Control Engineering Practice 83 (2019) 188–202

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Detection and mitigation of actuator attacks on small unmanned aircraft systems

Devaprakash Muniraj, Mazen Farhood *

Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061, United States

ARTICLE INFO

Keywords:
Unmanned aircraft system
Actuator attacks
Pulse-width modulation
Watermarking
Estimation

ABSTRACT

Unmanned aircraft systems (UAS) are susceptible to malicious attacks originated by intelligent adversaries, and the actuators constitute one of the critical attack surfaces. In this paper, the problem of detecting and mitigating attacks on the actuators of a small UAS is addressed. Three possible solutions of differing complexity and effectiveness are proposed to address the problem. The first method involves an active detection strategy, whereby carefully designed excitation signals are superimposed on the control commands to increase the detectability of the attack. In the second method, an unknown input observer is designed, which in addition to detecting the attack also estimates the magnitude of the attack. The third method entails designing an actuator system that makes use of variable frequency pulse-width modulated signals to improve the resilience of the actuator against malicious attacks. The effectiveness of the proposed methods is demonstrated using flight experiments and realistic MATLAB simulations that incorporate exogenous disturbances, such as steady winds, atmospheric turbulence, and measurement noise.

1. Introduction

Small unmanned aircraft systems (sUAS) are increasingly employed in diverse applications, such as aerial photography (Colomina & Molina, 2014; Watts and Perry et al., 2010), infrastructure inspection (Montambault, Beaudry, Toussaint, & Pouliot, 2010; Mulero-Pázmány, Negro, & Ferrer, 2013), precision agriculture (Zhang & Kovacs, 2012), emergency management (Merino, Caballero, Martínez-de Dios, Maza, & Ollero, 2012), and package transport (Hoareau, Liebenberg, Musial, & Whitman, 2017). With the rising demand for integration of sUAS into the national airspace (Federal Aviation Administration, 2017), the need to address the vulnerabilities of sUAS to security threats is becoming more urgent. Small unmanned aircraft systems, especially the ones used in civilian applications, are acutely susceptible to security threats because of the public availability of information related to their subsystems and architecture (Kim, Wampler, Goppert, & Hwang, 2012). This information enables adversaries to launch sophisticated attacks, such as stealthy data integrity attacks, which are much more effective than passive attacks, such as eavesdropping. The publicly available information is also being used by researchers to demonstrate the vulnerabilities of existing unmanned aircraft systems by successfully compromising the navigation sensors (Kerns, Shepard, Bhatti, & Humphreys, 2014), ground control station commands and telemetry data (Yoon, Liu, Hovakimyan, & Sha, 2017), and the gyroscopic sensors (Son and Shin et al., 2015).

A variety of approaches have been proposed in the literature to ensure the safety and security of sUAS against various types of cyber–physical attacks, see Birnbaum and Dolgikh et al. (2016), Mitchell and Chen (2014), Muniraj and Farhood (2017) and Yoon et al. (2017). However, almost all of these approaches only address security threats to the sensors, the communication link, or the autopilot system. Little attention has been devoted to address the security vulnerabilities of the actuators of an sUAS. While the approaches studied herein apply to sUAS in general, the focus of the presentation will be on fixed-wing sUAS, as the physical platform used in the experimental validation is a fixed-wing radio-controlled (R/C) aircraft. The actuators of a typical fixed-wing sUAS consist of servo motors and a DC motor. The servo motors are used to actuate the three control surfaces, namely, the elevator, aileron, and rudder, and the DC motor controls the propeller. The inputs to the servo motors and the DC motor are in the form of pulse-width modulated (PWM) signals, which are digital signals that encode analog values. Since the actuators do not interact with any system external to the sUAS, they were considered to be less vulnerable to cyber–physical attacks than the other subsystems in the sUAS, such as the sensors, the communication link, and the autopilot. However, just as gyroscopic sensors, which are equally isolated from external systems, have been shown in Son et al. (2015) to be susceptible to attacks using acoustic signals, actuators too can be attacked by altering the PWM signals using intentional electromagnetic interference (Selvaraj and Dayanıklı et al., 2018). Since civilian sUAS do not in general have redundant actuators, an attack on even one of the actuators may result in loss of control of the sUAS. In such a scenario, it is essential to have detection and mitigation schemes in place to overcome the adverse effects of actuator attacks.

* Corresponding author.
E-mail addresses: devapm@vt.edu (D. Muniraj), farhood@vt.edu (M. Farhood).

https://doi.org/10.1016/j.conengprac.2018.10.022
Received 30 January 2018; Received in revised form 30 July 2018; Accepted 28 October 2018
Available online 1 December 2018

A considerable body of work exists on actuator fault detection and mitigation for aircraft (Bateman, Noura, & Ouladsine, 2011; Ducard & Geering, 2008; Heredia, Ollero, Bejar, & Mahtani, 2008; Yu & Jiang, 2015; Zhang & Jiang, 2008). The main assumption in these works is that the fault mode of the actuator is assumed to lie in a finite set which is known a priori, and the different fault detection algorithms determine the fault mode using knowledge about the model of the aircraft. In the case of actuator attacks, the type of attack is not known a priori and because of this crucial difference, most of the available approaches cannot be used for detecting actuator attacks. There is another line of research that addresses the problem of actuator attack detection for systems modeled using linear differential equations. In Pasqualetti, Dörfler, and Bullo (2013), the authors consider different types of attacks, including sensor and actuator attacks, on systems defined by linear descriptor models and propose monitors to detect such attacks. The authors in Fawzi, Tabuada, and Diggavi (2014) consider the problem of state estimation for systems described by linear time-invariant models, where a subset of the sensors and actuators are under attack. The authors in Teixeira, Shames, Sandberg, and Johansson (2015) use a fault detection filter to detect threats such as denial-of-service and replay attacks on linear time-invariant systems. Since the dynamics of sUAS are highly nonlinear and their operational environment is uncertain, the solutions proposed in these works cannot be applied for detecting actuator attacks on sUAS. The present work is aimed at addressing the specific challenges that lie in detecting and mitigating actuator attacks for sUAS.

In this work, two different approaches are considered in addressing the problem. The first approach, called the *software approach*, involves developing algorithms for detection of actuator attacks and does not require any hardware modifications. However, a separate mitigation strategy is required to ensure safe operation of the sUAS in the event of an attack. Under this approach, two methods are proposed: the active detection method and the estimation-based detection method. In the active detection method, judiciously designed excitation signals are superimposed on the control commands to increase the detectability of the actuator attack. The estimation-based detection method involves the use of an unknown input observer, which in addition to detecting the actuator attack also estimates the magnitude of the attack. The second approach, called the *resilient hardware approach*, focuses on identifying the vulnerabilities of the existing actuators and using that information to design actuators that are resilient to malicious attacks. The proposed method in this case entails designing an actuator that uses variable frequency pulse-width modulated signals to improve its resilience against actuator attacks. Thus, the resilient hardware approach does not require a separate mitigation strategy.

The paper is organized as follows. Section 2 presents background information on the operation and security vulnerabilities of servo motors used in sUAS. A brief description of the sUAS platform used in the flight tests is provided in Section 3. In Section 4, the two methods under the software approach are presented along with simulation and flight test results. The resilient hardware approach and a detailed analysis of its effectiveness are presented in Section 5. Finally, conclusions and some topics of future work are discussed in Section 6.

2. Servo motors used in sUAS

This section provides a brief overview of the operation and security vulnerabilities of a typical servo motor used as an actuator in an sUAS.

2.1. Servo motor operation

A servo motor used in an sUAS consists of six different components, namely, a DC motor, a control horn, a gear reduction system, a potentiometer, a servo plug, and a microprocessor, as shown in Fig. 1. The DC motor is connected to the control horn through the gear reduction system, whose purpose is to increase the torque. The potentiometer is attached to the servo shaft and provides measurement of the shaft position to the internal control circuit. Based on the incoming reference signal and the signal received from the potentiometer, the internal control circuit of the microprocessor controls the speed of the DC motor. The servo plug has three wires, out of which one wire provides positive voltage to the servo typically in the range of 4.8 V to 6 V, the second wire serves as the ground, and the third wire provides the reference signal to the servo. The rotational range of most servos is limited to ± 90 deg. Depending on the manner in which the incoming reference signal is processed, the servo motors are classified as either analog servos or digital servos. In an analog servo, the power sent to the DC motor is in the form of constant voltage low frequency pulses with a frequency of 50 pulses/s, and the length of each pulse is varied until full power is applied to the motor. When the desired servo position is reached and no external force is applied on the control horn, no power is sent to the DC motor, thereby ensuring that the control horn is not continuously rotating. As a result of using low frequency voltage pulses, analog servos tend to respond sluggishly to commands and have a large deadband. Digital servos, which operate using high frequency voltage pulses of the order of 300 pulses/s, overcome many of the limitations of analog servos. Since the power sent to the motor is turned on/off more frequently, digital servos respond faster to the commands and also have a lower deadband. Analog servos are increasingly being replaced by digital ones in unmanned aircraft systems.

The input to the servo motor is a constant frequency pulse-width modulated (PWM) signal with a frequency of 50 pulses/s. A PWM signal is essentially a rectangular pulse train, where the signal can be either in a high corresponding to a voltage of 5 V or a low corresponding to a voltage of 0 V. However, in many servos a threshold voltage of 2.8 V is used to account for noise in the signal. If the voltage of the signal is less than the threshold voltage, then it is registered as a low. By modulating the time the signal is in a high, the PWM signal is able to encode analog values. A rising edge is the time at which the voltage transitions from a low to a high, and a falling edge is defined as the time when the voltage transitions from a high to a low. The difference between the falling edge and the rising edge within a time period is called the pulse width. Given an analog reference signal $s_r(t)$, the corresponding PWM signal $s_p(t)$ can be mathematically written as $s_p(t) = \max(\text{sgn}(s_r(t) - s_c(t)), 0)$, where $s_c(t)$ is a carrier signal, and sgn and max denote the signum function and the maximum function, respectively. Based on the type of the carrier signal used, PWM signals are classified as either leading-edge modulated signals, trailing-edge modulated signals, or double-edge modulated signals (Sun, 2012). The PWM signals used to control the servos of an sUAS fall under the category of trailing-edge modulated signals, where the rising edge of the signal occurs at fixed instants of time and the trailing edge is modulated as the reference signal varies. A sawtooth signal with a time period of 20 ms is used as the carrier signal to generate a trailing-edge modulated PWM signal with a pulse frequency of 50 pulses/s. The angular position of the servo control horn is determined by the pulse width, where pulse widths of 1.9 ms and 1.1 ms correspond to the two extreme angular positions of the servo and a pulse width of 1.5 ms corresponds to the nominal servo position.

2.2. Security vulnerabilities of servo motors

The servo plug provides an easy attack surface because of its susceptibility to electromagnetic interference. The authors in Selvaraj et al. (2018) exploited this attack surface to modify the PWM signal using intentional electromagnetic interference. The attacker circuit in that case consisted of a waveform generator, a buffer amplifier, and a solenoid or an antenna. The attacker's objective is to design a waveform such that the antenna generates magnetic fields that induce time-varying voltages in the servo plug. The effect of the attack is to cause a negative voltage in the signal wire, thereby modifying the trailing edge of the pulse. For instance, if the attacker manages to induce a negative voltage of 2.2 V in the signal wire of the servo plug, then the trailing edge of

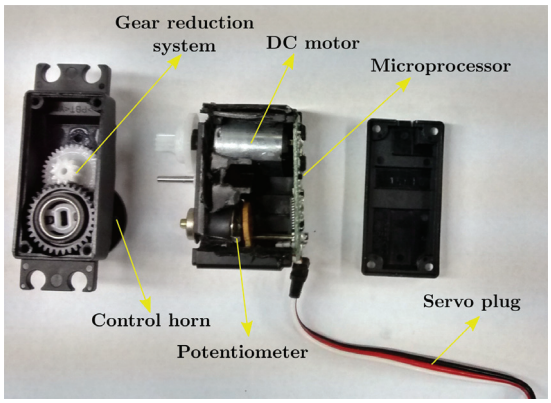


Fig. 1. A disassembled Futaba S3152 servo.

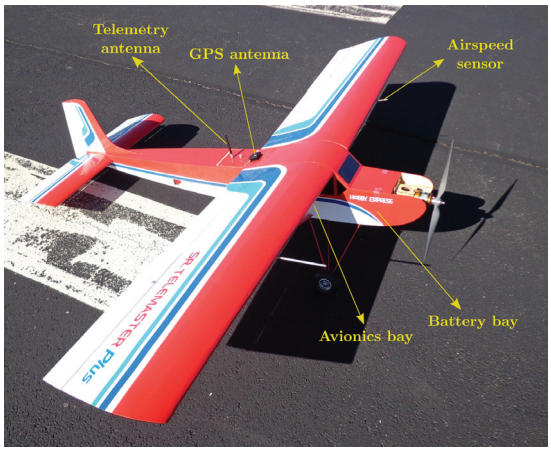


Fig. 2. The sUAS platform used in the flight tests.

the pulse can be modified by varying the phase difference between the induced signal and the original PWM signal. The attacker only needs to induce a negative voltage of 2.2 V and not 5 V because of the tolerance voltage of 2.8 V set in the servo. One simple solution to prevent such attacks would be to lower the tolerance voltage so that the attacker is forced to expend more energy to induce a larger negative voltage. However, the downside to this simple solution of lowering the tolerance voltage is that more noise is now allowed into the system, thereby reducing the accuracy and reliability of the servo.

Based on the above model, the attacked signal, denoted by $\bar{u}(t)$ and expressed in milliseconds, is related to the original signal $u(t)$ through the following relation:

$$\bar{u}(t) = u(t) + b(t), \text{ where } (-u(t) + 1.1) \leq b(t) \leq 0. \quad (1)$$

Under the above attack model, the attacker can only reduce the magnitude of the control signal $u(t)$ and not increase it. In order for the attack to be successful and effective, the attacker must possess knowledge about the pulse frequency of the PWM signal and also maintain a small separation distance between the antenna and the servo plug. The attack model described here is based on Selvaraj et al. (2018), where a realistic and practical actuator attack is discussed. The methods proposed in this work are not restricted to this attack model only, but are rather applicable to more general attacks, as will be seen later.

3. sUAS platform

This section provides a brief description of the sUAS platform used in the flight tests. A Senior Telemaster Plus fixed-wing radio-controlled aircraft from Hobby Express (2018) is the aerial platform of the sUAS. The aircraft has an electric-motor driven propeller and three control surfaces, namely, the elevator, aileron, and rudder. The control surfaces are actuated by Futaba S3152 digital servos, and the electric motor driving the propeller is controlled by a Jeti Advance 70 Pro Opto brushless electronic speed controller. The mass of the sUAS is 5.71 kg. The wingspan and the mean aerodynamic chord of the aircraft are 2.39 m and 0.3594 m, respectively.

The sUAS is controlled by a fully programmable autopilot system consisting of a 3DR Pixhawk (Pixhawk, 2018) and a Gumstix Overo Fire (Gumstix, 2017). A customized version of the open-source ArduPilot firmware is used in the Pixhawk. The Pixhawk contains a 168 MHz, 32 bit Cortex M4 core processor with 256 KB RAM. The Pixhawk is used for basic input/output tasks and redundancy management in the event of failures in the controller. The computational resources of the Pixhawk are not adequate for handling more computationally demanding algorithms, and so a Gumstix Overo Fire is used as the back-end onboard computer to execute these tasks. The brain of the Gumstix consists of a 600 MHz TI-OMAP 3530 with ARM Cortex A8 processor. The Gumstix uses a version of Linux called the Yocto project as the operating system and provides sufficient computing power for running computationally demanding algorithms in near-real-time. The detection algorithms presented in this work and the path-following controllers are coded in the Python programming language and executed on the Gumstix. A serial interface is used as the interconnection between the Pixhawk and the Gumstix.

The sUAS platform contains the following sensors: a MS5611 barometric pressure sensor for altitude measurement, a 4525DO differential pressure sensor for airspeed measurement, a U-blox NEO-7 GPS module for position measurement, and an MPU 6000 inertial measurement unit (IMU). While the pressure sensor and the IMU are mounted inside the Pixhawk, the differential pressure sensor and the GPS module are connected to the Pixhawk though the I^2C and the serial interfaces, respectively. Every 40 ms, sensor data from the Pixhawk is sent to the Gumstix and control commands from the Gumstix are sent to the Pixhawk. The sUAS also transmits data to a ground control station through an XBEE 900 Pro, point-to-multipoint radio modem. The sUAS platform is shown in Fig. 2 with the important subsystems marked on it.

4. Detection of actuator attacks using the software approach

This section presents the active detection method and the estimation-based detection method. The description of each method is followed by detailed analysis using nonlinear simulations and validation through flight experiments. The merits and shortcomings of each method are also discussed. The main difference between the two methods is that the active detection method is not model dependent, whereas the estimation-based detection method requires a mathematical model of the sUAS. Although the active detection method does not explicitly require a model of the sUAS, if available, a mathematical model, even a low fidelity one, can be utilized in tuning the parameters of the resultant watermarking-based detector; otherwise, the parameter tuning has to be done based on flight test data.

4.1. Active detection method

The basic idea of the active detection method is inspired from digital watermarking (Cox, Miller, & Bloom, 2002), which is a process used to secure digital data. In digital watermarking, the sender of a digital file merges the contents of the file with information related to the file called a watermark; the receiver of the file then uses the watermark to verify the authenticity of the digital file. In the active detection method outlined here, the control commands sent to the actuators act as the digital file and judiciously designed excitation signals that are superimposed with the control commands act as the

watermarks. Since sUAS do not have sensors to measure the actual angular displacement of the control surfaces, other measurements from the IMU that provide information about the actual control commands applied to the control surfaces are analyzed. The watermarking-based detector relies on the following observation: when the actuators are not under attack, the intended effects of the watermarks must be observed in the sensor measurements; a failure to observe the intended effects of the watermarks is an indication of an attack on the actuators.

Since the excitation signals act as additional disturbances to the system, it is important to ensure that their deleterious effects on the performance of the sUAS are minimized while at the same time increasing their effectiveness in detecting the actuator attacks. In this regard, the excitation signals must be able to excite the aircraft dynamics with minimum excursions in the position and attitude of the sUAS. Ideally, the variations in position and attitude due to the excitation signals must be indistinguishable from the response of the sUAS to atmospheric disturbances so that it is not obvious to the attacker that an active detection system is used. The design of the watermarking-based detector is explained in the following paragraphs.

Design of watermarking signals

There are four watermarking signals associated with the four control inputs. To minimize the time interval of excitation, all the watermarking signals are simultaneously applied to the control commands. For the watermarking-based detector to effectively isolate the actuator under attack, it is important that the responses of the sUAS to the four watermarking signals be decoupled. Multitone signals, which are sums of harmonic signals and frequently used in frequency-domain system identification of aircraft, are chosen as the watermarking signals. In particular, a two-tone input consisting of a sum of harmonic sinusoids with different phase lags is used. The advantage of using multitone signals as the watermarks is that the signals can be designed to be orthogonal to each other, thereby ensuring that the sUAS responses to the watermarking signals are decoupled. The design of the watermarking signals essentially entails choosing the frequencies and the phase lags of the constituent harmonic signals such that the energy content of each watermarking signal is maximized while minimizing the amplitude of the signal. To simplify the exposition, the following sets and variables are defined.

Suppose there are N sets of watermarking signals, for some integer $N > 0$. Let $\mathbf{W}_i = \{\mathbf{u}_{E,i}, \mathbf{u}_{A,i}, \mathbf{u}_{R,i}, \mathbf{u}_{T,i}\}$ denote the ith set of watermarking signals, where $\mathbf{u}_{E,i}$, $\mathbf{u}_{A,i}$, $\mathbf{u}_{R,i}$, and $\mathbf{u}_{T,i}$ designate vectors that represent the watermarking signals corresponding to the elevator, aileron, rudder, and throttle control inputs, respectively. Each of these vectors has m_i entries, where each entry corresponds to the value of the signal at a particular time instant. Let T_i denote the time interval of the ith watermarking signals, which is computed as $T_i = (m_i - 1)\Delta t$, where Δt is the sampling period of the signals. Let $\mathbf{W} = \{\mathbf{W}_1, \dots, \mathbf{W}_N\}$ denote the set consisting of the N sets of watermarking signals. Every T_w seconds, for some judiciously chosen positive scalar $T_w > T_i$, the watermarking-based detector selects a set from $\mathbf{W}$ at random and superimposes the corresponding watermarking signals with the control commands from the sUAS controller. By pseudorandomly changing the watermarking signals, it is ensured that the attacker will not be able to predict the future watermarks, thereby preventing any countermeasures from the attacker. A monitoring interval is defined as a time interval of length T_w seconds starting from the time when the watermarking signals are injected. Since $T_w > T_i$, the monitoring interval is larger than the time interval of the watermarking signals and is chosen as such to completely capture the responses of the sUAS to the watermarking signals. As some of the dynamic modes of the sUAS are lightly damped, the effect of the watermarking signals on sUAS outputs such as the linear and angular accelerations persists beyond the duration of the watermarking signals, thus necessitating a value of T_w greater than T_i to fully represent the outcomes of applying these signals. A specific value for T_w depends on the sUAS platform among other factors, and further details on how this value is chosen will be provided later in this subsection when evaluating the method through simulations and experiments. The ith watermarking signals are given by

$$\mathbf{u}_{x,i} = \begin{bmatrix} u_{x,i}(0) & u_{x,i}(1) & \cdots & u_{x,i}(m_i - 1) \end{bmatrix}^T, \text{ for } x = E, A, R, T,$$
$$\text{where, for } j = 0, \dots, m_i - 1,$$

$$u_{E,i}(j) = a_1 \cos(j\omega_{1,i}\Delta t + \phi_{1,i}) + a_2 \cos(j\omega_{2,i}\Delta t + \phi_{2,i}),$$
$$u_{A,i}(j) = a_3 \cos(j\omega_{3,i}\Delta t + \phi_{3,i}) + a_4 \cos(j\omega_{4,i}\Delta t + \phi_{4,i}),$$
$$u_{R,i}(j) = a_5 \cos(j\omega_{5,i}\Delta t + \phi_{5,i}) + a_6 \cos(j\omega_{6,i}\Delta t + \phi_{6,i}),$$
$$u_{T,i}(j) = a_7 \cos(j\omega_{7,i}\Delta t + \phi_{7,i}) + a_8 \cos(j\omega_{8,i}\Delta t + \phi_{8,i}).$$

The frequencies $\omega_{1,i}, \dots, \omega_{8,i}$ are distinct harmonics of the fundamental frequency $2\pi/T_i$. By choosing the frequencies to be distinct harmonics, the set of watermarking signals is ensured to be orthogonal. The watermarking signal design problem boils down to appropriately choosing the frequencies $\omega_{k,i}$ and phase differences $\phi_{k,i}$ for $k = 1, \dots, 8$ and $i = 1, \dots, N$.

The frequencies $\omega_{k,i}$, for $k = 1, \dots, 8$ and $i = 1, \dots, N$, are chosen within the interval $[2\pi/T_i \;\; \bar{\omega}]$, where $\bar{\omega}$ is to be selected based on the bandwidth of the aircraft's frequency response. Additionally, $\bar{\omega}$ should satisfy the constraint $\bar{\omega} < \pi/\Delta t$ to prevent aliasing, where $\pi/\Delta t$ is the Nyquist frequency. Since the dynamics of an sUAS are nonlinear, it is important to take into account the effect of nonlinearities on the output response of the sUAS during the selection of the $\omega_{k,i}$'s. If the input to a nonlinear system is a sum of sinusoids of different frequencies, then the output frequencies not only contain the input frequencies, but also the harmonics and the intermodulation frequencies (Rugh, 1981). Thus, to make sure that a watermarking signal in one channel, say the elevator channel, does not influence the responses of the sUAS to the watermarking signals of the other three channels, the frequencies $\omega_{3,i}, \dots, \omega_{8,i}$ must be chosen to be different from the harmonic frequencies and the intermodulation frequencies corresponding to $\omega_{1,i}$ and $\omega_{2,i}$.

Consider the following assumptions: (i) the velocity of the sUAS is constant during the time interval when the watermarking signals are injected; (ii) the small angle approximations hold for the trigonometric terms containing the angle of attack (AoA) and the angle of sideslip (AoSS); and (iii) the aerodynamic model is linear in the angular rates, AoA, AoSS, and the control inputs. Under these assumptions, it can be easily verified from the aircraft equations of motion, presented in Stevens, Lewis, and Johnson (2015) among others, that the input–output map from the control inputs $(\delta_E, \delta_A, \delta_R, \delta_T)$ to the accelerations $(a_x, a_y, a_z, \dot{p})$ contains only second-order nonlinear terms pertaining to the inertial couplings. Here, a_x, a_y, a_z denote the three linear accelerations, and $\dot{p}$ denotes the roll acceleration. The assumptions (i)-(iii) are reasonable in this study given the proposed excitation signals and the fact that the sUAS is assumed to operate within normal flight regimes, where nonlinear aerodynamic effects due to high angles of attack are not present. Thus, the sUAS is approximated as a second-order nonlinear system for the purposes of this work. Volterra–Wiener theory of nonlinear systems (Rugh, 1981) can be used to determine the frequency response of a second-order nonlinear system to a two-tone watermarking signal. Given a second-order nonlinear system subjected to a two-tone input with frequencies ω_1 and ω_2, the output spectrum consists of the following frequency components (Wu, Lang, & Billings, 2007): the input frequencies ω_1 and ω_2, the harmonics $2\omega_1$ and $2\omega_2$, and the intermodulation frequencies $\omega_1 + \omega_2$ and $|\omega_1 - \omega_2|$. Bearing in mind the preceding observations, the frequencies $\omega_{1,i}, \dots, \omega_{8,i}$ are chosen using Algorithm 1. If the algorithm does not compute all the frequencies for some $i \in \{1, \dots, N\}$, then the corresponding time interval T_i needs to be increased until all the frequencies are chosen.

Once the frequencies $\omega_{k,i}$, for $k = 1, \dots, 8$ and $i = 1, \dots, N$, are chosen, the phase lags $\phi_{k,i}$ need to be computed to complete the design of the watermarking signals. As mentioned earlier, each watermarking signal is to be designed such that the energy content of the signal is maximized while minimizing the amplitude of the signal. The crest factor (or peak

Algorithm 1: Selecting the frequencies of the watermarking signals

Inputs: time-intervals T_i, for $i = 1, \dots, N$, and $\bar{\omega}$.
Outputs: watermarking frequencies $\omega_{k,i}$ for $k = 1, \dots, 8$ and $i = 1, \dots, N$.
procedure
  for $i = 1 : N$ **do**
    Initialize the set $\mathbf{\Theta}_1$ which consists of the harmonics of $2\pi/T_i$ within the interval $[2\pi/T_i \ \ \bar{\omega}]$.
    Initialize the sets $\mathbf{\Theta}_2 = \{\ \}$, $\mathbf{\Theta}_3 = \{\ \}$.
    for $j = 1 : 4$ **do**
      Choose frequencies $\omega_{2j-1,i} \in \mathbf{\Theta}_1$ and $\omega_{2j,i} \in \mathbf{\Theta}_1$ such that
      $\omega_{2j-1,i} \neq \omega_{2j,i}$, $\omega_{2j-1,i} \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$, $\omega_{2j,i} \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$,
      $2\omega_{2j-1,i} \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$, $2\omega_{2j,i} \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$,
      $(\omega_{2j-1,i} + \omega_{2j,i}) \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$, and $|\omega_{2j-1,i} - \omega_{2j,i}| \notin \mathbf{\Theta}_2 \cup \mathbf{\Theta}_3$.
      Update the sets $\mathbf{\Theta}_2$ and $\mathbf{\Theta}_3$
      $\mathbf{\Theta}_2 \leftarrow \{\omega_{2j-1,i}, \omega_{2j,i}\} \cup \mathbf{\Theta}_2$,
      $\mathbf{\Theta}_3 \leftarrow \{2\omega_{2j-1,i}, 2\omega_{2j,i}, (\omega_{2j-1,i} + \omega_{2j,i}), |\omega_{2j-1,i} - \omega_{2j,i}|\} \cup \mathbf{\Theta}_3$.

factor) is a useful measure that relates the peak amplitude of a signal to its root mean square (rms) value (Boyd, 1986). The crest factor of a signal represented by the vector $\mathbf{u}$ is denoted by $CF(\mathbf{u})$ and is defined as $CF(\mathbf{u}) = \|\mathbf{u}\|_\infty / \|\mathbf{u}\|_2$, where $\|\mathbf{u}\|_\infty$ and $\|\mathbf{u}\|_2$ are the ∞-norm and 2-norm of $\mathbf{u}$, respectively. The design objective then is to choose the phase lags $\phi_{k,i}$ such that the crest factor of each watermarking signal is minimized. This problem is very similar to the problem of designing optimal multi-axes inputs in frequency-domain system identification. Morelli (2003) provides a simple algorithm to compute the phase lags of a multitone signal such that the crest factor of the signal is minimized. The amplitudes $a_1, \dots, a_8$ are the inputs to the algorithm in addition to the frequencies $\omega_{k,i}$. Due to paucity of space, the algorithm is not provided here; see Morelli (2003) for more details.

A representative set of watermarking signals designed using the above method is shown in Fig. 3. The following parameters are used in designing the watermarking signals: a time interval of 2 s, a sampling period of 40 ms, $\bar{\omega} = 6.0$ Hz, and amplitudes $a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = a_7 = a_8 = 0.1$. The following watermarking frequencies are obtained following Algorithm 1: $\omega_1 = 1.0$ Hz, $\omega_2 = 2.0$ Hz, $\omega_3 = 1.25$ Hz, $\omega_4 = 2.5$ Hz, $\omega_5 = 1.75$ Hz, $\omega_6 = 3.5$ Hz, $\omega_7 = 2.25$ Hz, and $\omega_8 = 4.5$ Hz.

Detection principle

The detection principle uses the frequency contents of the three linear accelerometer measurements a_x, a_y, a_z and the roll acceleration measurement $\dot{p}$. Specifically, the frequency content of each of the linear accelerations a_x, a_y, and a_z is analyzed to detect actuator attacks in the throttle channel, the rudder channel, and the elevator channel, respectively. The frequency content of the roll acceleration $\dot{p}$ is used to detect actuator attacks in the aileron channel. The reason for choosing a_x, a_y, a_z, and $\dot{p}$ is that these measurements capture the direct feed-through effects from the respective control commands, which was found to enhance the detection capability of the detector.

The detection metric for the elevator channel is denoted by D_E and is given by the following expression:

$$D_E = \frac{P_{a_z}(\omega_{1,i}) + P_{a_z}(\omega_{2,i})}{P_{\delta_E}(\omega_{1,i}) + P_{\delta_E}(\omega_{2,i})}, \tag{2}$$

where $P_{a_z}(\omega_{1,i})$ denotes the normalized power of the signal a_z over the monitoring interval at the frequency $\omega_{1,i}$; the normalization is done with respect to the total power of the signal over the monitoring interval. Thus, $P_{a_z}(\omega_{1,i})$ is a measure of the relative power of the signal a_z concentrated at the frequency $\omega_{1,i}$ compared to the total power of the signal over the monitoring interval. The other terms in (2) are defined similarly. An actuator attack on the elevator channel is detected if $D_E > R_E$, where R_E is a threshold obtained from simulations with a nonlinear model of the sUAS or from flight tests. The detection metrics for the other three channels are similarly defined as

$$D_A = \frac{P_{\dot{p}}(\omega_{3,i}) + P_{\dot{p}}(\omega_{4,i})}{P_{\delta_A}(\omega_{3,i}) + P_{\delta_A}(\omega_{4,i})}, \quad D_R = \frac{P_{a_y}(\omega_{5,i}) + P_{a_y}(\omega_{6,i})}{P_{\delta_R}(\omega_{5,i}) + P_{\delta_R}(\omega_{6,i})}, \quad \text{and}$$
$$D_T = \frac{P_{a_x}(\omega_{7,i}) + P_{a_x}(\omega_{8,i})}{P_{\delta_T}(\omega_{7,i}) + P_{\delta_T}(\omega_{8,i})}.$$

The detection metrics D_E, D_A, D_R, and D_T are computed after T_w seconds from when the watermarking signals are injected, and the following four conditions are checked: $D_E > R_E$, $D_A > R_A$, $D_R > R_R$, and $D_T > R_T$. If a condition is violated, then the corresponding actuator is deemed to be under attack. Watermarking-based detection relies on the premise that the responses of the sUAS to the watermarking signals, as observed from the frequency contents of the linear acceleration measurements and the roll acceleration measurement, are affected by the presence of actuator attacks. Consequently, a zero-frequency attack input, i.e., an attack input that is constant during the monitoring interval, cannot be detected by the watermarking-based detector. This is one of the shortcomings of this approach.

Evaluation of the method through nonlinear simulations and flight experiments

As an illustrative example, the proposed active detection method is used to detect different types of actuator attacks on the sUAS described in Section 3. The effectiveness of the method is studied through extensive nonlinear simulations as well as flight experiments. During the simulations and flight tests, the aircraft is controlled by a linear time-invariant (LTI) $\mathcal{H}_\infty$ path-following controller. The synthesis procedure for the LTI path-following controller and the nonlinear model of the sUAS used in the simulations are described in Muniraj, Palframan, Guthrie, and Farhood (2017). The choice of the controller is arbitrary since the performance of the watermarking-based detector does not depend on the type of controller used.

In the simulations, the atmospheric disturbances applied consist of a 3.5 m/s steady wind in the North-East direction and medium turbulence generated by the low-altitude Dryden wind turbulence model (Gage, 2003). The wind speed and turbulence intensity used in the simulations are typical worst-case conditions that can be handled by sUAS of the type considered in this example. The sensor measurements of the aircraft consist of the airspeed V_a, the body-axis angular rates p, q, and r, the Euler angles ϕ, θ, and ψ, and the aircraft position (N, E, H) in the North-East-Down frame. Each of the measurements is assumed to be corrupted by sensor noise, which is generated in the simulations from a zero-mean normal distribution. The standard deviations, $\sigma_{(\cdot)}$, used for the different measurements are as follows: $\sigma_{V_a} = 2$ m/s, $\sigma_{p/q/r} = 0.5$ deg/s, $\sigma_{\phi/\theta/\psi} = 0.57$ deg, and $\sigma_{N/E/H} = 2$ m. The values for the standard deviations are obtained from sensor specifications and laboratory tests. During the simulations, the aircraft is subjected to four different types of possible actuator attacks: struck actuator attacks where the actuator commands are frozen in time, sinusoidal attacks where the attack input is a sinusoidal signal, random attacks where the attack inputs are generated pseudorandomly, and scaling attacks where the attack inputs vary linearly with time starting from zero.

The type of path the sUAS is tasked to follow, the type of attack, and the intensity of attack are among the factors that may impact the performance of the watermarking-based detector. To study the effects of these factors, extensive simulations are performed. Specifically, three different paths that are widely used in many sUAS applications are considered, namely, a circle of radius 110.5 m, a moderate lemniscate generated as in Muniraj et al. (2017), and a race track composed of two straight line segments of length 500.0 m and two semi-circular segments of radius 110.5 m. The attack intensity is reflected by the magnitude of the attack in the case of a struck actuator attack, the amplitude and frequency of the attack in the case of a sinusoidal attack, and the rate of change of the attack magnitude in the case of a scaling attack. Struck actuator attacks of different magnitudes, ranging from 1.3 ms to 1.7 ms in

steps of 0.1 ms, are included in the simulations. Frequencies ranging from 0.1 Hz to 6.0 Hz in steps of 0.2 Hz and amplitudes ranging from 0.05 ms to 0.3 ms in steps of 0.05 ms are considered for the sinusoidal attack signals. Scaling attacks with different intensities are simulated by changing the rate of variation of the attack signal from 0.05 ms/s to 0.15 ms/s in steps of 0.02 ms/s. The duration of the actuator attack is chosen such that the sUAS does not reach an unrecoverable state during the simulations, since the simulations solely focus on attack detection without employing any mitigation strategies. For each combination of path, type of attack, and intensity of attack, the actuator attack is executed as follows: at each time instant $t_k = 8k$ s for $k = 0, 1, \dots, 1000$, the attack signal is applied for a duration of 4 s. That is, every 8 s an attack signal is applied for the duration of 4 s. The reason for waiting 4 s after the end of an attack signal before starting another attack signal is to allow the aircraft to recover from the previous attack. Depending on the type and intensity of attack, it may be possible to extend the duration of each attack instance beyond 4 s. For instance, the considered random attack and sinusoidal attacks may be applied over longer durations without causing loss of control. However, to enable comparisons between the different types of actuator attacks, the attack duration for all the four types of attacks is set to 4 s.

Ten sets of watermarking signals, denoted by $\mathbf{W}_i$ for $i = 1, \dots, 10$, are designed using the procedure described earlier, where each set corresponds to a time interval T_i given by $T_i = 1.42 + 0.1i$. The values for T_i are chosen such that they are contained within the bandwidth of the aircraft's frequency response. The frequency $\bar{\omega}$, which is one of the inputs to Algorithm 1, is chosen as 6.0 Hz for the sUAS considered in this example. A sampling period of 40 ms and a monitoring interval of 4 s are used. As explained earlier, the monitoring interval has to be larger than the time interval of the watermarking signals. Another consideration in choosing the value of T_w is the number of data points available for computing the normalized power values in (2). An accurate estimate of the normalized power requires a large number of data points, but increasing the value of T_w also delays the detection of the actuator attack. In this study, the multiple signal classification (MUSIC) algorithm (Stoica & Moses, 2005) is used to compute the power spectral density and hence the normalized power. It is found that collecting 100 data points, which requires a monitoring interval of 4 s, is sufficient to accurately estimate the normalized power.

Since the watermarking signals act as disturbances on the control commands, it is important to first verify that the path-following performance of the sUAS is not degraded as a result of the use of the watermarking signals. Simulations are therefore performed in the absence of actuator attacks, where watermarking signals chosen pseudorandomly from $\mathbf{W} := \{\mathbf{W}_1, \dots, \mathbf{W}_{10}\}$ every $T_w = 4$ s are superimposed on the control commands. As in Muniraj et al. (2017), in these simulations the mean path error (MPE) is used as a measure of the path-following performance of the sUAS. For closed paths, such as a lemniscate or a circular path, MPE is computed over each cycle, or completed round, of the path. Specifically, the Euclidean distance between the aircraft position and the closest point on the reference path is measured at each time instant in the cycle, and then the MPE over the cycle is calculated as the mean of these distances. The mean path errors are computed for each cycle of the path and are compared with those from simulations where the watermarking signals are not injected. Fig. 4 shows the distributions of the mean path errors for the lemniscate path for the two cases. It is observed from the figure that the watermarking signals do not have a significant effect on the path-following performance, and the maximum MPE is found to increase by only about 2%.

The parameters of the watermarking-based detector, namely, the thresholds R_E, R_A, R_R, and R_T, are specific to a particular sUAS platform. For the sUAS platform considered in this study, the values for the thresholds are obtained from data gathered during different flight tests when the aircraft is not under any attack. Firstly, the metrics D_E, D_A, D_R, and D_T are computed from the flight test data and their distributions are plotted. Then, each threshold value is chosen as 10% less than the minimum value in the corresponding distribution. The

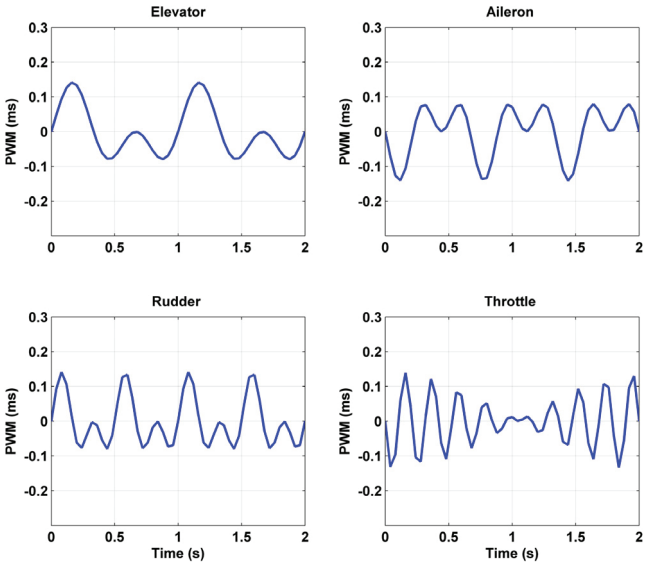


Fig. 3. A representative set of watermarking signals.

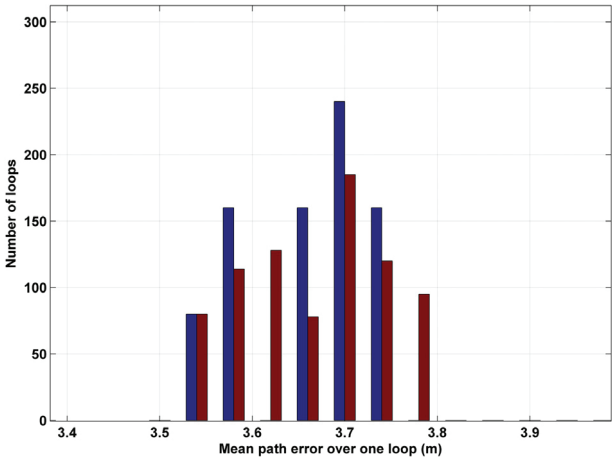


Fig. 4. Effect of watermarking signals on the path-following performance; red bars correspond to the simulations where the watermarking signals are superimposed with the control commands, and blue bars correspond to the simulations without the watermarking signals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

threshold values thus obtained are given as $R_E = 0.43$, $R_A = 0.55$, $R_R = 0.48$, and $R_T = 0.42$.

A summary of the simulation results is provided in Table 1. The performance of the detector is assessed in terms of the true positive rate (TPR), the false positive rate (FPR), and the mean maximum position error before detection (MMPE) for the four different types of attack. MMPE is a measure of the degradation in path-following performance of the sUAS before the actuator attack is detected. It is obtained as follows: the maximum position error before detection is computed for each monitoring interval where an attack is detected, and then MMPE is calculated as the mean of these errors. True positive rate is the probability that the detector correctly detects the actuator attack, whereas false positive rate is the probability that the detector identifies an attack when there is no attack. Given the TPR and the FPR, the false negative rate (FNR) and the true negative rate (TNR) can be easily computed. For instance, a TPR of 99% corresponds to an FNR of 1%. Since the FPR does not depend on the type of attack, all four types of attack have the same FPR, which is computed using data from all the simulation segments where the sUAS is not under an actuator attack. It is observed from Table 1

Table 1
Performance of the watermarking-based detector from simulations and flight tests.

Type of attack	Simulations			Flight tests	
	TPR (%)	FPR (%)	MMPE (m)	TPR (%)	FPR (%)
Struck actuator attack	99.37	0.42	5.41	100.00	0.85
Sinusoidal actuator attack	83.74	0.42	3.59	75.00	0.85
Random actuator attack	99.12	0.42	3.23	100.00	0.85
Scaling actuator attack	60.44	0.42	4.25	70.00	0.85

that the performance of the detector depends on the type of attack: the detector has a better performance for the random actuator attack and the struck actuator attack compared to the sinusoidal and scaling actuator attacks. The reason for the lower detection rates in the case of sinusoidal and scaling actuator attacks is that the attack inputs do not change the output frequency characteristics sufficiently for certain scaling and sinusoidal attacks. This scenario happens when an attack input of small amplitude in the case of a sinusoidal attack or low intensity in the case of a scaling attack has frequency content such that the frequency components with significant energy are concentrated very close to zero. The detector expectedly does not perform well in this case since, as mentioned before, the watermarking-based detector cannot detect zero-frequency attack inputs, and further its performance degrades as the attack inputs approach this limiting case. Due to the specific nature of the attack, the degradation in path-following performance before detection, as measured using the MMPE metric, is higher for the struck actuator attack and the scaling actuator attack compared to the other two attacks. It is observed from the simulations that the performance of the detector is not influenced by the type of path and is only dependent on the type of actuator attack.

In addition to the extensive simulations, flight tests are also conducted using the sUAS platform described in Section 3. During the flight tests, the actuator attacks are simulated through the autopilot software. The monitoring interval and the watermarking signals used in the flight tests are the same as the ones used in the simulations. However, one difference in implementation between the simulations and the flight tests is in the time interval between two successive actuator attacks. During the flight tests, actuator attacks are simulated every 12 s instead of every 8 s as in the simulations. However, the time duration of the attack signals is retained as 4 s. A longer time interval between two successive actuator attacks is used to ensure that the sUAS does not go out of control during the flight tests. The sUAS is tasked to follow the lemniscate path during the flight tests. The intensities of the attacks considered during the flight tests are the same as those used in the simulations for the struck actuator attack and the scaling attack. But the attack intensities considered for the sinusoidal attack are different: frequencies ranging from 1.0 Hz to 5.0 Hz in steps of 1.0 Hz and amplitudes ranging from 0.1 ms to 0.3 ms in steps of 0.1 ms are simulated. In the flight tests, each combination of the type of attack and the intensity of attack is applied 10 times, and the actuator attacks are implemented one channel at a time. Results from a representative flight test is shown in Fig. 5, where a random actuator attack in the elevator channel is simulated. The figure shows the time evolution of D_E, D_A, D_R, and D_T; the red lines in the figure denote the threshold values, and the black vertical lines denote the time interval of the simulated attack. Since the inequality $D_E > R_E$ is violated during the time interval when the attack is simulated, the detector successfully detects the actuator attack in the elevator channel. A flight segment with only one simulated actuator attack is shown in Fig. 5 to clearly display the time evolution of D_E, D_A, D_R, and D_T. The detection performance of the watermarking-based detector from the flight tests is summarized in Table 1. It is observed that the conclusions drawn from the simulation results also hold for the flight test results.

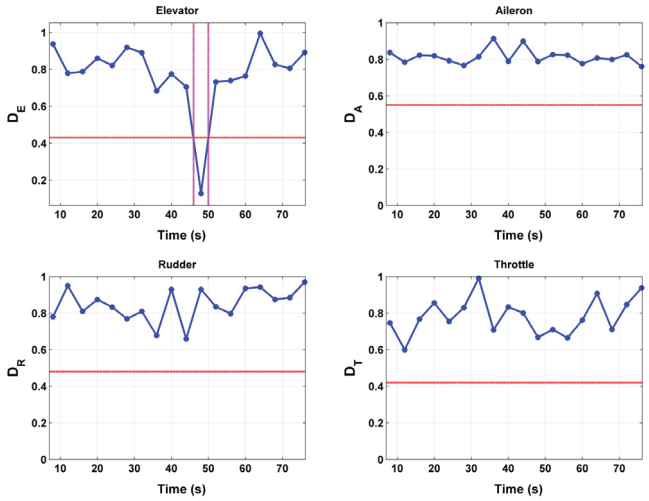


Fig. 5. Results from a representative flight test where a random actuator attack is simulated from $t = 46$ s to $t = 50$ s in the elevator channel.

Merits and shortcomings of the method

One of the important merits of this method is that a system model is not required in order to design the detector. However, if a system model is available, then it can be used to obtain the thresholds R_E, R_A, R_R, and R_T. Otherwise, the thresholds can be obtained from flight tests as explained earlier. Also, the detection performance is not strongly influenced by exogenous disturbances, such as winds, atmospheric turbulence, and sensor noise, as evidenced from the simulation and flight test results. Finally, the detector is easy to implement and does not incur a high computational overhead, which is an important requirement for implementation in an sUAS with limited computational capabilities.

The major shortcoming of this method is that it can only detect the presence of an actuator attack and not the type or magnitude of the attack. Knowledge about the type and magnitude of the attack would be very helpful in developing mitigation strategies through controller reconfiguration, for example. Since the method relies on the frequency response characteristics of the sUAS, this method cannot detect actuator attacks that are constant in magnitude during the monitoring interval. The detection latency of the method is dependent on the value of T_w, a large value for T_w results in a higher detection latency. But reducing the value of T_w would result in a power spectrum with a low resolution and thereby increase the error in the computation of the detection metric, leading to a degraded detection performance. In effect, there is a tradeoff between detection performance and detection latency.

4.2. Estimation-based detection method

The estimation-based detector proposed in this section overcomes some of the limitations of the watermarking-based detector discussed previously. In this method, the actuator attack detection problem is cast as an unknown input estimation problem, where the unknown inputs are the actuator attack signals and the wind velocities. This method makes it possible to explicitly estimate the type and magnitude of the actuator attack, which should be helpful in developing mitigation strategies.

Unknown input estimators for nonlinear systems fall under the following three categories: extended Kalman filter (EKF)-based estimators (Caglayan & Lancraft, 1983; Kim, Lee, & Park, 2009; Lu, Van Eykeren, van Kampen, de Visser, & Chu, 2015), Bayesian estimators (Fang, Srivas, de Callafon, & Haile, 2017), and observers designed based on simplified models using LMI-based techniques (Ha & Trinh, 2004) or nonlinear estimation techniques, such as sliding mode observers (Fridman, Shtessel, Edwards, & Yan, 2008). The method proposed in this section makes use of an unknown input estimator

called the two-stage extended Kalman filter (TSEKF), which falls under the first category of unknown input estimators described before. The recursive nature of TSEKF and its low computational cost (Hsieh & Chen, 1999) make it appealing for real-time implementation on an sUAS. The TSEKF, originally proposed in Caglayan and Lancraft (1983), is an extension of the optimal two-stage Kalman filter (TSKF), which was introduced in Friedland (1969). Although the TSEKF does not guarantee convergence to an optimal solution like the TSKF, it was found to give promising results in solutions to various problems. For instance, the TSEKF and its variants have been used by researchers to estimate unknown biases in an INS–GPS system (Kim et al., 2009), to reconstruct sensor faults in the IMU (Lu, van Kampen, de Visser, & Chu, 2016), and to estimate the speed and rotor flux of induction machines (Hilairet, Auger, & Berthelot, 2009). The novelty of this work is in formulating the actuator attack detection problem as an unknown input estimation problem and using a TSEKF to estimate the actuator attacks from sensor measurements of an sUAS in the presence of exogenous disturbances, such as sensor noise, steady wind, and atmospheric turbulence. The formulation of the unknown input estimation problem is described next.

Problem formulation

The nonlinear model used in the TSEKF is based on the rigid-body equations of motion for a fixed-wing aircraft, available in Stevens et al. (2015) among others. There are three reference frames that are of interest in writing down the equations, namely, the Earth-fixed inertial reference frame denoted by $\mathcal{F}_I$, the body-fixed reference frame denoted by $\mathcal{F}_b$, and the wind reference frame denoted by $\mathcal{F}_w$. The inertial reference frame has its origin on the surface of the Earth and has components $(\boldsymbol{x}_I, \boldsymbol{y}_I, \boldsymbol{z}_I)$, which point to the North, East, and downwards, respectively. The origin of the $\mathcal{F}_b$ frame is affixed to the aircraft center of gravity (CG), and its $(\boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b)$ components point towards the aircraft nose, right wingtip, and downwards, respectively. The wind reference frame has its origin at the aircraft CG and is obtained from the $\mathcal{F}_b$ frame by applying the following rotations: a left-handed rotation about the $\boldsymbol{y}_b$-axis by the angle of attack α, followed by a right-handed rotation about the resulting $\boldsymbol{z}$-axis by the sideslip angle β.

The rotation between the $\mathcal{F}_I$ frame and the $\mathcal{F}_b$ frame is defined by the roll, pitch, and yaw Euler angles, denoted as ϕ, θ, and ψ, respectively. The linear velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_I$ expressed in $\mathcal{F}_b$ is denoted as $\boldsymbol{v} = [u,\ v,\ w]^T$. Likewise, the angular velocity is denoted as $\boldsymbol{\omega} = [p,\ q,\ r]^T$. Denoting the velocity of the wind relative to $\mathcal{F}_I$ as $\boldsymbol{v}_w = [u_w,\ v_w,\ w_w]^T$, the linear velocity of the aircraft relative to the wind can be written as $\bar{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{v}_w$. The airspeed of the aircraft V_a, the angle of attack α, and the angle of sideslip β are defined as

$$V_a = (\bar{\boldsymbol{v}}^T\bar{\boldsymbol{v}})^{\frac{1}{2}}, \quad \alpha = \arctan\frac{w - w_w}{u - u_w}, \quad \text{and} \quad \beta = \arcsin\frac{v - v_w}{V_a}.$$

By virtue of the aircraft's symmetry about the xz-plane of the body-fixed reference frame, the inertia terms I_{xy}, I_{yx}, I_{yz}, and I_{zy} are set to zero. It is also assumed that the other cross-product of inertia I_{xz} is negligibly small. The input vector $\boldsymbol{u}$ is given by $\boldsymbol{u} = [\delta_E,\ \delta_A,\ \delta_R,\ \delta_T]^T$, where δ_E, δ_A, and δ_R are the elevator, aileron, and rudder deflections, respectively, and δ_T is the throttle input. The unknown actuator attack input vector is denoted by $\boldsymbol{u}_a = [\delta_E^a,\ \delta_A^a,\ \delta_R^a,\ \delta_T^a]^T$. F_x, F_y, F_z, M_x, M_y, and M_z denote the components of the net external force and the external moment acting on the aircraft expressed in the body-fixed reference frame. Finally, the wind velocity $\boldsymbol{v}_w$ is decomposed for the purpose of estimation into two components, namely, a steady-wind component denoted by $\bar{\boldsymbol{v}}_w$ and a zero-mean wind component $\boldsymbol{w}$ modeled as white Gaussian noise with covariance defined by $E\{\boldsymbol{w}(t)\boldsymbol{w}^T(t+\tau)\} = \boldsymbol{Q}\delta_\tau$, where $\delta_\tau = 1$ for $\tau = 0$ and $\delta_\tau = 0$ for $\tau \neq 0$.

Having made the necessary definitions, the nonlinear estimation model can be formally written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t), \boldsymbol{b}(t)), \quad \boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{b}(t)) + \boldsymbol{\eta}(t), \tag{3}$$

where $\boldsymbol{\eta}(t)$ denotes the sensor noise and is modeled as a Gaussian white noise process with covariance defined by $E\{\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t+\tau)\} = \boldsymbol{R}\delta_\tau$. The state and measurement vectors are given by $\boldsymbol{x} = \left[\boldsymbol{\omega}^T,\ \boldsymbol{v}^T,\ \phi,\ \theta\right]^T$ and $\boldsymbol{y} = \left[\boldsymbol{\omega}^T,\ \boldsymbol{v}^T,\ \phi,\ \theta,\ V_a,\ a_x,\ a_y,\ a_z\right]^T$, respectively. The unknown input vector $\boldsymbol{b}$ is given by $\boldsymbol{b} = \left[\boldsymbol{u}_a^T,\ \bar{\boldsymbol{v}}_w^T\right]^T$ and is composed of the actuator attack input vector and the steady wind component of the wind vector. The state equation (3) is given by following set of equations:

$$\begin{aligned}
\dot{u}(t) &= -qw + rv + F_x(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})/m - g\sin\theta,\\
\dot{v}(t) &= -ru + pw + F_y(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})/m + g\cos\theta\sin\phi,\\
\dot{w}(t) &= -pv + qu + F_z(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})/m + g\cos\theta\cos\phi,\\
\dot{p}(t) &= (M_x(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) + (I_y - I_z)qr)/I_x,\\
\dot{q}(t) &= (M_y(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) + (I_z - I_x)pr)/I_y,\\
\dot{r}(t) &= (M_z(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) + (I_x - I_y)pq)/I_z,\\
\dot{\phi}(t) &= p + (q\sin\phi + r\cos\phi)\tan\theta,\\
\dot{\theta}(t) &= q\cos\phi - r\sin\phi,
\end{aligned}$$

where m is the aircraft mass and g is the acceleration due to gravity. The net external forces and moments are written in terms of the aerodynamic coefficients and the thrust force as

$$\begin{aligned}
F_x(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_x(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}S + T(V_a, \delta_T, \delta_T^a),\\
F_y(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_y(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}S,\\
F_z(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_z(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}S,\\
M_x(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_l(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}Sb,\\
M_y(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_m(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}S\bar{c},\\
M_z(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b}) &= C_n(\bar{\boldsymbol{v}}, \boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{b})\bar{q}Sb,
\end{aligned}$$

where S, b, and $\bar{c}$ denote the wing area, wingspan, and the mean aerodynamic chord, respectively. $T(V_a, \delta_T, \delta_T^a)$ denotes the thrust force acting on the aircraft and is assumed to act along the $\boldsymbol{x}$-axis of the $\mathcal{F}_b$ frame. $\bar{q}$ denotes the dynamic pressure and is given by $\bar{q} = 0.5\rho V_a^2$, where ρ is the density of air. The aerodynamic model and the thrust model are dependent on the sUAS, and in this work, the aerodynamic coefficients are assumed to have the following model structure, which is same as in Muniraj et al. (2017):

$$\begin{aligned}
C_x &= C_{x_0} + C_{x_\alpha}\alpha + C_{x_{\delta_T}}\bar{\delta}_T + C_{x_T}T(V_a, \delta_T, \delta_T^a)/(\bar{q}S),\\
C_y &= C_{y_0} + C_{y_\beta}\beta + C_{y_{\delta_A}}\bar{\delta}_A + C_{y_{\delta_R}}\bar{\delta}_R + (C_{y_p}p + C_{y_r}r)b/(2V_a),\\
C_z &= C_{z_0} + C_{z_\alpha}\alpha + C_{z_{\delta_E}}\bar{\delta}_E + C_{z_q}q\bar{c}/(2V_a) + C_{z_T}2T/(\rho SV_a^2),\\
C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_{\delta_A}}\bar{\delta}_A + C_{l_{\delta_R}}\bar{\delta}_R + (C_{l_p}p + C_{l_r}r)b/(2V_a),\\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_E}}\bar{\delta}_E + C_{m_q}q\bar{c}/(2V_a),\\
C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_{\delta_A}}\bar{\delta}_A + C_{n_{\delta_R}}\bar{\delta}_R + (C_{n_p}p + C_{n_r}r)b/(2V_a),
\end{aligned} \tag{4}$$

where $\bar{\delta}_E = \delta_E + \delta_E^a$, $\bar{\delta}_A = \delta_A + \delta_A^a$, $\bar{\delta}_R = \delta_R + \delta_R^a$, and $\bar{\delta}_T = \delta_T + \delta_T^a$.

The TSEKF is used to estimate the state $\boldsymbol{x}$ and the unknown input vector $\boldsymbol{b}$ using the system model described in Eqs. (3) to (4). Although the state and measurement equations are provided in continuous-time, the TSEKF itself is implemented in discrete-time. The unknown input vector is modeled in the TSEKF as $\boldsymbol{b}_{k+1} = \boldsymbol{b}_k + \boldsymbol{w}_k^{\boldsymbol{b}}$, where $\boldsymbol{b}_{k+1} = \boldsymbol{b}(t_{k+1})$ and $t_{k+1} = k\Delta t$, with Δt being the sampling time. $\boldsymbol{w}_k^{\boldsymbol{b}}$ is a zero-mean Gaussian white noise process with the covariance defined as $E\{\boldsymbol{w}_k^{\boldsymbol{b}}(\boldsymbol{w}_{k+\tau}^{\boldsymbol{b}})^T\} = \boldsymbol{Q}^{\boldsymbol{b}}\delta_\tau$. It is also assumed that the unknown actuator attack inputs and the wind disturbances are independent processes, which is a reasonable assumption to make since it is very difficult for the attacker to accurately measure atmospheric turbulence and modify the attack inputs accordingly. The TSEKF consists of two components, namely, the *bias-free filter*, which estimates the state vector assuming that the unknown input vector is zero, and a *bias filter*, which estimates the unknown input vector. The final state estimate is a combination of the estimates from the bias-free filter and the bias filter. Due to space considerations, the TSEKF algorithm is not provided here; the interested reader is referred to Caglayan and Lancraft (1983) for the details of the algorithm.

Table 2
Thrust coefficients used in the polynomial thrust model.

Term	Value	Term	Value	Term	Value	Term	Value
p_{00}	-1.0835×10^{4}	p_{10}	-1.4884×10^{1}	p_{01}	3.6875×10^{1}	p_{20}	-5.3886×10^{-2}
p_{11}	4.6225×10^{-2}	p_{02}	-4.9518×10^{-2}	p_{30}	-4.9386×10^{-3}	p_{21}	4.1153×10^{-4}
p_{12}	-5.3590×10^{-5}	p_{03}	3.2746×10^{-5}	p_{40}	-8.3496×10^{-6}	p_{31}	6.9799×10^{-6}
p_{22}	-4.7319×10^{-7}	p_{13}	2.7065×10^{-8}	p_{04}	-1.0653×10^{-8}	p_{50}	-5.2178×10^{-7}
p_{41}	2.9825×10^{-8}	p_{32}	-2.5547×10^{-9}	p_{23}	1.3625×10^{-10}	p_{14}	-4.9870×10^{-12}
p_{05}	1.3646×10^{-12}						

An actuator attack in a particular channel is detected if the estimation of the unknown actuator attack input corresponding to that channel exceeds a threshold. The threshold values are used to minimize the number of false positives, which can occur when the TSEKF estimates a non-zero attack input vector of small magnitude due to the presence of exogenous disturbances, even though the sUAS is not under any actuator attack. Further details on how to choose the threshold values will be given in the next subsection.

Simulations and flight experiments

The simulation setup explained in the previous section and used to study the effectiveness of the active detection method is also used here. The values for the inertia matrix of the aircraft are the same as those in Muniraj et al. (2017). The values for the aerodynamic coefficients in the nonlinear estimation model are the same as the ones used for the simulation model. However, a different thrust model is used for estimation. In the simulations, a table-lookup thrust model is used with the airspeed and the PWM throttle command as the inputs. Since a table-lookup model is computationally expensive and not suitable for real-time implementation, a polynomial thrust model that approximates the table-lookup thrust model is used instead in the estimation. Specifically, a fifth-order polynomial in two variables is used and is given by

$$\begin{aligned} T(V_a, \delta_T) = {} & p_{00} + p_{10} V_a + p_{01} \delta_T + p_{20} V_a^2 + p_{11} V_a \delta_T + p_{02} {\delta_T}^2 \\ & + p_{30} V_a^3 + p_{21} V_a^2 \delta_T + p_{12} V_a {\delta_T}^2 + \\ & p_{03} {\delta_T}^3 + p_{40} V_a^4 + p_{31} V_a^3 \delta_T + p_{22} V_a^2 {\delta_T}^2 + p_{13} V_a {\delta_T}^3 \\ & + p_{04} {\delta_T}^4 + p_{50} V_a^5 + \\ & p_{41} V_a^4 \delta_T + p_{32} V_a^3 {\delta_T}^2 + p_{23} V_a^2 {\delta_T}^3 + p_{14} V_a {\delta_T}^4 + p_{05} {\delta_T}^5, \end{aligned}$$

where the coefficients are provided in Table 2.

The following values for the covariance matrices are considered:

$$\begin{aligned} \boldsymbol{Q} &= \mathrm{diag}(1 \times 10^{-2}, 1 \times 10^{-2}, 1 \times 10^{-2}), \\ \boldsymbol{Q}^{\boldsymbol{b}} &= \mathrm{diag}(9 \times 10^{-6}, 4 \times 10^{-6} \boldsymbol{I}_2, 2.5 \times 10^{-7}, 9 \times 10^{-6} \boldsymbol{I}_3), \\ \boldsymbol{R} &= \mathrm{diag}(7.6 \times 10^{-5} \boldsymbol{I}_3, 1.0, 1.0 \times 10^{-2} \boldsymbol{I}_2, 9.9 \times 10^{-5} \boldsymbol{I}_2, 4.0, 9 \times 10^{-2} \boldsymbol{I}_3), \end{aligned}$$

where $\mathrm{diag}(M_1, \ldots, M_e)$ denotes the block diagonal augmentation of matrices $M_1, \ldots, M_e$. The covariance matrices are initially chosen based on the expected noise characteristics, which can be typically obtained from flight test data. Then, these matrices are fine-tuned during filter design to result in smaller estimation errors.

The threshold values used in the detector are chosen based on simulations where the sUAS is not subjected to any actuator attack; however, the sUAS is subjected to the same form of exogenous disturbances as given in Section 4.1. Data from these simulations are used in the estimation-based detector to estimate the unknown actuator attack input vector. Although the true actuator attack input vector is zero, the estimate of this vector need not be zero due to the presence of disturbances. Based on the distributions of the estimation errors thus obtained, the threshold value for each channel is chosen as 110% of the maximum absolute value of the estimation error for that channel. This inflated value compensates for the potentially nonzero estimate of the attack input vector in attack-free situations. The threshold values are selected to be symmetric about zero and are given as follows: 0.04 ms, 0.045 ms, 0.035 ms, and 0.035 ms for the elevator, aileron, rudder, and throttle channels, respectively.

Table 3
Performance of the estimation-based detector from simulations and flight tests.

Control channel	False positive rate (%)		
	Simulation case 1	Simulation case 2	Flight tests
Elevator	0.84	13.17	9.12
Aileron	1.38	18.23	13.51
Rudder	1.29	17.96	15.32
Throttle	0.43	7.09	5.56

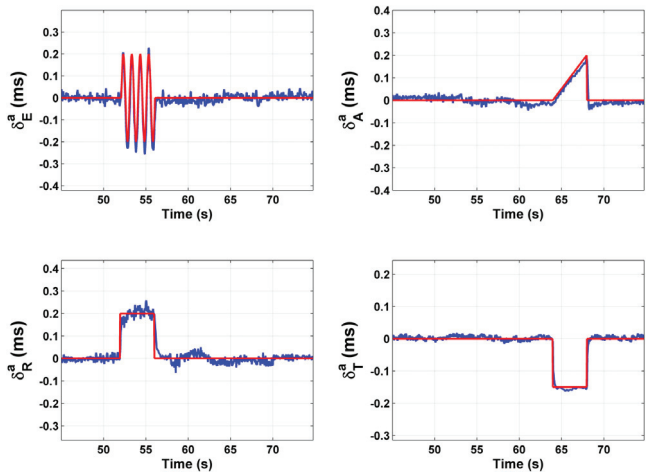


Fig. 6. A representative simulation comparing the estimated actuator attack input (shown in blue) with the actual actuator attack input (shown in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

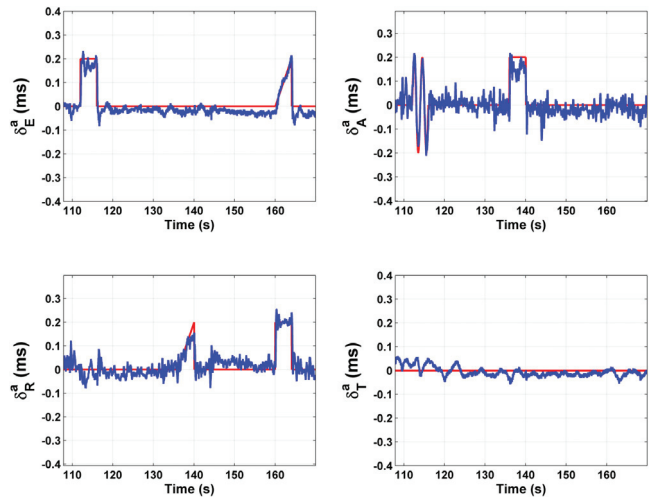


Fig. 7. A representative flight test segment comparing the estimated actuator attack input (shown in blue) with the actual actuator attack input (shown in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Simulations are performed where the sUAS is subjected to the same types and intensities of attacks, same paths, and same disturbances as the ones used in the simulations for the watermarking-based detector. The duration of the attack signals and the manner in which they are implemented are also the same as described before. In addition, a fifth type of attack is considered in this case, a bias injection attack, where the magnitude of the actuator attack signal remains constant throughout the attack duration. During the simulations, six different values are considered for the magnitude of the bias injection attack, namely -0.3 ms, -0.2 ms, -0.1 ms, 0.1 ms, 0.2 ms, and 0.3 ms. Since the detector makes use of the aerodynamic model of the sUAS, it is important to assess the performance of the detector under conditions where the aerodynamic model contains uncertainties. To this end, two different simulation cases are considered in this study. In case 1, the aerodynamic model used in the estimation is the same as the aerodynamic model used in the simulations. In case 2, the aerodynamic model used in the estimation is the same as in case 1; however, the aerodynamic model used in the simulations has uncertainties in the aerodynamic coefficients. Specifically, the aerodynamic coefficients $C_i(k)$ at time instant t_k are replaced by $C_i(k) + \delta C_i(k)$ for $i = x, y, z, l, m, n$, where $\delta_{C_i}^- \leq \delta C_i(k) \leq \delta_{C_i}^+$ and $\nu_{C_i}^- \leq \delta C_i(k+1) - \delta C_i(k) \leq \nu_{C_i}^+$. $\delta C_i(k)$ is a pseudorandom sequence with specified bounds, where the values of these bounds are the same as those given in Table III of Palframan, Fry, and Farhood (2017). As far as the flight test results are concerned, each combination of attack type and attack intensity is simulated 10 times. During all the flight tests, the sUAS is tasked to follow the race track path.

The simulation and flight test results are summarized in Table 3, where the FPR of the detector for each of the four control channels is provided. In all the simulations and flight tests, the detector is able to detect all the actuator attacks, thereby resulting in a TPR of 100%. However, the false positive rates differ for the flight tests and the two simulation cases. Not surprisingly, when the aerodynamic model contains uncertainties (simulation case 2), the FPR is higher for all the four control channels. The values of the FPR from the flight tests lie in between the values for simulation case 1 and those for simulation case 2, thereby indicating that the results obtained based on the simulation environment under case 2 constitute a worst-case evaluation of the detector's performance. Figs. 6 and 7 show representative segments from simulations and flight tests, where a comparison is made between the output of the TSEKF and the actual unknown actuator attack input.

It was observed from the simulation studies that for all the attacks, the estimation errors lie within ± 0.2 ms. The estimation errors for the struck actuator attack, the scaling attack, and the bias attack are concentrated within ± 0.1 ms. As for the sinusoidal actuator attack and the random actuator attack, the estimation errors are more uniformly distributed within the interval $[-0.2,\ 0.2]$. One of the reasons for the increased estimation errors for the sinusoidal and random actuator attacks is found to be the phase lag that exists between the estimated actuator attack input and the actual attack input.

Merits and shortcomings of the method

The main advantage of the estimation-based detection method is that it enables estimation of the magnitude of the actuator attack, which is not possible using the watermarking-based detector. Also, the estimation-based detector is able to detect different types of actuator attacks, including the bias injection attack. The estimation-based detector has a lower detection latency compared to the watermarking-based detector because of the absence of a monitoring interval in the estimation-based detection. The detection latency for the estimation-based detector depends on the threshold values: a higher threshold value, which might result in a lower FPR, could come at the expense of a higher detection latency.

In order to guarantee lower false positive rates and better estimation performance, the estimation-based detector requires an accurate model of the sUAS. However, obtaining an exact model of the sUAS is a formidable task, and invariably any model used will contain inaccuracies that will impact the performance of the detector. The estimation-based detector is also computationally more expensive than the watermarking-based detector, as its implementation requires carrying out operations such as matrix inversion.

5. Resilient hardware approach

The previous two methods discussed in Section 4 do not require any changes to the servos and are purely algorithmic. Although the two methods are shown to detect different types of actuator attacks, a separate mitigation strategy is needed to ensure the safe operation of the sUAS under actuator attacks. Hence, each of the previous two methods would serve as one part of a two-pronged strategy, involving attack detection and then mitigation. The approach proposed in this section obviates the need for the two-pronged approach and makes the servo resilient to actuator attacks through hardware modifications.

As explained in Section 2, the typical servos used in an sUAS are controlled using constant-frequency PWM signals. To design an effective actuator attack and maximize its impact, the attacker needs to know about the pulse frequency of the PWM signals, and so the attacker will initially attempt to determine this frequency. By using a constant-frequency PWM signal to control the servo, a fixed target is provided to the attacker, which makes it easier for the attacker to discern the frequency. The servo can therefore be made resilient to the actuator attacks by just increasing the randomness in the servo's operation, which can be achieved by continuously changing the targeted pulse frequency pseudorandomly. This strategy is akin to moving target defense, which is used in network security to secure computer networks against malicious attacks (Jajodia, Ghosh, Subrahmanian, Swarup, Wang, & Wang, 2012). The moving target defense forms the core of the resilient hardware approach. The proposed servo is described in detail in the following subsection.

Proposed resilient actuator system

The proposed servo will be referred to as the resilient actuator system. The resilient actuator system is composed of a microcontroller and the six components of the servo described in Section 2, namely, the DC motor, the control horn, the gear reduction system, the potentiometer, the servo plug, and the built-in microprocessor. The resilient actuator system does not require modifications to any of the six components of the servo. The only hardware modification required is the addition of a microcontroller. Instead of constant-frequency PWM signals, the resilient actuator system uses variable-frequency PWM signals. The variable-frequency PWM signals are generated by the autopilot and decoded by the microcontroller, which then transforms the variable-frequency PWM signals into constant-frequency PWM signals before sending them to the built-in microprocessor of the servo. The frequencies of the variable-frequency PWM signal are chosen pseudorandomly from a set of frequencies denoted by $\boldsymbol{F}$. The elements of $\boldsymbol{F}$ are selected from the interval $[f_l,\ f_u]$, where $f_l \geq F_s$ and $f_u < 500$ Hz. F_s is the sampling frequency of the control signal, and f_u is limited by the maximum pulse width of a PWM signal, which is 2 ms. Each element in $\boldsymbol{F}$ is assigned a positive integer from the interval $[1,\ |\boldsymbol{F}|]$, where $|\boldsymbol{F}|$ denotes the cardinality of $\boldsymbol{F}$. A pseudorandom number generator is used to generate pseudorandom binary sequences, and by combining a sufficient number of bits, a pseudorandom integer sequence is generated. The Blum–Blum–Shub (BBS) pseudorandom number generator, also known as the $x^2 \bmod N$ pseudorandom number generator, is used in the resilient actuator system because of its non-predictability property (Blum, Blum, & Shub, 1986). The inputs to the BBS pseudorandom number generator are N and x_0, where $N = pq$ is a product of two distinct prime numbers p and q, each of which is congruent to $3 \bmod 4$, and x_0 is an integer co-prime to N. Given N and x_0, one can generate the pseudorandom sequence forward in time; however, due to the non-predictability property of the BBS pseudorandom number generator, the

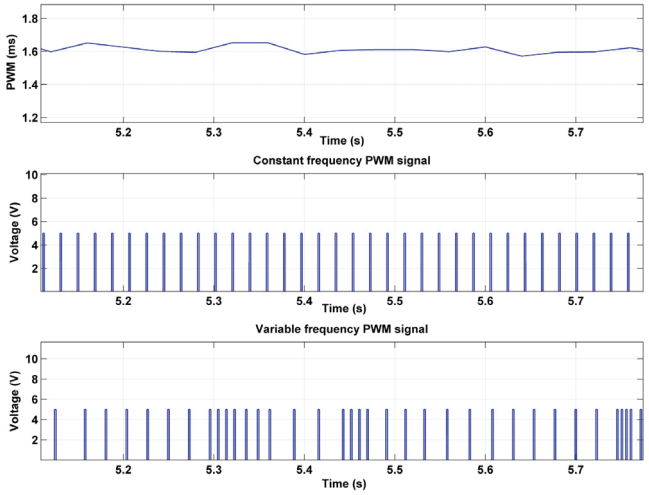


Fig. 8. A representative segment of the control signal along with the corresponding constant-frequency and variable-frequency PWM signals.

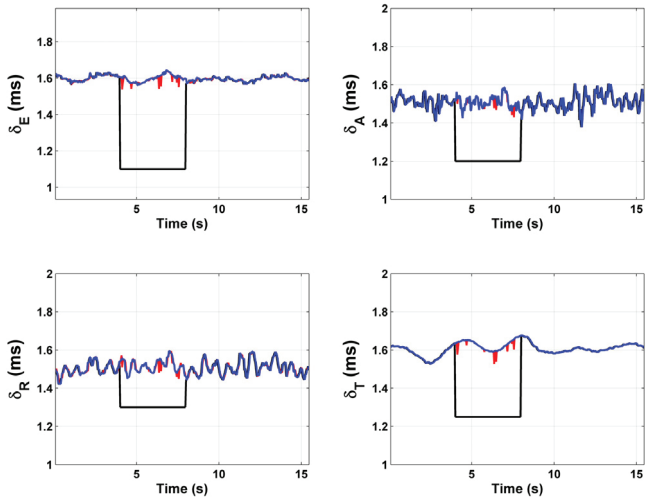


Fig. 9. A representative MATLAB simulation of the resilient actuator system under struck actuator attacks; the control signals sent by the autopilot are shown in blue, the control signals read by the servos are shown in red, and the control signals that would have been read by the servos in the absence of variable-frequency PWM signals are shown in black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

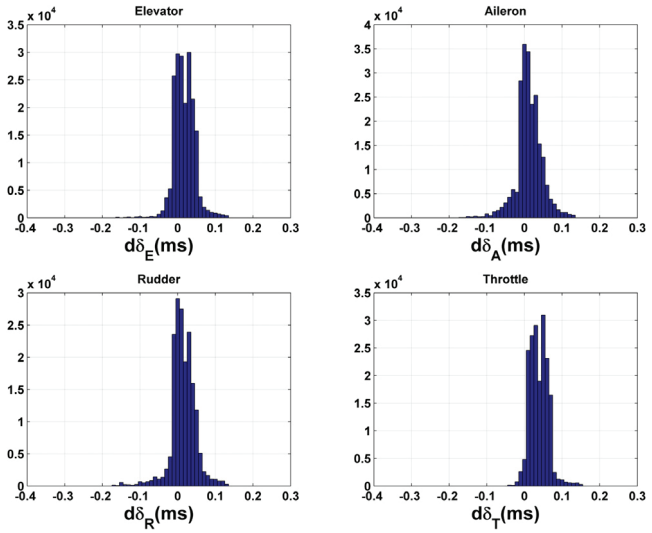


Fig. 10. Histograms of the differences between the control signals sent by the autopilot and the control signals read by the microcontroller.

sequence cannot be generated backwards. Since the frequency sequence cannot be predicted without knowing x_0 and N, any malicious adversary will not be able to inflict significant damage on the sUAS due to an actuator attack.

During system initialization, the autopilot and the microcontroller are time-synchronized, and the inputs to the pseudorandom number generator, x_0 and N, are shared between the autopilot and the microcontroller. Depending on the pseudorandom integer sequence, the autopilot generates the variable-frequency PWM control signals by changing the pulse frequency. The microcontroller on the receiving end generates the same pseudorandom integer sequence, and since the autopilot and the microcontroller are time-synchronized, the variable-frequency PWM signals can be decoded by the microcontroller. The variable-frequency PWM signal is transformed into a constant-frequency PWM signal by the microcontroller before sending it to the servo's built-in microprocessor. Fig. 8 shows a segment of the control signal sent by the autopilot, along with the corresponding constant-frequency and variable-frequency PWM signals.

Although the autopilot can demand rapid changes in the control commands, the actual control commands sent to the control surfaces are determined by the servo dynamics. The servo acts as a low-pass filter by attenuating high frequency inputs. This information is used in the resilient actuator system to limit the rate of change of actuator commands even before they are sent to the servo. Based on the characteristics of the servo used in this work, the maximum rate of change of control commands is limited to 0.625 ms/s.

The resilient actuator system creates a moving target through the use of variable-frequency PWM signals. Depending on the type of attack and the attack signal's frequency, there could be time instances where the control signal sent by the autopilot is slightly altered by the actuator attack even with the resilient actuator system in place. But the intended effect of the actuator attack would not be achieved unless the attacker is able to predict the frequency sequence used in the variable-frequency PWM signal. An appropriately designed robust sUAS controller should be able to mitigate the effects of such small random changes in the control commands without any significant performance degradation. A detailed analysis is presented next to corroborate this point.

Analysis results

A software implementation of the resilient actuator system is implemented in MATLAB, and simulations are performed. Different actuator attacks are simulated, and the PWM signal read by the microcontroller is compared with the PWM signal sent by the autopilot. The values of N and x_0 used in the simulations are $N = 501826793$ and $x_0 = 351278754$, and the set $\boldsymbol{F}$ is given by $\boldsymbol{F} = \{F_1, F_2, \ldots, F_{15}\}$, where $F_i = 10^3/(2i+3)$ for $i = 1, \ldots, 15$, with the frequencies F_i given in Hertz. In the absence of actuator attacks, the PWM signal decoded by the microcontroller will be the same as the PWM signal sent by the autopilot. Since the resilient actuator system is independent of the sUAS platform, only the race track path is used in all the simulations. The five different types of actuator attacks described in the previous sections are also considered here. As before, the duration of each of the attack signals is 4 s, and each combination of attack type and attack intensity is applied 1000 times. In the previous simulations, it was sufficient to simulate the intended effect of the attack, that is, the PWM values corresponding to the control commands were simply altered to reflect the intended effect of the attack. However, the simulations in this case are supposed to demonstrate the efficacy of the resilient actuator system in weeding out the intrusive PWM signals through the frequency-hopping mechanism. To do so, the exact implementation of the attack must be simulated.

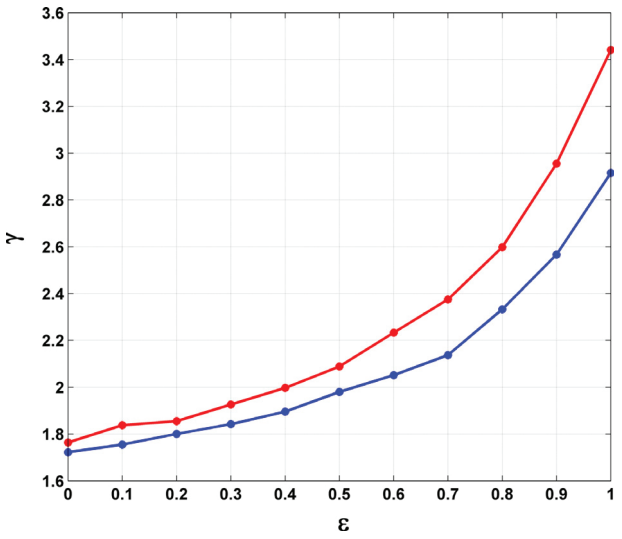


Fig. 11. Upper bounds on the robust $\mathcal{W}$-to-ℓ_2-gains obtained from IQC analysis; the red curve corresponds to the case where $d\delta_i$ are incorporated in the analysis; the blue curve corresponds to the case where $d\delta_i$ are not accounted for in the analysis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

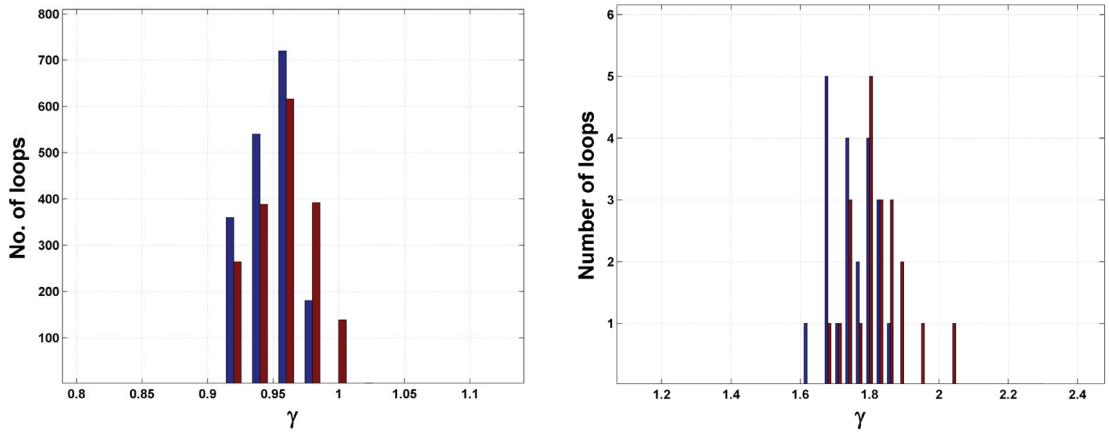
The attack mechanism described in Section 2.2 is used to implement the actuator attacks, as outlined next. First, the attack signal is transformed into a trailing-edge modulated PWM signal with a pulse frequency of 50 Hz, which is the most common frequency used for PWM signals. Suppose that the rising and falling edges of the PWM attack signal are denoted by $t_r^a(j)$ and $t_f^a(j)$, respectively, for positive integers j, and that the rising and falling edges of the PWM control signal sent by the autopilot are denoted by $t_r(j)$ and $t_f(j)$, respectively. The pulse width of the jth pulse of the PWM attack signal is then given by $(t_f^a(j)-t_r^a(j))$ ms. If there is no actuator attack, the jth pulse of the PWM control signal has a pulse width of $(t_f(j)-t_r(j))$ ms. In the presence of an actuator attack, the jth pulse of the PWM control signal has a pulse width equal to

$$\min\left((t_f(j)-t_r(j)), \max\left((t_f^a(j)-t_r(j)), 1.1\right)\right) \text{ ms}$$

if $t_f^a(j)$ lies within the time interval of the attack, where the value 1.1 in the previous expression is the minimum pulse width of the PWM signal recognizable by the servo. If $t_f^a(j)$ is outside the time interval of an attack, then the jth pulse of the PWM control signal has a pulse width of $(t_f(j)-t_r(j))$ ms. Thus, the attacker can only decrease the PWM value of the control signal.

Fig. 9 shows a representative simulation, where struck actuator attacks are simulated in each of the four control channels. Specifically, struck actuator attacks with magnitudes of 1.1 ms, 1.2 ms, 1.3 ms, and 1.25 ms are applied in the elevator, aileron, rudder, and throttle channels, respectively. The duration of each of the attack signals is 4 s. It is observed from the figure that the resilient actuator system is able to mitigate the actuator attacks in each of the four channels and that the difference in the PWM values between the control signal sent by the autopilot and the control signal read by the microcontroller is small. For all the four channels, the absolute value of the maximum difference between the two signals in this particular case is less than 0.07 ms. The metric used to assess the performance of the resilient actuator system during the simulations is the difference between the control signal sent by the autopilot and the control signal read by the microcontroller, which for a channel i at time t is denoted by $d\delta_i(t)$, where $i = E, A, R, T$ corresponding to the elevator, aileron, rudder, and throttle channels, respectively. The sign convention for $d\delta_{(\cdot)}(\cdot)$ is as follows: if the PWM value of the control signal sent by the autopilot is greater than the PWM value of the control signal read by the microcontroller, then $d\delta_{(\cdot)}(\cdot)$ is positive. Since the attack mechanism is such that the attacker can only reduce the PWM value of the control signal, $d\delta_{(\cdot)}(\cdot) \geq 0$ during the time interval of an attack. If an actuator attack is applied to channel i and the resilient actuator system is able to completely mitigate this attack, then $d\delta_i(t)\equiv 0$ during the time interval of the attack. For each of the four channels, the values of $d\delta_i(t)$ are computed from all the simulations. Fig. 10 shows the distributions of $d\delta_i$ for $i = E, A, R, T$. Firstly, it is observed from the figure that $d\delta_i$ is negative in some cases. The reason for the occurrence of the negative values of $d\delta_i$ is that the control commands demanded by the autopilot in certain cases are rate-limited by the resilient actuator system. Secondly, as expected, the resilient actuator system does not completely mitigate the actuator attacks; however, the differences between the control signal PWM values sent by the autopilot and the PWM values read by the microcontroller are small and lie within ± 0.1 ms. Depending on the attacker's frequency and the type of attack, the control commands may be randomly altered by a small value due to the actuator attack. A properly designed sUAS controller should be able to mitigate the effects of $d\delta_i$. In the next paragraph, tools from robust control theory are used to show that the effects of $d\delta_i$ on the performance of the sUAS are not significant. While it is possible for the attacker to employ variable-frequency PWM signals instead of constant-frequency PWM signals as assumed in this study, it is unlikely for this scenario to happen in actual operations because of the limited energy available to the attacker. To induce variable-frequency voltage signals using the attack mechanism described in Section 2.2, a larger bandwidth and hence more energy are required. Nevertheless, preliminary results show that the variable-frequency PWM attack signals are actually less effective against the resilient actuator system than the constant-frequency ones.

Integral quadratic constraint (IQC) theory provides a rigorous approach to systematically examine the robust stability and performance of controlled systems (Megretski & Rantzer, 1997). The IQC analysis approach uses linear fractional transformations (LFTs) on uncertainties to express systems and can handle a wide range of uncertainties, including static and dynamic, time-invariant and time-varying perturbations, sector-bounded nonlinearities, and time-varying delays. In IQC analysis, the uncertain plant and the controller are modeled as an upper LFT of a causal, stable nominal system, denoted by $\boldsymbol{M}$, and a perturbation operator, denoted by $\Delta \in \boldsymbol{\Delta}$. This LFT $(\boldsymbol{M}, \Delta)$ maps the disturbance input $\boldsymbol{w} \in \mathcal{W} \subseteq \ell_2$ to the performance output $\bar{\boldsymbol{z}}$, where ℓ_2 denotes the normed space of square summable vector-valued sequences. $(\boldsymbol{M}, \Delta)$ is said to have a robust $\mathcal{W}$-to-ℓ_2-gain performance level of γ if it is robustly stable and $\|(\boldsymbol{M}, \Delta)\|_{\mathcal{W}\to\ell_2} < \gamma$ for all $\Delta \in \boldsymbol{\Delta}$. Recently, a mathematical tool for IQC analysis of fixed-wing UAS controllers has been developed (Palframan et al., 2017), where unmodeled dynamics, nonlinearities, aerodynamic uncertainties, delays, and actuator saturation are characterized using different IQC multipliers. This tool will be utilized in this work to assess analytically the effect of the attack-induced control errors $d\delta_i$ on the worst-case performance of the sUAS $\mathcal{H}_\infty$ path-following controller. For space considerations, the following demonstrates only how the attack-induced control errors are incorporated into the uncertain UAS framework. The reader is referred to Palframan et al. (2017) for the specifics of the framework, including the various uncertainties used to characterize the uncertain operational environment of UAS flight controllers. Each $d\delta_i$, for $i = E, A, R, T$, is modeled as an exogenous input signal, which has a constant power spectral density within the frequency range $[-\omega_0, \omega_0]$. From the simulation results, the power spectral density of each $d\delta_i$ is computed, and it is found that, for all the input channels, most of the energy of the control error signal is concentrated within the frequency band $[-0.32\pi, 0.32\pi]$. Therefore, ω_0 is set equal to 0.32π. The performance output is chosen as $\bar{\boldsymbol{z}} = [\bar{X}, \bar{Y}, \bar{H}]^T$, where $\bar{X}$, $\bar{Y}$, and $\bar{H}$ denote the position errors of the sUAS with respect to the desired position on the reference path; see Muniraj et al. (2017). Fig. 11 shows the upper bounds on the robust $\mathcal{W}$-to-ℓ_2-gain performance level of the uncertain LFT system $(\boldsymbol{M}, \epsilon\Delta)$ for different values of the uncertainty scale factor ϵ, as obtained from IQC analysis. For the sake of



(a) Simulation results. (b) Flight test results.

Fig. 12. Distributions of the $\mathcal{W}$-to-ℓ_2-gains from simulations and flight tests; the red bars correspond to the case where $d\delta_i$ are incorporated; the blue bars correspond to the case where $d\delta_i$ are not included. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

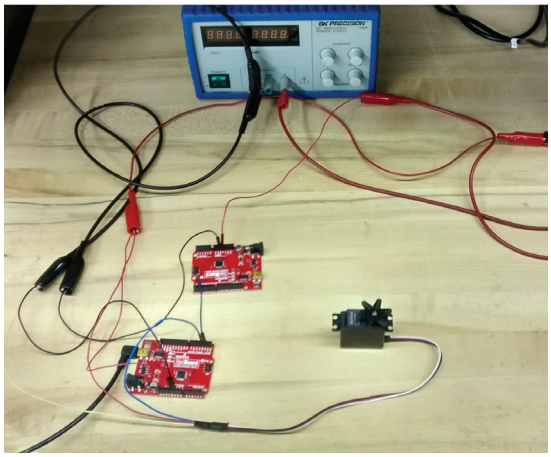


Fig. 13. A prototype of the resilient actuator system.

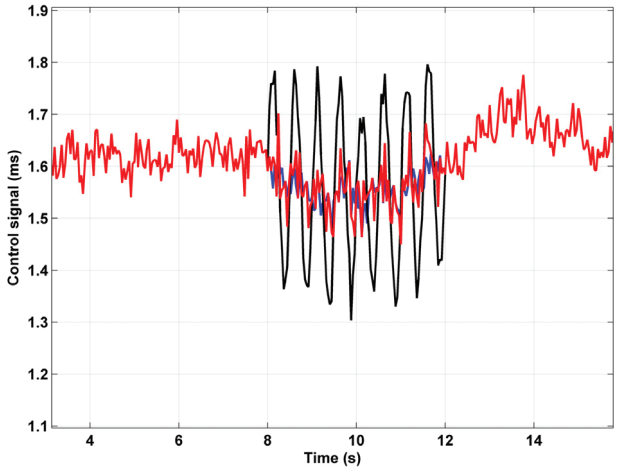


Fig. 14. Results from a representative test using the prototype resilient actuator system; the blue curve represents the control signal sent by the autopilot, the black curve indicates the attacked control signal, and the red curve indicates the control signal read by the servo. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

comparison, IQC analysis results are also shown for the case where the exogenous signals that model $d\delta_i$ are not included in the analysis. The maximum degradation in γ due to the effect of $d\delta_i$ is 17% and occurs when the scaling factor $\epsilon = 1$.

Since IQC analysis provides only an upper bound on the $\mathcal{W}$-to-ℓ_2-gain performance level, the percentage degradation in γ of 17% due to $d\delta_i$ might be a conservative estimate. Thus, in addition to IQC analysis, nonlinear simulations and flight tests are conducted, where exogenous inputs are added to each of the four control channels. In the simulations and flight tests, the sUAS is controlled by the $\mathcal{H}_\infty$ path-following controller and is tasked to follow the race track path. The exogenous inputs are pseudorandomly generated from a uniform distribution over the interval $[-\omega_0, \omega_0]$. The $\mathcal{W}$-to-ℓ_2-gain is computed for each loop using information about the wind disturbances, sensor noise, attack-induced control errors, and position errors. The distributions of the $\mathcal{W}$-to-ℓ_2-gains thus computed from simulations and flight tests are shown in Fig. 12. The results presented correspond to 1800 cycles of the reference path for the simulations and 21 cycles of the reference path for the flight tests. The figure shows histograms for both the case where $d\delta_i$ are incorporated and the case where $d\delta_i$ are not included. The percentage degradation in the maximum value of γ due to the effect of $d\delta_i$ is 5% from the simulations and 9% from the flight tests, both of which are less than the value predicted from IQC analysis. Thus, our analysis indicates that actuator attacks do not have a significant effect on the control performance of sUAS equipped with the resilient actuator system.

Table 4
Comparison of the three methods.

	Watermarking-based detector	Estimation-based detector	Resilient actuator system
System model	Not needed	Needed	Not needed
Computational overhead	Low	High	Low
Hardware modification	Not needed	Not needed	Needed[a]
Attack mitigation	Not included[b]	Not included[b]	Included
Detection latency	Varies[c]	Low	Not applicable (no separate detection)

[a] Requires an additional microcontroller.
[b] Requires a separate mitigation strategy.
[c] Depends on the monitoring interval (trade-off between detection performance and detection latency).

A prototype of the resilient actuator system that uses variable-frequency PWM signals is shown in Fig. 13. The objective of building the prototype is to demonstrate the proof-of-concept of the proposed resilient actuator system. The prototype consists of two Sparkfun RedBoards representing the autopilot and the microcontroller, a servo, a

5 V power supply, and connecting wires. The two RedBoards are programmed using the Arduino IDE, and they communicate with each other through pulse-width modulated signals. For the sake of demonstration, the RedBoard representing the autopilot is programmed to simulate different types of actuator attacks, and the control signals read by the RedBoard representing the microcontroller are recorded. Fig. 14 shows a representative test where a sinusoidal actuator attack of amplitude 0.2 ms and frequency 2 Hz is simulated. It is observed from the figure that the prototype resilient actuator system is able to mitigate the effects of the actuator attack.

Merits and shortcomings of the approach

The main advantage of the resilient hardware approach is that it does not require a separate mitigation strategy, as the approach ensures resilience against actuator attacks by design. The resilient hardware approach only requires an additional microcontroller to interpret the variable-frequency PWM signals and hence is easy to implement on servos currently used in sUAS. A minor shortcoming of the method is that the actuator attacks are not completely mitigated since there is a marginal difference between the control signals applied to the servos and the control signals sent by the autopilot during an actuator attack. However, the discrepancies between the two control signals do not have a significant effect on the performance of the sUAS, as evidenced from IQC analysis, nonlinear simulations, and flight tests.

6. Conclusions and future work

This work presented three different methods to address the problem of detection and mitigation of actuator attacks on sUAS. The watermarking-based detector and the estimation-based detector make use of the knowledge about the sUAS to detect actuator attacks. While the watermarking-based detector only detects the existence of an actuator attack, the estimation-based detector can also estimate the unknown actuator attack inputs. The resilient hardware approach explicitly addresses the security vulnerabilities of the servos through a hardware modification. Since the resilient hardware approach by design ensures the safe operation of sUAS under actuator attacks, it does not require a separate mitigation strategy like the other two methods. All of the three methods are evaluated using extensive nonlinear simulations in MATLAB and flight tests on a fixed-wing sUAS. The merits and shortcomings of the methods are also discussed from the standpoint of implementation on an actual sUAS. Table 4 gives a comparison of the important characteristics of the three methods. To the best of the authors' knowledge, this is the first work that investigates the problem of detection and mitigation of actuator attacks for sUAS.

Some possible areas of future work include building compact augmented servos with appropriately programmed microcontrollers and conducting flight tests. Another area of future work is to subject the sUAS to real actuator attacks in flight tests, such as the ones described in Selvaraj et al. (2018), and evaluate the effectiveness of the methods.

Acknowledgments

This work was funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation (NSF) sponsored industry/university cooperative research center (I/UCRC) under NSF Grant Nos. IIP-1539975 and CNS-1650465 along with significant contributions from C-UAS industry members. The authors would like to thank Micah Fry for providing the MATLAB code for IQC analysis and for his assistance during the flight tests.

References

Bateman, F., Noura, H., & Ouladsine, M. (2011). Fault diagnosis and fault-tolerant control strategy for the aerosonde uav. *IEEE Transactions on Aerospace and Electronic Systems*, *47*(3), 2119–2137.

Birnbaum, Z., Dolgikh, A., Skormin, V., OBrien, E., Muller, D., & Stracquodaine, C. (2016). Unmanned aerial vehicle security using recursive parameter estimation. *Journal of Intelligent and Robotic Systems*, *84*(1–4), 107–120.

Blum, L., Blum, M., & Shub, M. (1986). A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing*, *15*(2), 364–383.

Boyd, S. (1986). Multitone signals with low crest factor. *IEEE Transactions on Circuits and Systems*, *33*(10), 1018–1022.

Caglayan, A. K., & Lancraft, R. E. (1983). A separated bias identification and state estimation algorithm for nonlinear systems. *Automatica*, *19*(5), 561–570.

Colomina, I., & Molina, P. (2014). Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, *92*, 79–97.

Cox, I., Miller, M. L., & Bloom, J. A. (2002). *Digital watermarking*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Ducard, G., & Geering, H. P. (2008). Efficient nonlinear actuator fault detection and isolation system for unmanned aerial vehicles. *Journal of Guidance Control and Dynamics*, *31*(1), 225–237.

Fang, H., Srivas, T., de Callafon, R. A., & Haile, M. A. (2017). Ensemble-based simultaneous input and state estimation for nonlinear dynamic systems with application to wildfire data assimilation. *Control Engineering Practice*, *63*, 104–115.

Fawzi, H., Tabuada, P., & Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, *59*(6), 1454–1467.

Federal Aviation Administration. (2017). FAA Aerospace Forecast (Fiscal Years 2017-2037).

Fridman, L., Shtessel, Y., Edwards, C., & Yan, X. -G. (2008). Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems. *International Journal of Robust and Nonlinear Control*, *18*(4–5), 399–412.

Friedland, B. (1969). Treatment of bias in recursive filtering. *IEEE Transactions on Automatic Control*, *14*(4), 359–367.

Gage, S. (2003). Creating a unified graphical wind turbulence model from multiple specifications. In *AIAA modeling and simulation technologies conference and exhibit*, Austin, Texas.

Gumstix. (2017). Gumstix Overo Fire. URL https://store.gumstix.com/coms/overo-coms/overo-firestorm-y-com.html.

Ha, Q. P., & Trinh, H. (2004). State and input simultaneous estimation for a class of nonlinear systems. *Automatica*, *40*(10), 1779–1785.

Heredia, G., Ollero, A., Bejar, M., & Mahtani, R. (2008). Sensor and actuator fault detection in small autonomous helicopters. *Mechatronics*, *18*(2), 90–99.

Hilairet, M., Auger, F., & Berthelot, E. (2009). Speed and rotor flux estimation of induction machines using a two-stage extended Kalman filter. *Automatica*, *45*(8), 1819–1827.

Hoareau, G., Liebenberg, J. J., Musial, J. G., & Whitman, T. R. (2017). Package transport by unmanned aerial vehicles, US Patent 9, 731, 821.

Hobby Express, (2018). Senior Telemaster Plus. URL http://hobbyexpress.com/senior-telemaster-laser-cut-kit/.

Hsieh, C. -S., & Chen, F. -C. (1999). Optimal solution of the two-stage Kalman estimator. *IEEE Transactions on Automatic Control*, *44*(1), 194–199.

Jajodia, S., Ghosh, A. K., Subrahmanian, V., Swarup, V., Wang, C., & Wang, X. S. (2012). *Moving target defense II: Application of game theory and adversarial modeling, Vol. 100*. Springer.

Kerns, A. J., Shepard, D. P., Bhatti, J. A., & Humphreys, T. E. (2014). Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, *31*(4), 617–636.

Kim, K. H., Lee, J. G., & Park, C. G. (2009). Adaptive two-stage extended Kalman filter for a fault-tolerant INS-GPS loosely coupled system. *IEEE Transactions on Aerospace and Electronic Systems*, *45*(1), 125–137.

Kim, A., Wampler, B., Goppert, J., & Hwang, I. (2012). Cyber attack vulnerabilities analysis for unmanned aerial vehicles. In *Proc. AIAA Infotech @ Aerospace 2012*, Garden Grove, California.

Lu, P., van Kampen, E. -J., de Visser, C., & Chu, Q. (2016). Nonlinear aircraft sensor fault reconstruction in the presence of disturbances validated by real flight data. *Control Engineering Practice*, *49*, 112–128.

Lu, P., Van Eykeren, L., van Kampen, E. -J., de Visser, C., & Chu, Q. (2015). Double-model adaptive fault detection and diagnosis applied to real flight data. *Control Engineering Practice*, *36*, 39–57.

Megretski, A., & Rantzer, A. (1997). System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, *42*(6), 819–830.

Merino, L., Caballero, F., Martínez-de Dios, J. R., Maza, I., & Ollero, A. (2012). An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent and Robotic Systems*, *65*(1), 533–548.

Mitchell, R., & Chen, I. (2014). Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(5), 593–604.

Montambault, S., Beaudry, J., Toussaint, K., & Pouliot, N. (2010). On the application of VTOL UAVs to the inspection of power utility assets. In *1st international conference on applied robotics for the power industry*.

Morelli, E. A. (2003). Multiple input design for real-time parameter estimation in the frequency domain. *IFAC Proceedings Volumes*, *36*(16), 639–644.

Mulero-Pázmány, M., Negro, J. J., & Ferrer, M. (2013). A low cost way for assessing bird risk hazards in power lines: Fixed-wing small unmanned aircraft systems. *Journal of Unmanned Vehicle Systems*, *2*(1), 5–15.

Muniraj, D., & Farhood, M. (2017). A framework for detection of sensor attacks on small unmanned aircraft systems. In *Proceedings of the international conference on unmanned aircraft systems* (pp. 1189–1198).

Muniraj, D., Palframan, M. C., Guthrie, K. T., & Farhood, M. (2017). Path-following control of small fixed-wing unmanned aircraft systems with $\mathcal{H}_\infty$ type performance. *Control Engineering Practice*, *67*, 76–91.

Palframan, M. C., Fry, J. M., & Farhood, M. (2017). Robustness analysis of flight controllers for fixed-wing unmanned aircraft systems using integral quadratic constraints. *IEEE Transactions on Control Systems Technology*, http://dx.doi.org/10.1109/TCST.2017.2766598.

Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, *58*(11), 2715–2729.

Pixhawk. (2018). Pixhawk Autopilot hardware. URL https://docs.px4.io/en/flight_controller/pixhawk.html.

Rugh, W. J. (1981). *Nonlinear system theory*. Johns Hopkins University Press Baltimore.

Selvaraj, J., Dayanıklı, G. Y., Gaunkar, N. P., Ware, D., Gerdes, R. M., & Mina, M. (2018). Electromagnetic induction attacks against embedded systems. In *Proceedings of the 2018 Asia conference on computer and communications security (ASIACCS '18)* (pp. 499–510). New York, NY, USA: ACM.

Son, Y., Shin, H., Kim, D., Park, Y. -S., Noh, J., Choi, K., et al. (2015). Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX security symposium* (pp. 881–896).

Stevens, B. L., Lewis, F. L., & Johnson, E. N. (2015). *Aircraft control and simulation: Dynamics, controls design, and autonomous systems*. John Wiley & Sons.

Stoica, P., & Moses, R. L. (2005). *Spectral analysis of signals* (1st ed.). Prentice Hall.

Sun, J. (2012). Pulse-width modulation. In *Dynamics and control of switched electronic systems* (pp. 25–61). Springer.

Teixeira, A., Shames, I., Sandberg, H., & Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, *51*, 135–148.

Watts, A. C., Perry, J. H., Smith, S. E., Burgess, M. A., Wilkinson, B. E., Szantoi, Z., et al. (2010). Small unmanned aircraft systems for low-altitude aerial surveys. *Journal of Wildlife Management*, *74*(7), 1614–1619.

Wu, X., Lang, Z. Q., & Billings, S. (2007). Analysis of output frequencies of nonlinear systems. *IEEE Transactions on Signal Processing*, *55*(7), 3239–3246.

Yoon, M. -K., Liu, B., Hovakimyan, N., & Sha, L. (2017). Virtualdrone: virtual sensing, actuation, and communication for attack-resilient unmanned aerial systems. In *Proceedings of the 8th international conference on cyber-physical systems* (pp. 143–154). ACM.

Yu, X., & Jiang, J. (2015). A survey of fault-tolerant controllers based on safety-related issues. *Annual Reviews in Control*, *39*, 46–57.

Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, *32*(2), 229–252.

Zhang, C., & Kovacs, J. M. (2012). The application of small unmanned aerial systems for precision agriculture: A review. *Precision Agriculture*, *13*(6), 693–712.