2017 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 13-16, 2017, Miami, FL, USA

A Framework for Detection of Sensor Attacks on Small Unmanned Aircraft Systems

Devaprakash Muniraj and Mazen Farhood

***Abstract*— The work presented in this paper is part of an overall effort to design a secure autopilot, resilient against malicious attacks on both the cyber and physical layers, for a small unmanned aircraft system (UAS). This paper specifically deals with identification of malicious attacks on the sensors of a small UAS. A framework is presented wherein techniques from statistical analysis are used in a probabilistic setting to detect sensor attacks. The paper describes in detail the design of anomaly detectors and the Bayesian network. A case study involving detection of a spoofing attack on the GPS is used throughout the paper to illustrate the proposed approach. The anomaly detectors are designed based on a simulation dataset, and are re-tuned based on flight tests conducted on a small fixed-wing UAS platform. The performances of the detectors are studied under different external disturbances and conclusions are drawn.**

I. INTRODUCTION

Unmanned aircraft systems (UAS) are increasingly becoming ubiquitous in both civilian and defense applications. Now that plans for the integration of UAS into the national airspace are underway [1], it is becoming more and more obvious that one of the critical barriers to this integration is ensuring the safety and security of these systems. The security concerns are even more acute in civilian applications, where transmissions are unencrypted and the UAS architecture is widely known [2]. There is a transition in the nature of security threats to UAS from passive confidentiality breaches, such as eavesdropping, to active integrity breaches, such as jamming and spoofing. The attackers are becoming increasingly smart and employ sophisticated mechanisms to compromise the UAS stealthily. In such a scenario, it is essential to develop an autopilot system capable of actively detecting and mitigating malicious cyber-physical attacks.

UAS are highly coupled nonlinear systems with an operational environment characterized by uncertainties and disturbances like sensor noise, wind gusts, and atmospheric turbulence. The control design process for a UAS begins with developing a mathematical model of the physical system using techniques from system identification [3]. The mathematical model is only an approximation of the physical system as many assumptions and simplifications are typically made to obtain a tractable model. The resulting nonlinear model is further simplified, for instance, by linearization, to obtain a plant model that is amenable to control design. The modeling inaccuracies and neglected nonlinearities, along with atmospheric disturbances and sensor noise, provide a hotbed for attackers to masquerade their attacks. Figure 1 shows the modeling inaccuracies and disturbances in a typical UAS that can be exploited by an adversary. One of the main challenges in detecting malicious attacks on UAS is to be able to distinguish between the response of the UAS to usual disturbances and the response due to malicious attacks.

Cyber-security has been an active research area over the last two decades as evident from the numerous papers and review articles published in this area [4]–[6]. Research on the security of cyber-physical systems (CPS) is fairly recent, however, and the publications in this area stem from diverse application domains. The existing research on CPS security can be categorized in many ways, for instance, based on the field of application, type of threats addressed, approach used (theoretical versus heuristic), etc. One such classification is to categorize the works into ones which address the problem of external intrusion detection using algorithmic methods [7]–[20] and works which use hardware-enhanced security methods to identify and mitigate internal security threats [21]–[24]. The works which address the problem of cyber-physical security for UAS predominantly fall under the category of algorithmic methods, and specifically behavior-specification-based methods, wherein normal system behavior is formally

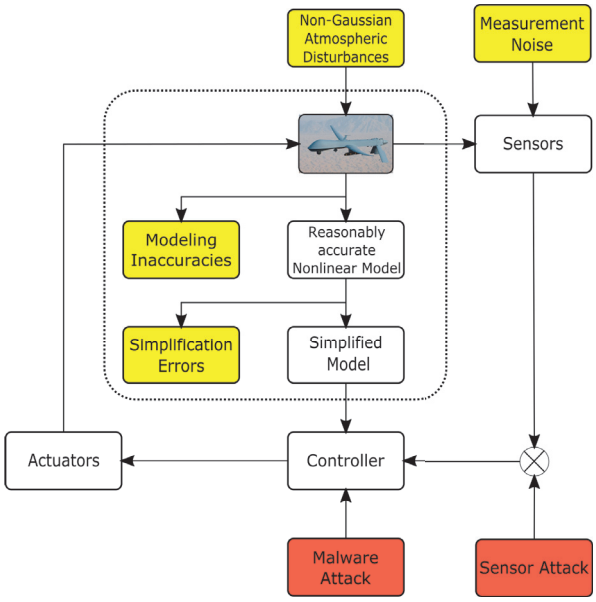


Fig. 1: Modeling inaccuracies and disturbances in a typical UAS that can be exploited by an attacker.

The authors are with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061, USA. Email: (devapm@vt.edu, farhood@vt.edu)

This work is funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation sponsored industry/university cooperative research center (I/UCRC) under NSF Award No. IIP-1161036 along with significant contributions from C-UAS industry members.

978-1-5090-4494-8/17/$31.00 ©2017 IEEE

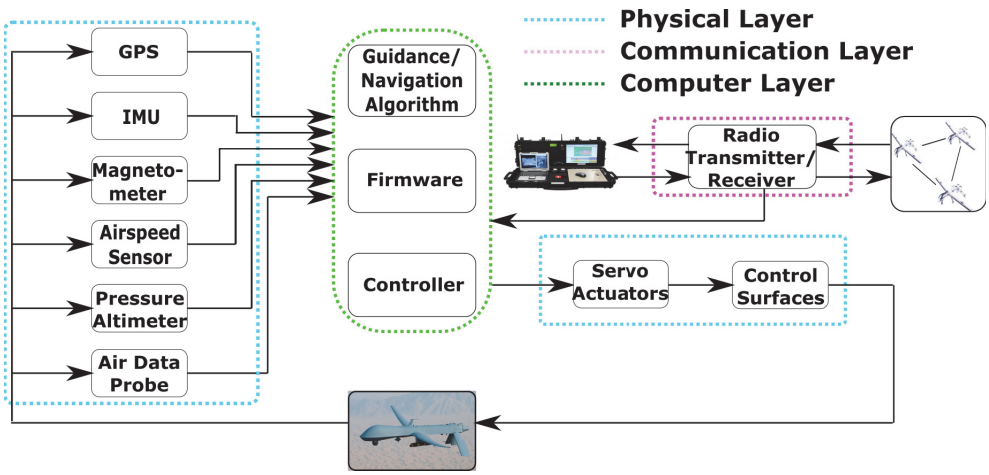


Fig. 2: Diagram showing different layers in a typical unmanned aircraft system.

defined using temporal logic or finite state machines and deviations from the normal behavior are identified as intrusions. In [12], normal system operation is defined in terms of behavioral rules, which specify a certain number of states, some safe and others unsafe, and the behavior specification is translated into a state machine. The authors in [15] define normal system behavior using linear and metric temporal logic formulas in a Bayesian network framework.

Most of the existing literature on cyber-physical security for UAS do not explicitly consider the UAS model and the resultant uncertainties, thereby increasing the probability of triggering false alarms. In this paper, we propose a framework for identification of cyber-physical attacks on the sensors of a small UAS (sUAS). By incorporating knowledge about the physical system and using a probabilistic framework, the proposed approach minimizes the false alarm rates by concentrating on the response of the UAS to attacks and not on the attack mechanism itself.

Throughout this paper, we consider a case study that concerns detection of a spoofing attack on the GPS to illustrate the proposed methods. The outline of the paper is as follows. In section II, the proposed framework is presented. Section III describes the design of different anomaly detectors and compares their performances in terms of the false alarm rate and detection latency. Section IV describes the flight tests, where spoofing attacks on the GPS are simulated, and the retuning of the anomaly detectors based on the data gathered from the flight tests. In Section V, the Bayesian network for attack detection is described and some operational scenarios are simulated using a Bayesian inference tool called Hugin-Lite. Finally, in Section VI, conclusions and some topics of future work are discussed.

II. FRAMEWORK FOR DETECTION OF SENSOR ATTACKS

The security threats encountered by a UAS may target one or more of the following three layers [25]: *physical layer* consisting of the sensors, actuators, ground control station, and communication hardware, *computer layer* consisting of the controller software, firmware, guidance and navigation algorithms, and *communication layer* consisting of the radio-frequency links to the ground control station and inter-UAS communication in the case of multiple UAS; see Figure 2. In this paper, the problem of detecting cyber-physical attacks on the physical layer and especially on the sensors is addressed. This work is part of an overall effort to design a secure autopilot, resilient against malicious attacks on all the three layers. The framework consists of a layered approach, where *attack indicators* at each layer are identified and then used as evidences in a Bayesian network to detect an attack.

We assume that only a subset of the sensors is compromised, and our approach aims to identify these compromised sensors. Specifically, the sensors in a typical UAS can be classified as *safe sensors* or *vulnerable sensors*. Safe sensors are sensors that are intrinsic to the UAS, in the sense that, they do not interact with an outside system. Such sensors are highly unlikely to be compromised by an attacker and some examples include the inertial measurement unit (IMU) (assuming it is reasonably protected against electromagnetic interference), airdata probe, airspeed sensor, and pressure altimeter. Vulnerable sensors, on the other hand, are sensors that rely on an external agent for sensing and are highly susceptible to adversarial attacks. Examples of such sensors include GPS, RADAR, LIDAR, and vision-based sensors. The communication links used for sending/receiving data in a UAS network, which are susceptible to attacks such as false-data injection and jamming, are categorized as vulnerable sensors. In this work, it is assumed that only the vulnerable sensors can be compromised. Although it is assumed that the safe sensors cannot be compromised, they can provide faulty measurements under atypical operational conditions, such as severe atmospheric disturbances or transient sensor faults.

The attack detection approach consists of the following two types of anomaly detection methods:

1) Anomaly detection using *attack signatures*, which are based on measurements from the safe sensors, and
2) Anomaly detection based on *residuals*, which are computed by a residual generator from the outputs of a state estimator.

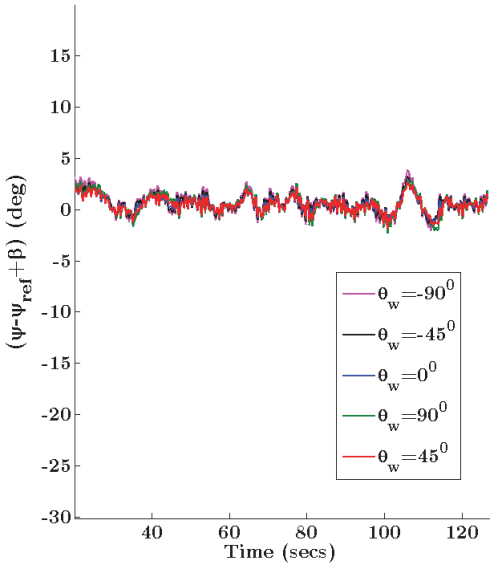
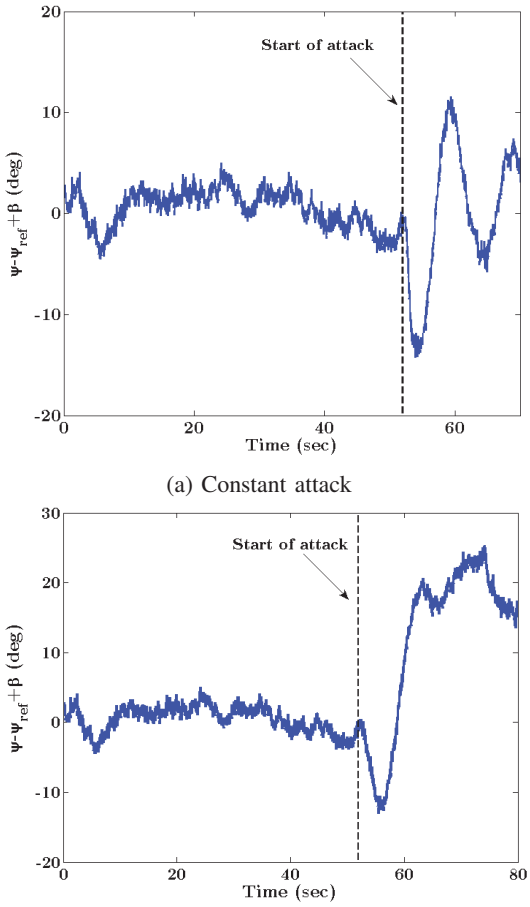


Fig. 3: Effect of wind orientation on $(\psi - \psi_{ref} + \beta)$.

A. Anomaly Detection Based on Attack Signatures

Attack signatures, which encode information about a possible attack on the physical layer of the UAS, are identified based on the knowledge of the physical system. It is important to note that the attack signatures are based only on data from safe sensors. Attack signatures correspond to abnormal behavior in the time evolution of certain measurements or combinations of measurements during an attack on the physical layer. The foremost requirement of an attack signature is that it should be sensitive to malicious attacks on the UAS while being hardly sensitive to changes in wind disturbances and measurement noise. Based on simulation studies performed using an sUAS model, it is observed that for threats such as spoofing attack on GPS and replay attack, the terms $(\psi - \psi_{ref} + \beta)$ and $(\theta - \theta_{ref} - \alpha)$ satisfy the requirements for an attack signature. The symbols ψ, θ, α, and β denote the yaw angle, pitch angle, angle of attack, and angle of sideslip, respectively. ψ_{ref} and θ_{ref} are the yaw reference angle and pitch reference angle, respectively; they are provided by the motion planner or computed from the reference path generated by the motion planner. For instance, ψ_{ref} is the angle between the local tangent to the reference path at the current position and the North axis of the NED frame. $(\psi - \psi_{ref} + \beta)$ can be thought of as the deviation of the velocity vector projected onto the local horizontal frame from the local tangent to the reference path.

The effectiveness of $(\psi - \psi_{ref} + \beta)$ as an attack signature is shown in Figures 3 and 4. Figure 3 shows the variation of $(\psi - \psi_{ref} + \beta)$ for different orientations of the wind vector, θ_w, during closed-loop simulations performed using a Senior Telemaster UAS model [26]. A path-following controller composed of an inner-loop Proportional-Integral-Derivative controller and an outer-loop nonlinear guidance logic for way-point following is used in the simulations. During the simulations, the UAS is tasked to follow a straight line path in the presence of 4 m/s steady wind, medium level Dryden



(a) Constant attack

(b) Scaling attack

Fig. 4: Variation of $(\psi - \psi_{ref} + \beta)$ during a spoofing attack on GPS.

turbulence [27], and measurement noise. It is observed from the figure that $(\psi - \psi_{ref} + \beta)$ is hardly sensitive to changes in the wind disturbance. Figure 4 shows the variation of $(\psi - \psi_{ref} + \beta)$ during two different types of spoofing attack on the GPS latitude measurement, called the constant attack and the scaling attack. Both attacks introduce a bias into the GPS latitude measurement. The constant attack adds a constant bias, whereas, in the scaling attack, the added bias increases linearly with time. It is observed from Figure 4 that in the absence of any attack, $(\psi - \psi_{ref} + \beta)$ varies within ± 5 degrees mainly due to the effects of atmospheric turbulence and sensor noise, but in the presence of the attack, $(\psi - \psi_{ref} + \beta)$ increases in magnitude significantly, thereby serving as an indicator to detect attacks that are hidden under the guise of disturbances.

B. Anomaly Detection Based on Residuals

The second type of anomaly detection method uses residuals, which are computed from the measurements of the vulnerable sensors and the output of a state estimator. Each vulnerable sensor is associated with a state estimator, which provides estimates of the true uncompromised measurements made by the vulnerable sensor. The state estimator uses mea-

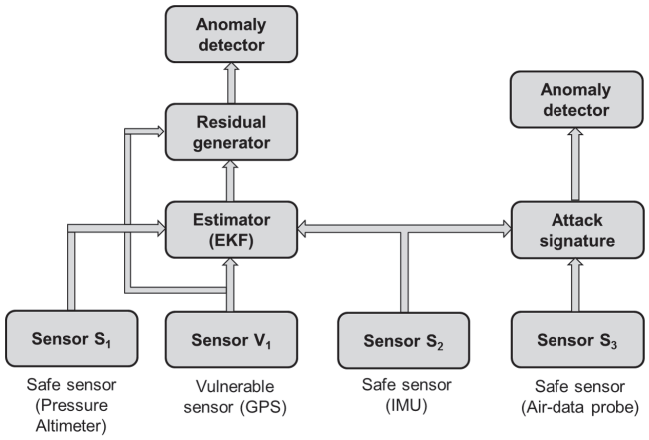


Fig. 5: Detection of spoofing attacks on the GPS using the attack detection framework.

surements from the safe sensors and the vulnerable sensor in computing the estimates. The output of the state estimator and the measurements from the vulnerable sensor are used in a residual generator to compute the residuals. The residuals from the residual generator are then used by the anomaly detector to detect a sensor attack. In the present framework, an Extended Kalman Filter (EKF) is used as the state estimator. In order to study the comparative performance, different types of anomaly detectors and residual generators are considered, and their performances are compared in terms of the detection latency and the false alarm rate. Three types of anomaly detectors are considered, two of them are based on statistical parametric methods such as the cumulative sum (CUSUM) and sequential probability ratio test (SPRT), and the third anomaly detector uses a non-parametric method based on binary hypothesis testing.

To keep the illustration of the anomaly detection method simple, we consider a case study where the vulnerable sensor is the GPS, and the safe sensors are the IMU and the pressure altimeter. Figure 5 shows the detection framework for the case study considered. The EKF estimates the UAS position and body-axis velocities using the accelerations a_x, a_y, and a_z, the body-axis angular rates p, q, and r, and the attitude angles ϕ, θ, and ψ. The dynamic system considered is assumed to have zero-mean, uncorrelated, Gaussian process and measurement noise, and is given in state-space form as

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) + w(t), \\ y(t) &= h(x(t), u(t)), \\ z(t_k) &= y(t_k) + v(t_k), \end{aligned} \tag{1}$$

where $x(t)$ is the state vector, $y(t)$ is the measurement output in continuous-time, and $z(t_k)$ is the discrete-time measurement output at time $t = t_k$. The process noise and the measurement noise are given by $w(t)$ and $v(t_k)$, respectively. The state vector is composed of the UAS positions in the NED frame x_N, x_E, and x_D, the UAS body-axis velocities u_b, v_b, and w_b, the accelerometer biases b_{a_x}, b_{a_y}, and b_{a_z}, and the gyro biases b_p, b_q, and b_r. The measurement vector consists of the UAS position x_N, x_E, and altitude H, where $H = -x_D$ is obtained from the pressure altimeter. The state equations are given by

$$\begin{aligned} \dot{u}_b &= (a_x - b_{a_x}) - (q - b_q)w_b + (r - b_r)v_b - gs\theta, \\ \dot{v}_b &= (a_y - b_{a_y}) - (r - b_r)u_b + (p - b_p)w_b - gc\theta s\phi, \\ \dot{w}_b &= (a_z - b_{a_z}) - (p - b_p)v_b + (q - b_q)u_b - gc\theta c\phi, \\ \dot{x}_N &= u_b c\theta c\psi + v_b(s\phi s\theta c\psi - c\phi s\psi) \\ &\quad + w_b(c\phi s\theta c\psi + s\phi s\psi), \\ \dot{x}_E &= u_b c\theta s\psi + v_b(s\phi s\theta s\psi + c\phi c\psi) \\ &\quad + w_b(c\phi s\theta s\psi - s\phi c\psi), \\ \dot{x}_D &= -u_b s\theta + v_b c\theta s\phi + w_b c\theta c\phi, \\ \dot{b}_{a_x} &= 0, \quad \dot{b}_{a_y} = 0, \quad \dot{b}_{a_z} = 0, \\ \dot{b}_p &= 0, \quad \dot{b}_q = 0, \quad \dot{b}_r = 0, \end{aligned} \tag{2}$$

where the sin and cos terms are abbreviated as s and c, respectively. The input vector is composed of the accelerometer outputs a_x, a_y, and a_z, the body-axis angular rates p, q, and r, and the Euler angles ϕ, θ, and ψ. The theory on EKF is widely discussed in the literature and is available in [28], [29] among others, and due to paucity of space is not discussed here. The measurement update occurs every 0.1 seconds. The input noise covariance matrix, Q, and the measurement noise covariance matrix, R, are chosen as

$$Q = \mathrm{diag}(2I_3, 1{\times}10^{-3}, 1{\times}10^{-3}, 5{\times}10^{-4}, 2{\times}10^{-3}I_3),$$

and $R = \mathrm{diag}(2, 2, 0.5)$, where diag denotes a diagonal matrix and I_3 is the identity matrix of size 3×3. The initial guess for the process noise covariance matrix is chosen as

$$P(0) = \mathrm{diag}(0.5I_3, 2, 2, 0.5, 2I_3, 1I_3).$$

The UAS position estimated by the EKF and the measurement from the GPS are used to compute the residual in the residual generator. Let $\epsilon(t_k) = z(t_k) - \hat{y}(t_k)$ denote the difference between the GPS measurement and the EKF estimate at time t_k. Two types of residual generators are considered in this work, the first residual generator is based on the 1-norm of $\epsilon(t_k)$, and the second residual generator is based on the χ^2 statistic of $\epsilon(t_k)$. Let $S(t_k) = R + C(t_k)^T P(t_k) C(t_k)$, where R is the measurement noise covariance matrix, and $P(t_k)$ and $C(t_k)$ are the prediction error covariance matrix and the observation matrix at time t_k, respectively. It is noted that $P(t_k)$ and $C(t_k)$ are obtained from the EKF. Given $S(t_k)$, the χ^2-residual is given by $s(t_k) = \epsilon(t_k)^T S(t_k)^{-1} \epsilon(t_k)$. In the absence of any attack, s has a χ^2-distribution. The change in the distribution of s during an attack is used in the anomaly detector to detect the attack. The design of the different anomaly detectors and their comparative detection performance are discussed in the next section.

III. DESIGN AND ANALYSIS OF ANOMALY DETECTORS

As mentioned earlier, three different types of anomaly detectors are considered, namely the sequential probability ratio test (SPRT) detector, cumulative sum (CUSUM) detector and the binary hypothesis testing (BHT) detector.

A. SPRT Anomaly Detector

In the SPRT anomaly detector, the test statistic $g(t_k)$ is computed and checked against a threshold h at every time instant [30]. If $g(t_k)$ exceeds the threshold, then the alarm is set. The algorithm used in the SPRT anomaly detector is given below:

$$g(t_k) = g(t_{k-1}) + s(t_k) - \nu,$$
$$g(t_k) = 0 \quad \text{if} \quad g(t_k) < a,$$
$$\text{ALARM} = \begin{cases} 1 & \text{if} \quad g(t_k) > h, \\ 0 & \text{otherwise}, \end{cases}$$

where $s(t_k)$ is the output of the residual generator. The parameters of the detector are the drift term ν, the reset value a, and the threshold h. The drift term prevents positive drifts due to noise in the sensor measurements, which could result in a false alarm. The reset value resets the test statistic to zero to prevent a negative drift which could increase the detection latency. The design of the detector involves choosing the three parameters such that the false alarm rate and the detection latency are minimum. The design procedure is detailed in the forthcoming paragraphs.

B. CUSUM Anomaly Detector

The algorithm used in the CUSUM anomaly detector is similar to that used in the SPRT detector. In fact, the CUSUM detector is a special case of the SPRT detector, where the reset value a is zero. The reset value, ideally, should be as small as possible in magnitude, as a higher value for a would result in an increase in detection time. A value of zero for a is the minimum possible. The CUSUM detector, therefore, has only two parameters: the drift term ν and the threshold h.

The design parameters of the SPRT and CUSUM detectors are obtained by solving an optimization problem, where the sum of the false positive and false negative rates is minimized based on a simulation dataset. The simulation dataset is generated from a number of nonlinear six-degree-of freedom simulations, where spoofing attacks on the GPS latitude and longitude measurements are simulated. The simulations are performed using a mathematical model of a small fixed-wing UAS platform derived based on flight test data [31]. During the simulations, the UAS is tasked to follow a reference path, which is enabled by a path-following controller designed based on the controller structure available in the open-source Ardupilot software [32]. The following factors are varied during the simulations:

1) *Reference path* - Two types of reference path, namely a straight line path and a circular path of radius $110.5\,\text{m}$, are considered.
2) *Type of attack* - Two types of GPS spoofing attack, namely the constant attack and the scaling attack, are simulated. In a constant attack, a constant bias is added to the GPS latitude and longitude measurements at every time instant. Whereas, in a scaling attack, a time-varying bias that scales linearly with time is added to the GPS latitude and longitude measurements.

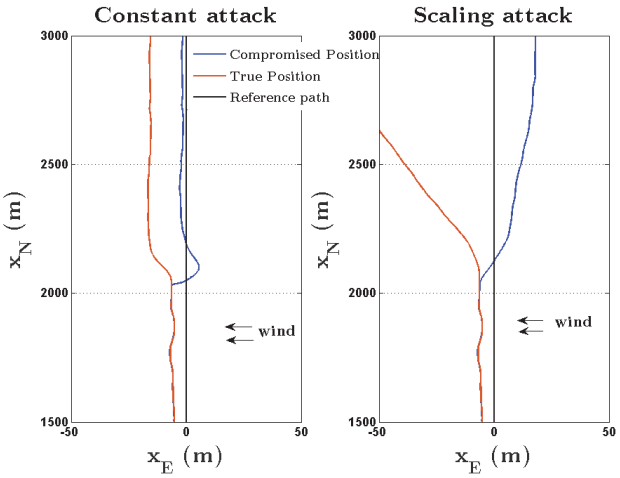


Fig. 6: Representative simulations of a constant attack of magnitude 15 m and a scaling attack of magnitude 1.25 m/s on the x_E position measurement.

3) *Magnitude of attack* - Four different attack magnitudes are considered for each of the two types of GPS attack. The attack magnitudes considered for the constant attack are $3\,\text{m}$, $5\,\text{m}$, $10\,\text{m}$ and $15\,\text{m}$, and the attack magnitudes for the scaling type of attack are $0.3125\,\text{m/s}$, $0.625\,\text{m/s}$, $1.25\,\text{m/s}$, and $2.5\,\text{m/s}$. It is noted that the attack is injected after the GPS measurements are converted to position measurements in the NED frame.
4) *Wind disturbance* - A steady wind of magnitude $4\,\text{m/s}$ is considered during the simulations, and the direction of wind is varied from $0°$ to $360°$ in steps of $10°$ or $45°$. It is noted that the direction of wind is changed only for the simulations where the reference path is a straight line.
5) *Direction of attack* - In addition to varying the magnitude of attack, the direction of attack is also varied from $0°$ to $360°$ in steps of $10°$ or $45°$. For instance, when the direction of attack is $0°$, only the x_N position measurement is modified during the attack, and for an attack direction of $90°$, only the x_E position measurement is modified.

The above cases result in a simulation dataset composed of 3264 different simulation cases with 2720 hours of simulation time. Two representative simulation cases for a straight line reference path are shown in Figure 6. The simulation dataset is used to choose the optimal parameter values for the SPRT and the CUSUM detectors. The resulting parameter values for the SPRT detector are $\nu = 0.578$, $h = 3.146$, and $a = -3.373$. The parameter values for the CUSUM detector are $\nu = 0.570$ and $h = 3.663$.

C. Binary Hypothesis Testing (BHT) Anomaly Detector

The third type of anomaly detector is based on binary hypothesis testing. The BHT detector relies on the premise that in the absence of attacks and during normal system operation in the presence of uncertainties and disturbances,

the residuals, $\epsilon(t_k)$, almost always lie within an ellipsoid that remains invariant under state transitions. The ellipsoid defines a safe region and, at every time instant, the residuals are checked for violations of the safe region. A monitoring interval is defined, and the information about the violations of the safe region gathered over the monitoring interval is used in the BHT anomaly detector to detect an attack.

The invariant ellipsoid is defined using the simulation dataset described earlier along with tools from convex optimization to obtain a minimum volume ellipsoid encompassing a finite set [33]. Let the invariant ellipsoid be described as

$$\boldsymbol{\varepsilon} = \{v \mid \|Av + b\|_2 \leq 1\},$$

where $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $\|\cdot\|_2$ is the standard Euclidean norm. The finite set $\mathcal{C}$ is defined as $\mathcal{C} = \{x_1, \ldots, x_N\} \subset \mathbb{R}^3$, where each x_i is computed from the EKF output and the UAS position measurement from the simulation dataset. The problem of finding the minimum volume ellipsoid can be written as

$$\begin{aligned} &\text{minimize} \quad \log\det A^{-1} \\ &\text{subject to} \quad \|Ax_i + b\|_2 \leq 1,\ i = 1, \ldots, N, \end{aligned}$$

where A and b are the variables of the optimization problem. Let the number of violations of the invariant ellipsoid within the monitoring interval be denoted by the random variable X. The random variable X is generated from one of the two probability distributions, $f \in \mathbb{R}^{m+1}$ and $g \in \mathbb{R}^{m+1}$, where m is the length of the monitoring interval and is chosen as 40 time steps. The probability distribution f corresponds to a normal situation when there is no attack, and the probability distribution g corresponds to a situation when there is an attack on the sensor. The distributions, f and g, are obtained from the simulation dataset by first computing the residuals $\epsilon(t_k)$ at each time instant and then checking for violations of the invariant ellipsoid $\boldsymbol{\varepsilon}$ throughout the monitoring interval. We use a moving monitoring interval, whereby at each time instant, the values of $\epsilon(t_k)$ at the previous 40 time steps including the current time step are considered.

Let $T \in \mathbb{R}^{2\times(m+1)}$ denote a non-negative matrix, where the sum of each column entries is equal to one. Then, the detection probability matrix can be defined as

$$D = \begin{bmatrix} Tf & Tg \end{bmatrix} = \begin{bmatrix} 1 - P_{fp} & P_{fn} \\ P_{fp} & 1 - P_{fn} \end{bmatrix},$$

where P_{fp} and P_{fn} are the probabilities of false positive and false negative, respectively. Since the detector design problem is a bi-criterion vector optimization problem with competing objectives P_{fp} and P_{fn}, it is solved by scalarization, resulting in the following scalar optimization problem:

$$\begin{aligned} &\text{minimize} \quad P_{fp} + \lambda P_{fn} \\ &\text{subject to} \quad t_{1j} + t_{2j} = 1, \quad t_{ij} \geq 0,\ i = 1, 2 \text{ and } j = 1, \ldots, m+1. \end{aligned}$$

The optimization variables are t_{ij}, where t_{ij} are the elements of T. For each positive value of λ, a Pareto-optimal detector is obtained. In this work, a value of 21 is chosen for λ, and a deterministic likelihood ratio detector is used. Namely, given a value of X, a likelihood ratio threshold test is applied to determine if X was generated by distribution f (no attack) or g (attack); see [33] for more details.

Anomaly Detector	**Residual Generator**	**False Positive Rate** (%)	**False Negative Rate** (%)	**Mean Detection Latency** (sec)
CUSUM	1-norm	4.83	0.88	0.12 (CA) 1.05 (SA)
CUSUM	χ^2	1.05	0.63	0.13 (CA) 2.30 (SA)
SPRT	1-norm	5.47	0.91	0.12 (CA) 0.94 (SA)
SPRT	χ^2	0.97	0.55	0.11 (CA) 2.13 (SA)
BHT	-	0.17	0.74	0.67 (CA) 0.72 (SA)

Note: CA denotes constant attack and SA denotes scaling attack

TABLE I: Comparison of the performance of different anomaly detectors based on the simulation dataset.

D. Comparative Performance of the Anomaly Detectors

The performances of the three anomaly detectors with the 1-norm residual generator and the χ^2-residual generator are summarized in Table I. The performance metrics considered are the false positive rate, false negative rate, and the average time taken by the detector to detect the attack. It is observed that the χ^2-residual generator reduces the number of false alarm rates significantly compared to the 1-norm residual generator for both the SPRT and the CUSUM anomaly detectors. The SPRT anomaly detector with the χ^2-residual generator has the lowest false negative rate, and the BHT anomaly detector has the lowest false positive rate among all the detectors. In terms of detection latency, the SPRT and the CUSUM anomaly detectors have comparable performance, and both detectors have a lower mean detection latency for the constant attack compared to the scaling attack. The BHT anomaly detector has comparable values of mean detection latency for both the constant and scaling types of attack.

IV. DETECTOR TUNING BASED ON FLIGHT TESTS

The anomaly detectors described in the previous section are designed based on a simulation dataset. In order to assess the effectiveness of the designed anomaly detectors in detecting attacks in the presence of actual exogenous disturbances experienced by an sUAS, a large number of flight tests with simulated spoofing attacks on the GPS are conducted. The objective of the flight tests is to subject the sUAS to different atmospheric conditions by varying the following factors:

1) *Type of controller* - Five different types of controllers are used in the flight tests. Four of them are path-following controllers that track a predefined geometric

path in space, and the fifth controller is a trajectory-tracking controller which tracks a time-parameterized path in space. Two of the path-following controllers are designed based on a lumped model, whereby the path-following dynamics are combined with the UAS dynamics. The lumped system is used to design a linear-time invariant (LTI) controller and a linear-parameter varying (LPV) controller, both with $\mathcal{H}_\infty$ type performance. The other two path-following controllers are based on a conventional cascaded architecture composed of an outer guidance loop and an inner stabilization loop. The interested reader is referred to [31] for more details on the design of the different controllers. The trajectory-tracking controller is a standard $\mathcal{H}_\infty$ controller.

2) *Type of path* - Three different types of reference path are considered, namely a circular path of radius $110.5\,\mathrm{m}$, a lemniscate path with a maximum curvature of 0.0071, and a time-parameterized circular path of radius $110.5\,\mathrm{m}$. The reference paths considered are restricted to planar paths.
3) *Type of attack* - Similar to the simulation dataset, two types of GPS spoofing attack are considered, namely the constant attack and the scaling attack.
4) *Magnitude of attack* - Two different magnitudes of attack are considered for the constant attack: $10\,\mathrm{m}$ and $15\,\mathrm{m}$. The attack magnitudes considered for the scaling attack are $2.5\,\mathrm{m/s}$ and $4.0\,\mathrm{m/s}$. In both types of attack, the bias value due to the attack is added to the x_E position measurement.

The flight tests are performed on a small fixed-wing UAS platform, which is based on the commercially available Senior Telemaster airframe [26]. The UAS platform consists of the following sensors: a barometric pressure sensor, a differential pressure sensor, a satellite-based augmentation system (SBAS) enabled U-blox NEO-7 GPS module, and a miniature MPU 6000 3-axis accelerometer/gyroscope. The angle of attack and angle of sideslip are provided by an in-house built five-hole airdata probe. The Autopilot system is composed of a 3DR Pixhawk [34] and a Gumstix Overo Fire [35]. The Pixhawk portion of the autopilot handles the input/output tasks and redundancy management, while the Gumstix portion of the autopilot executes the control algorithms. The UAS platform, along with the autopilot system, is shown in Figure 7. Figure 8 shows two representative segments from the flight tests, where a constant attack and a scaling attack are simulated. Figure 8a shows a flight segment where the UAS is tracking a time-parameterized circular path and a scaling attack of magnitude $2.5\,\mathrm{m/s}$ is simulated. In Figure 8b, the UAS tracks a lemniscate path with an LTI path-following controller, and a constant attack of magnitude $10\,\mathrm{m}$ is simulated.

The data obtained from the flight tests are used to re-tune the anomaly detectors designed in Section III. It is observed that the parameters of the anomaly detectors designed based on the simulation dataset are reasonably well tuned, and only

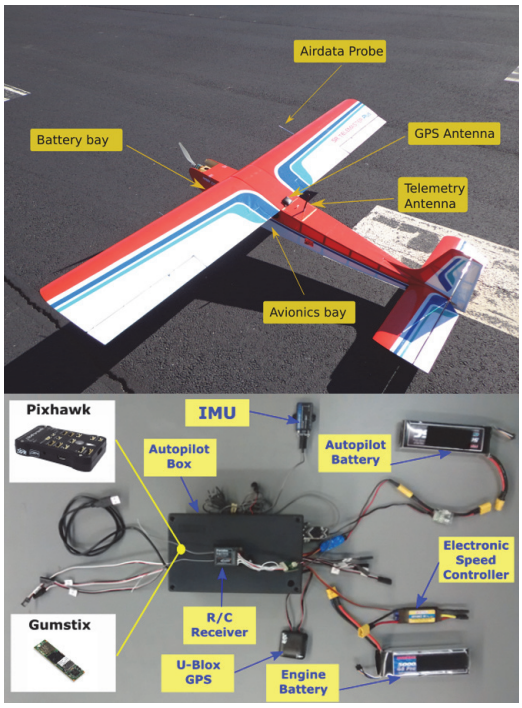
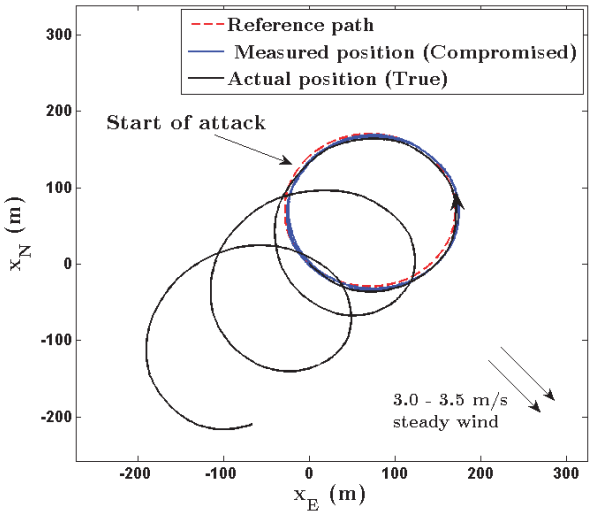


Fig. 7: The UAS platform and the autopilot system used in the flight tests.

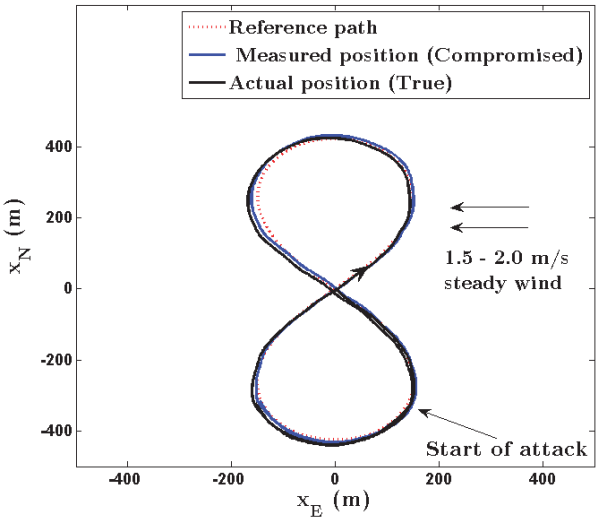
the drift term ν needed to be re-tuned to get rid of the false positives due to the presence of significant atmospheric disturbances. The re-tuned values of ν for the SPRT and the CUSUM anomaly detectors are $\nu = 2.53$ and $\nu = 1.87$, respectively. The performance of the SPRT anomaly detector after re-tuning is shown in Figure 9 for the data gathered from one of the flight tests, where the UAS is tracking a lemniscate path and a scaling attack of magnitude $2.5\,\mathrm{m/s}$ is simulated. The detection latency for the case shown in Figure 9 is $1.9\,\mathrm{s}$. It is noted that the anomaly detector is not implemented in real-time during the flight tests, but is run off-line after the flight tests. It is planned to implement the anomaly detectors onboard the UAS platform to assess the real-time performance of the anomaly detectors. Based on the analysis performed using the flight test data, it is observed that the anomaly detectors are hardly sensitive to the type of path or the type of controller. However, the type of attack and the attack magnitude have an influence on the detection latency. The detection latency is higher for the scaling type of attack compared to the constant attack as also noted in Section III. It is therefore inferred that a limited number of flight tests are sufficient to re-tune the anomaly detectors instead of extensive flight testing.

V. THE BAYESIAN NETWORK FRAMEWORK

Under atypical operational conditions, such as severe atmospheric turbulence or transient sensor faults, it is highly likely that the anomaly detection methods described in the previous sections may result in false positives. A probabilistic framework, therefore, can help in identifying attacks with lesser false alarm rates. In this work, we choose the Bayesian



(a) Scaling attack



(b) Constant attack

Fig. 8: Representative segments during flight tests with a simulated spoofing attack on the GPS.

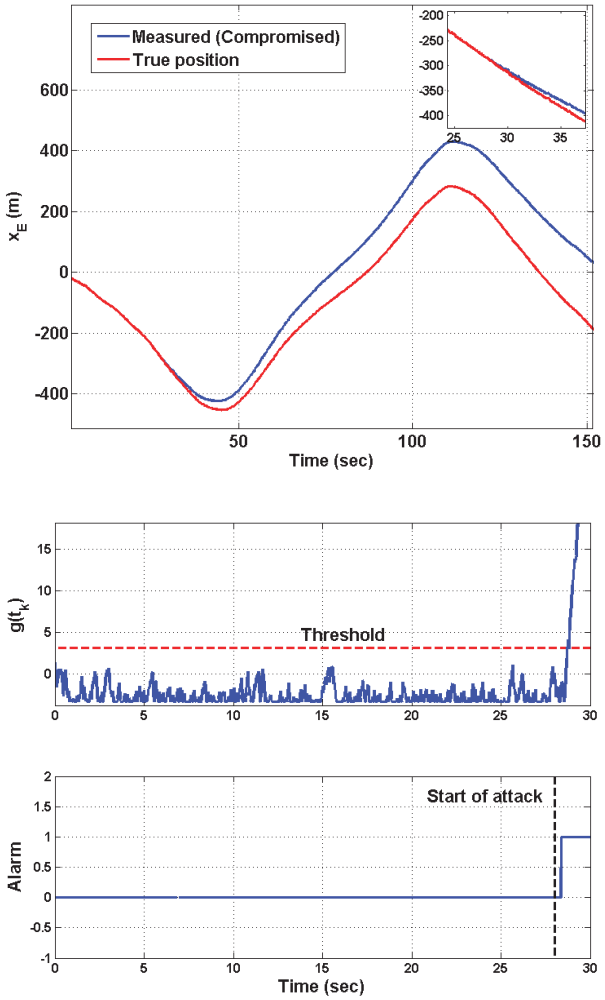


Fig. 9: Detection performance of the SPRT anomaly detector with the χ^2-residual generator for a segment gathered from one of the flight tests.

network (BN) as the probabilistic framework. A BN for a set of variables Z consists of a directed acyclic graph that encodes a set of conditional independence assertions about variables in Z, and a set P of local probability distributions associated with each variable. Many studies have used BN for anomaly detection in order to minimize the effect of uncertainties on the detection performance [9], [15]. The BN uses outputs from the residual-based anomaly detectors and the attack-signature-based anomaly detectors as evidences to declare an attack through Bayesian inference.

To illustrate the BN, we consider the case study discussed earlier, which involves a spoofing attack on the GPS. The BN for this case study is shown in Figure 10. The set of variables Z of the BN consists of the binary variables given by

$$Z = \{V_1, V_2, S_1, S_2, S_3, A_1, A_2, A_3\}.$$

The binary variables V_1 and V_2 take values in the set $\{\texttt{comp}, \texttt{not-comp}\}$ corresponding to whether the vulnerable sensor is compromised or not-compromised. S_1, S_2, and S_3 take values in the set $\{\texttt{faulty}, \texttt{not-faulty}\}$ corresponding to whether the measurement from the safe sensor is faulty or non-faulty. A_1, A_2, and A_3 take values in the set $\{\texttt{detected}, \texttt{not-detected}\}$ corresponding to whether an attack is detected or not. It is noted that the Attack node shown in Figure 10 serves as a binary addition node and is used only for convenience in the BN model. Except for the sensor nodes, each node has a parent and a conditional probability table (CPT) associated with it. For this case study, the conditional probabilities for the nodes are computed based on the simulation dataset. Firstly, the failure probabilities of the IMU, the airdata probe, and the pressure altimeter are chosen as 0.03, 0.08, and 0.08, respectively. The probabilities that the vulnerable sensors, V_1 and V_2, are compromised are assumed to be both equal to 0.85. The CPTs for the nodes A_1, A_2, and A_3 are constructed by simulating attacks and sensor failures as per the assumed probabilities and computing the probability with which the

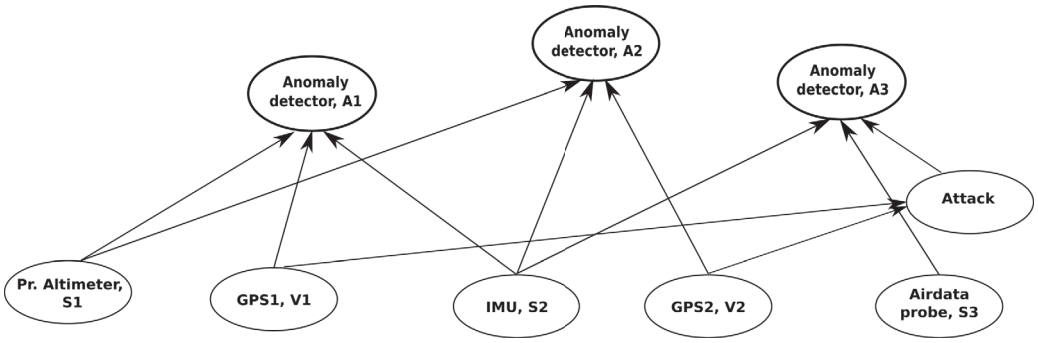


Fig. 10: The Bayesian network for the GPS spoofing attack case study.

$S_1 =$ not-faulty	not-detected	detected
comp, faulty	0.07	0.93
not-comp, faulty	0.96	0.04
comp, not-faulty	0.01	0.99
not-comp, not-faulty	0.995	0.005

$S_1 =$ faulty	not-detected	detected
comp, faulty	0.095	0.905
not-comp, faulty	0.95	0.05
comp, not-faulty	0.02	0.98
not-comp, not-faulty	0.992	0.008

TABLE II: The conditional probability table for node A_1.

anomaly detector detects the attack in each case. An SPRT detector with a χ^2-residual generator is considered as the anomaly detector for the nodes A_1 and A_2. The node A_3 corresponds to an attack-signature-based anomaly detector. A representative CPT, which corresponds to node A_1, is shown in Table II.

The BN is modeled using the free-to-use software HuginLite [36], which performs Bayesian inference using the method described in [37]. The BN model is subjected to different scenarios and Bayesian inference is performed to infer the compromised sensor from the two vulnerable sensors. During Bayesian inference, a threshold probability of 0.9 is used to declare whether a particular sensor is compromised or healthy. Consider a scenario where the node S_2, which denotes the IMU, gives faulty measurements, and because of these faulty measurements, the anomaly detector A_3 detects an attack. The anomaly detectors A_1 and A_2, however, do not detect an attack. When Bayesian inference is performed for this scenario with the known evidences, we obtain the probabilities of the sensors V_1 and V_2 being compromised as 0.033 and 0.021, respectively. The probabilities are small enough that no attack is declared. Consider another scenario where the airdata probe, which is denoted by the node S_3, is faulty and the anomaly detector A_1 detects an attack. The anomaly detectors A_2 and A_3 do not detect an attack. The reason that the anomaly detector A_3 does not detect the attack is due to the fault in the airdata probe. The probability that the sensor V_1 is compromised is inferred as 0.94, which is 0.042 less than the probability for the case where sensor S_3 is not faulty. Nevertheless, a probability of 0.94 is sufficient for declaring a sensor attack on V_1. In the absence of the Bayesian network, these two scenarios would have resulted in a difficult problem of ascertaining whether an attack did happen, given the evidences from the different anomaly detectors A_1, A_2, and A_3. These two scenarios elucidate the advantages of using the Bayesian network in the attack detection framework.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a framework for detection of cyber-physical attacks on the sensors of an sUAS. The framework uses knowledge of the physical system and techniques from statistical analysis to design anomaly detectors for detection of sensor attacks. The framework also makes use of a Bayesian network which uses the outputs of the anomaly detectors as evidences to infer an attack on the sensors. The explicit use of the dynamics of the physical system and the BN in the attack detection framework minimizes the false alarm rates, which is a crucial problem for sUAS that typically operate in a highly uncertain environment composed of atmospheric disturbances and sensor noise.

The work presented here addresses the problem of identifying sensor attacks on sUAS and considers only spoofing attacks. An extension of this work will include addressing other types of attack such as the replay attack, as well as other types of spoofing attack, apart from the constant and scaling attacks considered here. For instance, the present framework will not be able to detect threats that involve a combination of piecewise-constant attacks of small magnitude. Future work will include developing methods to detect such stealthy attacks. Another area of future work is to develop methods to detect attacks on the other two security layers of the UAS, namely the communication layer and the computer layer.

ACKNOWLEDGEMENT

The authors would like to thank Jean-Michel Fahmi for his help in generating the simulation dataset and designing the binary hypothesis testing detector.

REFERENCES

[1] Federal Aviation Administration, "FAA Aerospace Forecast (Fiscal Years 2016-2036)," pp. 30–33, 2016.

[2] (2016) Hackers are able to seize control of consumer drones and make them fall from the sky. [Online]. Available: http://www.recode.net/2016/10/28/13406082/hackers-control-consumer-drones-ftc-security

[3] R. V. Jategaonkar, *Flight Vehicle System Identification: A Time-Domain Methodology*, 2nd ed. Reston, VA: AIAA Progress in Aeronautics and Astronautics, 2015.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[5] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Başar, and J. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys*, vol. 45, no. 3, pp. 25:1–25:39, 2013.

[6] R. Mitchell and I. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014.

[7] ——, "A survey of intrusion detection techniques for cyber physical systems," *ACM Computing Surveys*, vol. 46, no. 4, pp. 55:1–55:29, 2013.

[8] J. Bigham, D. Gamez, and N. Lu, *Lecture Notes in Computer Science*. Springer, 2003, vol. 2776, ch. Safeguarding SCADA Systems with Anomaly Detection, pp. 171–182.

[9] S. Krishnamurthy, S. Sarkar, and A. Tewari, "Scalable anomaly detection and isolation in cyber-physical systems using bayesian networks," in *Proceedings of the ASME Dynamic Systems and Control Conference*, 2014.

[10] A. A. Cardenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection and response," in *Proceedings of the 6^{th} ACM Symposium on Information, Computer and Communications Security*, 2009, pp. 355–366.

[11] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, 2014.

[12] R. Mitchell and I. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593–604, 2014.

[13] J. V. Deshmukh, A. Donze, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, *Lecture Notes in Computer Science*. Springer, 2015, vol. 9333, ch. Robust Online Monitoring of Signal Temporal Logic, pp. 55–70.

[14] A. Jones, Z. Kong, and C. Belta, "Anomaly detection in cyber-physical systems: A formal methods approach," in *Proceedings of the 53^{rd} IEEE Conference on Decision and Control*, 2014, pp. 848–853.

[15] J. Schumann, P. Moosbrugger, and K. Y. Rozier, *Lecture Notes in Computer Science*. Springer, 2015, vol. 9333, ch. R2U2: Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems, pp. 233–249.

[16] Y. Mo and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *Proceedings of the 49^{th} IEEE Conference on Decision and Control*, 2010, pp. 5967–5972.

[17] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[18] J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee, "Sensor attack detection in the presence of transient faults," in *Proceedings of the 6^{th} ACM/IEEE International Conference on Cyber-Physical Systems*, 2015, pp. 1–10.

[19] Q. Zhu, C. Rieger, and T. Başar, "A hierarchical security architecture for cyber-physical systems," in *Proceedings of the 4^{th} International Symposium on Resilient Control Systems*, August 2011.

[20] K. G. Vamvoudakis, J. P. Hespanha, B. Sinopoli, and Y. Mo, "Detection in adversarial environments," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3209–3223, 2014.

[21] R. Inam, J. Mäki-Turja, M. Sjödin, S. M. H. Ashjaei, and S. Afshar, "Support for hierarchical scheduling in FreeRTOS," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, Sept 2011, pp. 1–10.

[22] K. G. Lyn, L. W. Lerner, C. J. McCarty, and C. D. Patterson, "The trustworthy autonomic interface guardian architecture for cyber-physical systems," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1803–1810.

[23] Z. Franklin, C. Patterson, L. Lerner, and R. Prado, "Isolating trust in an industrial control system-on-chip architecture," in *Resilient Control Systems (ISRCS), 2014 7th International Symposium on*, Aug 2014, pp. 1–6.

[24] L. Lerner, Z. Franklin, W. Baumann, and C. Patterson, "Application-level autonomic hardware to predict and preempt software attacks on industrial control systems," in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, June 2014, pp. 136–147.

[25] W. R. Dufrene, "Mobile military security with concentration on unmanned aerial vehicles," in *Proceedings of the 24^{th} AIAA/IEEE Digital Avionics Systems Conference*, 2005, pp. 8.D.3.1–8.D.3.8.

[26] Hobby express senior telemaster plus. [Online]. Available: http://www.hobbyexpress.com/senior_telemaster_plus_1034837_prd1.htm

[27] S. Gage, "Creating a unified graphical wind turbulence model from multiple specifications," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2003.

[28] B. Gibbs, *Advanced Kalman Filtering, Least-squares and Modeling: A Practical Approach*. Wiley New Jersey, 2011.

[29] J.L.Crassidis, *Optimal Estimation of Dynamic Systems*. CRC Press, 2012.

[30] F. Gustafsson, *Adaptive filtering and change detection*. Wiley New York, 2000.

[31] D. Muniraj, M. Palframan, K. Guthrie, and M. Farhood, "Path-Following Control of Small Fixed-Wing Unmanned Aircraft Systems with $\mathcal{H}_\infty$ Type Performance," 2017, submitted for publication to Control Engineering Practice.

[32] ArduPilot, "Ardupilot Autopilot Suite," 2016. [Online]. Available: http://ardupilot.org/ardupilot/

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[34] Pixhawk, "Pixhawk Autopilot hardware," 2017. [Online]. Available: https://pixhawk.org/modules/pixhawk

[35] Gumstix, "Gumstix Overo Fire," 2017. [Online]. Available: https://store.gumstix.com/coms/overo-coms/overo-firestorm-y-com.html

[36] Hugin Expert, "HuginLite: A Bayesian network based decision tool," 2017. [Online]. Available: http://www.hugin.com/index.php/hugin-lite/

[37] F. V. Jensen, S. L. Lauritzen, and K. G. Olesen, "Bayesian updating in causal probabilistic networks by local computations," *Computational Statistics Quarterly*, pp. 269–282, 1990.