbf{O}}^t = \{\bar{\mathbf{O}}_{-T_0} : \bar{\mathbf{O}}_t\}$ of the sensing data received by the controller, any employed detector (no matter the specific detector design) is effectively trying to solve the hypothesis problem:

H_0: normal behavior—i.e., $\mathbf{O}_{-T_0} : \mathbf{O}_t$ was received;

H_1: abnormal behavior—i.e., $\mathbf{O}_{-T_0} : \mathbf{O}_{-1}, \mathbf{O}_0^a : \mathbf{O}_t^a$ was received.

Thus, the sequence $\bar{\mathbf{O}}^t$ either comes from the distribution of the null hypothesis H_0, which is determined by the system uncertainties, or from a distribution of the alternative hypothesis H_1, *which is unknown but controlled by the attack signals*. For a given anomaly detector specified by a function $D : \bar{\mathbf{O}}^t \to \{0, 1\}$, false alarm occurs if $D(\bar{\mathbf{O}}^t) = 1$ when $\bar{\mathbf{O}}^t$ comes from H_0, and we denote the probability of false alarm as $p^{FA}(D)$; whereas true detection occurs if $D(\bar{\mathbf{O}}^t) = 1$ when $\bar{\mathbf{O}}^t$ comes from H_1, and we denote the probability of true detection as $p^{TD}(D)$.

III. Design of Stealthy Effective Attacks

In this section, we introduce an algorithm to attack consistently both the images and the sensors, ensuring both attack effectiveness and stealthiness. We assume that the attacker has access to the current sensor measurements and camera images (i.e., $\mathbf{z}_t, \mathbf{y}_t$). Now, to formally capture the attacker's goal, we start by formalizing the stealthiness notion and the attack impact on the control performance.

Definition 1: An attack is *strictly stealthy* if for a resulting observation sequence $\bar{\mathbf{O}}^t$, there exists no detector for which $p_t^{FA} < p_t^{TD}$ holds, for any $t \geq 0$. An attack is ϵ-*stealthy* if for a resulting observation sequence $\bar{\mathbf{O}}^t$ and a given $\epsilon > 0$, there exists no detector such that $p_t^{FA} < p_t^{TD} - \epsilon$ holds, for any $t \geq 0$.

From the definition, the attack sequence is strictly stealthy if there is no detector for which the probability of true detection is greater than the probability of false alarm. However, reaching that level of stealthiness may not be possible and therefore, we also define ϵ-stealthiness for which the difference between the probabilities of true detection and false alarm is less than ϵ for any employed detector.

Since the goal of the vision-based control is to follow the target vehicle, the attack degrades control performance by causing a deviation from the desired trajectory of the drone in the 3-D space. In particular, the attack goal is to force the drone away from the target vehicle by some α; this is equivalent to moving the target landing marker in *the camera*

frame (i.e., $\mathbf{P}_t^{\mathcal{C}}$) α-meters away (in 3-D space). Thus, we introduce the following definition.

Definition 2: An attack is α-*effective* if $\|\mathbf{P}_t^{\mathcal{C}}\|_2 \geq \alpha$ for some $t \geq 0$.

Now, the attack goal can be formalized as designing a sequence of false sensing information from time $t = 0$, i.e., $\{\mathbf{y}_0^f, \mathbf{z}_0^f\}, ..., \{\mathbf{y}_T^f, \mathbf{z}_T^f\}$, that is α-effective and ϵ-stealthy for all $t \in [0, T]$. A sequence of false sensing values that satisfy the attack goal is referred to as an (ϵ, α)-*attack sequence*.

A. Attack Design

The previous works [10], [12], [16], [29]–[31] have shown that stealthiness conditions require that the attack impact (i.e., control degradation) is compatible with the system dynamics. For example, if the attacker wants to cause deviation in the drone trajectory (compared to the unattacked system's trajectory), the deviation needs to increase smoothly and in accordance with the dynamics of the system.

Specifically, if $\mathbf{s}_t$ is used to denote the state deviation due to the attack, to satisfy the stealthiness condition, [16], [29] show that $\mathbf{s}_t$ needs to dynamically evolve over time as follows

$$\mathbf{s}_{t+1} = f(\hat{\mathbf{x}}_t^a, \mathbf{u}_t) - f(\hat{\mathbf{x}}_t^a - \mathbf{s}_t, \mathbf{u}_t), \tag{7}$$

where $\mathbf{s} \in \mathbb{R}^{12}$ and f is defined in (3), with some small nonzero initial condition $\mathbf{s}_0$; also, $\hat{\mathbf{x}}_t^a$ denotes the output of a sensor fusion that the attacker runs online using the sensing information at each time step (e.g., using $\mathbf{y}_t$, and potentially $\mathbf{z}_t$). Note that $\hat{\mathbf{x}}_t^a$ differs from the system's sensor fusion output $\hat{\mathbf{x}}_t$, as the former is obtained from the actual sensing measurements whereas the latter is computed using the compromised sensing information (i.e., $\mathbf{y}_t^f$).

We next show how the attacker can exploit the sequence $\mathbf{s}_t$, $t \geq 0$, to design a stealthy and effective sequence of false physical sensor and image data.

Attacks on Sensor Measurements: From (3), the physical sensors map the states $\mathbf{x}_t$ to the observation values with function h. Thus, to remain stealthy, the false sensor values should be obtained by mapping $\mathbf{x}_t - \mathbf{s}_t$ using function h – i.e., ideally, the falsified sensor values should be $\mathbf{y}_t^f = h(\mathbf{x}_t - \mathbf{s}_t) + \mathbf{v}_t$. However, there are two challenges to construct such falsified sensor data: no access to the actual state of the system $\mathbf{x}_t$ and measurement noise $\mathbf{v}_t$.

For the former, the estimated state value $\hat{\mathbf{x}}_t^a$ can be used instead of $\mathbf{x}_t$. For the latter, the attacker can subtract $h(\hat{\mathbf{x}}_t^a)$ from the current sensor measurements $\mathbf{y}_t$ to estimate the noise level. Combining these, the sequence of false physical sensor values for $t \geq 0$ can be computed as

$$\mathbf{y}_t^f = \mathbf{y}_t + \mathbf{a}_t = \mathbf{y}_t + h(\hat{\mathbf{x}}_t^a - \mathbf{s}_t) - h(\hat{\mathbf{x}}_t^a); \tag{8}$$

i.e., the attacker just needs to add $\mathbf{a}_t = h(\hat{\mathbf{x}}_t^a - \mathbf{s}_t) - h(\hat{\mathbf{x}}_t^a)$ to the current physical sensor measurement $\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t$. As the state estimation error $\mathbf{x}_t - \hat{\mathbf{x}}_t^a$ decreases, $\mathbf{y}_t^f$ from (8) approaches the ideal (from the perspective of the attack effectiveness and stealthiness) false sensor value – i.e., $\mathbf{y}_t^f = h(\mathbf{x}_t - \mathbf{s}_t) + \mathbf{v}_t$.

Algorithm 1 Design of stealthy and effective attacks by compromising both physical sensors and image starting at time 0 until T

Initialize s_0
for $t = 0 : T$ **do**
  Find $\hat{\mathbf{x}}_t^a$ and $\hat{\mathbf{P}}_t^{\mathcal{E}}$
  $\mathbf{y}_t^f = \mathbf{y}_t + h(\hat{\mathbf{x}}_t^a - \mathbf{s}_t) - h(\hat{\mathbf{x}}_t^a)$
  $\mathbf{P}_t^{\mathcal{C}f} = \mathbf{R}_{\mathcal{E}}^{\mathcal{C}}(\hat{\mathbf{x}}_t^a - \mathbf{s}_t)\Big(\hat{\mathbf{P}}_t^{\mathcal{E}} - (\hat{\mathbf{p}}_t^a - \mathbf{s}_t^{\mathbf{p}})\Big)$
  $s_c^f = \begin{bmatrix} s_{c_x}^f \\ s_{c_y}^f \end{bmatrix} = \frac{f_c}{Z_{\mathbf{P}_t^{\mathcal{C}f}}} \begin{bmatrix} X_{\mathbf{P}_t^{\mathcal{C}f}} \\ Y_{\mathbf{P}_t^{\mathcal{C}f}} \end{bmatrix}$
  $l^f = \frac{f_c}{Z_{\mathbf{P}_t^{\mathcal{C}f}}} l$
  Generate $\mathbf{z}_t^f$ with s_c^f and l^f
  $\mathbf{s}_{t+1} = f(\hat{\mathbf{x}}_t^a, \mathbf{u}_t) - f(\hat{\mathbf{x}}_t^a - \mathbf{s}_t, \mathbf{u}_t)$
end for

Attacks on Camera Images: Similarly, to remain stealthy, the attacker also needs to remove from the actual image, the impact of the deviation caused by the attack. To achieve this, our approach is to first identify the desired false position and size of the landing mark in the image plane. This is done by finding the desired false position of the landing mark in the camera frame and then using (4) to get the desired false position and size of the landing mark in the image plane (see for example Fig. 2).

When the drone is not under attack, the current position of the landing mark in the camera frame (i.e., $\mathbf{P}_t^{\mathcal{C}}$) satisfies (6). To satisfy the stealthiness constraint by removing the deviation, the idea is to replace all the system states in (6) with $\hat{\mathbf{x}}_t^a - \mathbf{s}_t$; i.e., the false position of the landing mark in the camera frame, denoted by $\mathbf{P}_t^{\mathcal{C}}$, should be computed as

$$\mathbf{P}_t^{\mathcal{C}f} = \mathbf{R}_{\mathcal{E}}^{\mathcal{C}}(\hat{\mathbf{x}}_t^a - \mathbf{s}_t)\Big(\hat{\mathbf{P}}_t^{\mathcal{E}} - (\hat{\mathbf{p}}_t^a - \mathbf{s}_t^{\mathbf{p}})\Big); \tag{9}$$

here, $\mathbf{R}_{\mathcal{E}}^{\mathcal{C}}(\hat{\mathbf{x}}_t^a - \mathbf{s}_t)$ is the rotation matrix from the earth frame to the camera frame evaluated by fake Euler angles associated with $\hat{\mathbf{x}}_t^a - \mathbf{s}_t$; $\hat{\mathbf{P}}_t^{\mathcal{E}}$ is the attacker's estimate of the position of the landing mark in the earth frame; $\mathbf{s}_t^{\mathbf{p}}$ captures the first three elements of the vector $\mathbf{s}_t$ (associated with the vector $\mathbf{p} = \begin{bmatrix} x, y, z \end{bmatrix}^T$). Note that to estimate $\hat{\mathbf{P}}_t^{\mathcal{E}}$, the attacker can either have its own resources (e.g., placing sensing such as GPS on the target and estimating the position using a Kalman filter) or can use the actual image $\mathbf{z}_t$ to find the relative position and then combine it with the drone position in the earth frame $\mathcal{E}$ (the first three elements of $\hat{\mathbf{x}}_t^a$).

Now, the attacker should design an image at each time $t \geq 0$ for which the vision module output (5) results in $\mathbf{P}_t^{\mathcal{C}f}$ from (9). Once $\mathbf{P}_t^{\mathcal{C}f}$ is obtained, the desired false location for the landing mark center in the image plane can be found using (4) – i.e.,

$$s_c^f = \begin{bmatrix} s_{c_x}^f \\ s_{c_y}^f \end{bmatrix} = \frac{f_c}{Z_{\mathbf{P}_t^{\mathcal{C}f}}} \begin{bmatrix} X_{\mathbf{P}_t^{\mathcal{C}f}} \\ Y_{\mathbf{P}_t^{\mathcal{C}f}} \end{bmatrix}. \tag{10}$$

On the other hand, the scale of the landing mark in the image plane needs to be compatible with the desired output of the vision module $\mathbf{P}_t^{\mathcal{C}f}$. If the actual physical length of the

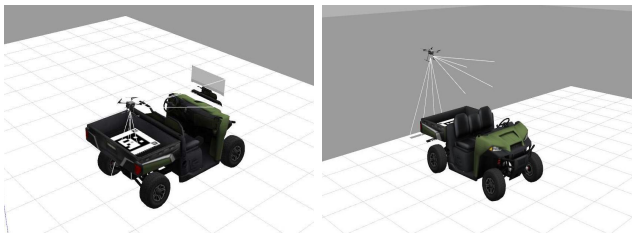

Fig. 3: The simulation environment.

squared ArUco marker is l, the length of the landing mark in the fake image plane l^f can be obtained as

$$l^f = \frac{f_c}{Z_{\mathbf{P}_t^{cf}}} l. \tag{11}$$

The resulting procedure is summarized in Algorithm 1.

Finally, to find a falsified image that satisfies (10) and (11), different methods can be employed. A promising approach is to use deep learning based generative models such as conditional generative adversarial network (cGAN) or similar generative approaches [32]. In this work, we use common computer vision methods, as described in the next section.

IV. Results and Evaluation

We now describe the performed evaluation of the effectiveness and stealthiness of the introduced attack methodology on the GVT and VTOL control tasks. Representative experimental videos are provided at [33].

A. Implementation Details

We evaluated the presented attacks using an open-source autonomous vehicle simulation platform Prometheus [34] powered by the PX4 flight controller and Gazebo 3D robotics simulator. The built-in quadcopter uses a sensor fusion model of IMU and GPS and is equipped with a down-facing camera with a resolution of 1280×720, allowing us to implement the control architecture from Fig. 1. In our simulations, the quadcopter identifies and tracks an ArUco marker placed on a ground vehicle to perform autonomous tracking and landing missions (as illustrated in Fig. 3).

Attacks on sensors were implemented using Algorithm 1. To generate false images, we assumed that the attacker has a base image with similar background, moving the position of the ground vehicle with the landing marker and resizing them by up-sampling (or down-sampling) using the formulas in Algorithm 1. Two different anomaly detectors were implemented to evaluate attack stealthiness: χ^2 and learning-based detectors. χ^2 detector uses a weighted norm of the residue generated by the Extended Kalman filter, raising alarm if the residue is larger than a predefined threshold. We also implemented a widely-used learning-based detector [28] that employs a long short-term memory (LSTM) architecture.

B. Attacks on Ground Vehicle Tracking

In this task, the drone's goal is to track the ground vehicle that follows a square trajectory clockwise illustrated in Fig. 4a with the red dashed line. The blue line shows the trajectory of the drone without attack, successfully tracking the vehicle. The orange trajectory shows the drone trajectory when the attack starts at the pink square. In this experiment, we assumed the elements of the initial $\mathbf{s}_0$ were all zero except the one associated with the roll angle which was set to 0.01. With such initial condition, the attacker was successful in deviating the drone trajectory along the Y axis up to $3\ m$ at which point the ground vehicle went out of the range of the camera's scope and we stopped the attack; note e.g., that red stars highlight the drone positions at the same times $t = 30\ s$ for trajectories under attack and without attack, illustrating significant deviations due to the attack. Similar results were shown for other $\mathbf{s}_0$ initializations, where the error was along the X axis, or the combination of both axes.

TABLE I: Average true detection rate for χ^2 and learning-based anomaly detectors for different values of the initial condition norm ($\|\mathbf{s}_0\|$) in the GVT task when $p^{FA} = .01$.

	p^{TD} (χ^2)	p^{TD} (learning-based)
$\|\mathbf{s}_0\| = .001$	.009	.011
$\|\mathbf{s}_0\| = .01$	.009	.014
$\|\mathbf{s}_0\| = .1$	.26	.55

TABLE II: Detection rate of the χ^2 and learning-based detectors for different values of α (deviation) in m when only images were under attack in the VTOL task, with $p^{FA} = .01$.

	$\alpha = .2$	$\alpha = .4$	$\alpha = .6$	$\alpha = .8$	$\alpha = 1$
p^{TD} (χ^2)	.01	.79	1	1	1
p^{TD} (learning-based)	.05	.42	1	1	1

Fig. 4b and Fig. 4c show the average alarm rate at each time step for 10 experiments when both χ^2 and learning-based anomaly detectors were used, respectively. As the attack started at time $t = 0$, the value of true alarm averages (for $t > 0$) $p^{TD} = .01$ was almost the same as the false alarm averages (for $t \geq 0$) for both detector, showing the ϵ-stealthiness of the attack according to Definition 1.

Table I illustrates the impact of the norm of the initial $\mathbf{s}_0$ on the stealthiness level. The true detection rate increases with the increase in the norm of the initial error state $\mathbf{s}_0$. Intuitively, this is caused by a huge initial error bias due to the attack at time $t = 0$ (attack start), which can be easily detected by an anomaly detector. On the other hand, for $\mathbf{s}_0$ arbitrarily close to zero the attack will not be detected, but it will take longer time for the attack to be effective. Therefore, there is a trade-off between the attack detectability and the time required to achieve significant performance degradation.

So far, we assumed that the attack is implemented on both sensors and images consistently according to Algorithm 1. However, Table II shows the probability of attack detection for different values of deviation when the sensors were not under attack and only images were compromised – even for a small deviation of $0.4\ m$ the attack was detected by χ^2 detector with true detection rate of 0.79 which is 79 times higher than the false alarm rate ($p^{FA} = 0.01$). Intuitively, this is caused by the inconsistency between the sensors and the images, with the sensor values being based on the true drone position whereas the attacked image were consistent with a different positioning value.

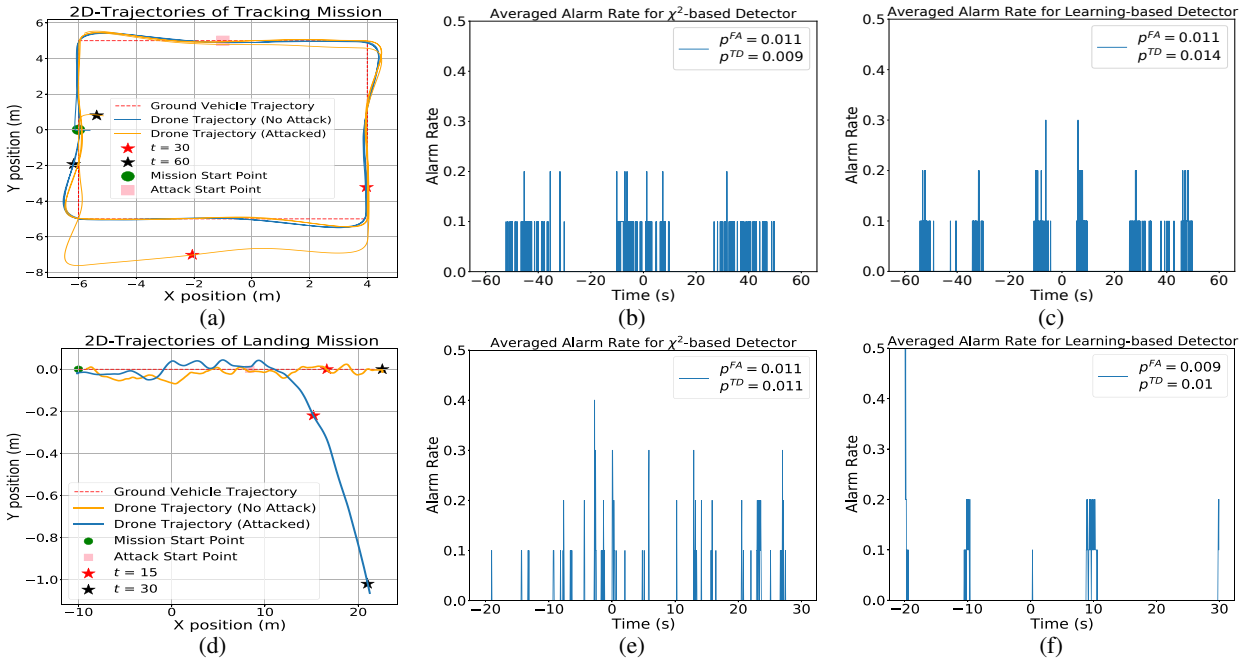

Fig. 4: (a) The trajectories of the UAV and ground vehicle in XY coordinates over time for the GVT. The blue and orange lines show the UAV trajectory without and during the attack, respectively; (b,c) The average alarm rate for χ^2 and learning-based anomaly detectors before and after the attack (attack starts at time $t = 0$) for the GVT task over 10 experiments; (d) The trajectories of the drone and the ground vehicle in XY coordinate for VTOL; (e,f) The alarm rate for χ^2 and learning-based anomaly detectors before and after the attack (starting at $t = 0$) for the VTOL task over 10 experiments.

TABLE III: Averaged true detection rate of χ^2 and learning-based anomaly detectors for different values of initial condition norm ($\|\mathbf{s}_0\|$) in VTOL task when $p^{FA} = .01$.

	p^{TD} (χ^2)	p^{TD} (learning-based)
$\|\mathbf{s}_0\| = .001$	.01	.01
$\|\mathbf{s}_0\| = .01$	.011	.01
$\|\mathbf{s}_0\| = .1$	.55	.79

TABLE IV: Detection rate of χ^2 and learning-based ADs for different values of α (deviation) in meter when only the image is under attack in GVT task when $p^{FA} = .01$.

	$\alpha = .1$	$\alpha = .2$	$\alpha = .3$	$\alpha = .4$	$\alpha = .5$
p^{TD} (χ^2)	.17	.33	.49	.65	.81
p^{TD} (learning-based)	.17	.33	.49	.64	.67

C. Attacks on Vertical Take-off and Landing

The VTOL mission was to land the drone on the *mobile* landing mark after it was detected by the vision module. The dashed red line in Fig. 4d shows the trajectory of the target ground vehicle; when the drone was not under attack, the drone was able to follow the vehicle and land on the marker (drone trajectory is shown in orange line). The blue line presents the drone trajectory when it was under attack. The black and red stars show the same time on both the under attack and non-attacked trajectories; thus, the attacker was able to deviate the drone 1 m away from the marker when the ground vehicle got out of range of the camera.

In this experiment, we also assumed that the elements of the initial error vector $\mathbf{s}_0$ were zero except the one associated with the roll angle which was 0.01. Similar to the GVT, Fig. 4e and Fig. 4f show the average alarm rate at each time step for 10 experiments; for χ^2 and learning-based anomaly detectors, respectively. Assuming the attack starts at time $t = 0$, the average true detection rate (for $t > 0$) $p^{TD} = 0.01$ was almost the same as the average false detection rate (for $t \geq 0$) for both detectors, demonstrating the attack's ϵ-stealthiness.

Moreover, Table III captures the impact of the norm of the initial error vector $\mathbf{s}_0$ on the attack stealthiness level – the true detection rate increases with the increase in the norm of $\mathbf{s}_0$. Finally, Table IV shows the true detection rate for different deviation levels for the VTOL task. As for the GVT, the attack will be detected if the attacker only compromises the image without consistently attacking the sensors.

V. Conclusion

In this work, we have analyzed vulnerability of quadcopters to stealthy attacks on both camera images and other sensors' measurements. We have provided an attack design framework to consistently falsify both the images and physical sensors data. Specifically, we have shown that the attacker has to exploit knowledge of the system dynamics to derive the suitable falsified images and sensing measurements, for the attack to be effective while remaining stealthy from any anomaly detector. In the experiments, we have considered two different tasks – ground vehicle tracking (GVT) and vertical take off and landing (VTOL) and evaluated the impact of the attacks on that system. To detect the presence of attack we have considered a physics-based (χ^2) and a learning-based anomaly detector. We have shown that the attacks result in significant drone trajectory deviations while remaining stealthy from the employed anomaly detectors.

REFERENCES

[1] I. H. Beloev, "A review on current and emerging application possibilities for unmanned aerial vehicles," *Acta technologica agriculturae*, vol. 19, no. 3, pp. 70–76, 2016.

[2] N. M. Rodday, R. d. O. Schmidt, and A. Pras, "Exploring security vulnerabilities of unmanned aerial vehicles," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 993–994.

[3] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in *2012 IEEE Conference on Technologies for Homeland Security (HST)*. IEEE, 2012, pp. 585–590.

[4] K. Hartmann and C. Steup, "The vulnerability of uavs to cyber attacks-an approach to the risk assessment," in *2013 5th international conference on cyber conflict (CYCON 2013)*. IEEE, 2013, pp. 1–23.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] N. Patel, P. Krishnamurthy, A. Tzes, and F. Khorrami, "Overriding learning-based perception systems for control of autonomous unmanned aerial vehicles," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 258–264.

[7] C. Fu, S. Li, X. Yuan, J. Ye, Z. Cao, and F. Ding, "Ad 2 attack: Adaptive adversarial attack on real-time uav tracking," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5893–5899.

[8] J. Tian, B. Wang, R. Guo, Z. Wang, K. Cao, and X. Wang, "Adversarial attacks and defenses for deep learning-based unmanned aerial vehicles," *IEEE Internet of Things Journal*, 2021.

[9] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2009, pp. 911–918.

[10] Mo, Yilin and Sinopoli, Bruno, "False data injection attacks in control systems," in *First workshop on Secure Control Systems*, 2010, pp. 1–6.

[11] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 1806–1813.

[12] A. Khazraei and M. Pajic, "Attack-resilient state estimation with intermittent data authentication," *Automatica*, vol. 138, p. 110035, 2022.

[13] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 82–92, 2015.

[14] C.-Z. Bai, F. Pasqualetti, and V. Gupta, "Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs," *Automatica*, vol. 82, pp. 251–260, 2017.

[15] T. Sui, Y. Mo, D. Marelli, X. Sun, and M. Fu, "The vulnerability of cyber-physical system under stealthy attacks," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 637–650, 2020.

[16] A. Khazraei and M. Pajic, "Resiliency of nonlinear control systems to stealthy sensor attacks," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 7109–7114.

[17] A. Khazraei and M. Pajic, "Perfect attackability of linear dynamical systems with bounded noise," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 749–754.

[18] M. Pajic, I. Lee, and G. J. Pappas, "Attack-resilient state estimation for noisy dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 82–92, March 2017.

[19] I. Jovanov and M. Pajic, "Relaxing integrity requirements for attack-resilient cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 4843–4858, Dec 2019.

[20] A. Khazraei, R. S. Hallyburton, Q. Gao, Y. Wang, and M. Pajic, "Learning-based vulnerability analysis of cyber-physical systems," in *13th IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, 2022.

[21] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[22] A. Paris, B. T. Lopez, and J. P. How, "Dynamic landing of an autonomous quadrotor on a moving platform in turbulent wind conditions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9577–9583.

[23] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle," *Journal of intelligent & robotic systems*, vol. 61, no. 1, pp. 221–238, 2011.

[24] O. Araar, N. Aouf, and I. Vitanov, "Vision based autonomous landing of multirotor uav on moving platform," *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 369–384, 2017.

[25] J. Lin, Y. Wang, Z. Miao, H. Zhong, and R. Fierro, "Low-complexity control for vision-based landing of quadrotor uav on unknown moving platform," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5348–5358, 2021.

[26] S. Dean and B. Recht, "Certainty equivalent perception-based control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 399–411.

[27] C. Kwon, W. Liu, and I. Hwang, "Analysis and design of stealthy cyber attacks on unmanned aerial systems," *Journal of Aerospace Information Systems*, vol. 11, no. 8, pp. 525–539, 2014.

[28] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proceedings of the 2018 workshop on cyber-physical systems security and privacy*, 2018, pp. 72–83.

[29] A. Khazraei, H. Pfister, and M. Pajic, "Resiliency of perception-based controllers against attacks," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 713–725.

[30] A. Khazraei, H. Pfister, and M. Pajic, "Attacks on perception-based control systems: Modeling and fundamental limits," *arXiv preprint arXiv:2206.07150*, 2022.

[31] R. S. Hallyburton, A. Khazraei, and M. Pajic, "Optimal myopic attacks on nonlinear estimation," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 5480–5485.

[32] A. Lambert, A. Shaban, A. Raj, Z. Liu, and B. Boots, "Deep forward and inverse perceptual models for tracking and prediction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 675–682.

[33] "Recorded video," https://drive.google.com/drive/folders/11GwQaKB_9Aclg88QvLTKNEgFyl8wPPIg.

[34] "Prometheus - open source autonomous drone project," https://github.com/amov-lab/Prometheus.