2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Unmanned Aerial Vehicle Security Using Recursive Parameter Estimation

Zachary Birnbaum Andrey Dolgikh Victor Skormin Edward O'Brien Daniel Muller
Electrical and Computer Engineering
Binghamton University
Binghamton, NY, USA
{zbrinba1, adolgik1, vskromin, eobrien8, dmuller5}@binghamton.edu

Abstract—**The proliferation of Unmanned Aerial Vehicles (UAVs) raises a host of new security concerns. Our research resulted in a prototype UAV monitoring system, which captures flight data and performs real-time estimation/tracking of airframe and controller parameters utilizing the Recursive Least Squares Method. Subjected to statistical validation and trend analysis, parameter estimates are instrumental for the detection of some classes of cyber attacks and incipient hardware failures that can invariably jeopardize mission success. Our results demonstrate that achieving efficient anomaly detection during flight is possible through the intelligent application of statistical methods to system behavioral profiling.**

Keywords — unmanned aerial vehicle, security, parameter estimation, safety

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) play an ever increasing role in today's world, ranging from research applications and military operations to civilian service [1]. Since 2011, the number of distinct UAV models rose by 20%, UAV models for military applications by 40%, and the number of countries involved in UAV production grew by 30% [2]. UAV development will continue to be an evolving field, as researchers in a variety of disciplines strive to find novel ways to use UAVs. [3]. There are countless ways UAVs can be integrated into society for domestic applications including search and rescue, weather forecasting, law enforcement, firefighting, precision farming, and communications [4].

However, the widespread use of UAVs gives rise to a plethora of new security concerns. With U.S. airspace suddenly filling up with UAVs, new sensing and avoidance capabilities for improved flight reliability are being developed [5]. UAV data integrity and health monitoring are now more important than ever to prevent drones from falling out of the sky or flying to an unauthorized location. Even military UAV data integrity has been brought into question [6, 7].

In this paper we put forth a system for health monitoring and security assessment of UAVs. The system operates in real time to accurately estimate the UAV control laws and airframe parameters. The system would detect immediate changes in critical UAV parameters due to hardware failure or cyber attack.

II. THREAT MODEL

We postulate UAV threat vectors as follows:

Hardware failure is the malfunctioning of the UAV's mechanical or electronic systems that leads to full or partial inability of the UAV to complete its mission. UAV hardware failure can occur for any number of reasons including environmental factors, battlefield damage, and general wear. It is of utmost importance that in the event of hardware failure, damage is minimized. This safeguarding can be done by utilizing existing adaptive control algorithms [8, 9]. If possible, in the event that hardware failure is detected, the UAV should take action accordingly.

Malicious hardware is a threat where part of the UAV's hardware has been tampered with to make it behave in a way not specified by its designers. The malicious hardware is a fairly recent attack vector that assumes that the hardware has been tampered with to accomplish a specific goal, which can range from failing under certain conditions to the exfiltration of confidential data. An example of this attack is implantation of monitoring devices into more than 100,000 computers to conduct surveillance [10].

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

Attacks against the **Flight Control Computer** can result in the adulteration of flight control laws by jeopardizing communication onboard the UAV, and resulting in the changing of mission parameters. The appropriate defenses against such attacks are determined by the specifics of the onboard hardware and software. These may include real-time estimation of immediate controller status and the flight plan based on readily available sensor data.

Attacks against the **Navigation Sensors** can result in a loss of positional information or receiving modified or misleading feedback information. For example, onboard GPS sensors can be attacked with various spoofing techniques as evidenced by Shepard et al. in [11].

Communications Channel attacks against a UAV can result in a loss of communication with the ground control station or in malicious commands being sent to and accepted by the UAV. UAVs currently used by the U.S. military accept only encrypted commands thus making this a difficult attack vector for military drones. However, most civilian drones currently use unencrypted communication links that make sending false and malicious data a viable possibility [12].

Attacks against the **Ground Control Station** (GCS) are attacks against a computer running dedicated software to control and communicate with the UAV. A successful attack would enable an attacker to send erroneous or malicious commands to the UAV. This attack vector is the most difficult to detect from the UAV because the UAV has no means to verify if the GCS has been exploited. While it is possible to implement multiple ground station authentication techniques, many civilian UAVs and ground control systems do not currently support this scheme [13], [14].

III. CONTRIBUTIONS

This paper focuses on addressing the first three types of attacks: hardware failure, malicious hardware and attacks against the flight control computer. Our approach is the development of a system that monitors UAV behavior and detects the aforementioned attacks.

This system utilizes the following technologies:

- Automatic detection of changes in UAV airframe dynamics indicative of mechanical degradation
- Automatic detection of changes in UAV flight control law indicative of cyber attacks

IV. APPROACH

Our approach relies on the ability to identify and track flight dynamics of the UAV. We use the Recursive Least Squares (RLS) method to identify immediate values of system parameters and detect deviations from their nominal values. Because the change of the flight dynamics is usually caused by incipient mechanical failures or ice buildup, we can use the RLS method to identify these conditions.

In a similar way, flight controller parameters may change unexpectedly under cyber attack, software bugs, or electronics failure. By monitoring flight controller parameters, we can detect if the controller behave according the design specifications.

A. Recursive Least Squares

This method offers a recursive modification of the Least Squares procedure ideal for real-time estimation/tracking of system parameters based on on-going measurements of relevant input/output signals.

The RLS method is well described in [15]. The RLS procedure is a numerically efficient algorithm that enables estimation/tracking of the parameters A of a linear equation $y(t) = V^T(t)A$, where $y(t)$ represents the output variable and $V(t)$ is the vector of input variables. At each time step, separated by a constant Δt, the procedure acquires two new measurements, $y(t)$ and $V(t)$, and updates the previously obtained parameter estimation $\hat{A}(t-1)$, i.e. computes $\hat{A}(t)$. With reasonable initial conditions and enough samples, the sequence of parameter estimates, $\hat{A}(t)$, t=1,2,3,... will eventually converge to the "true" parameter values A^{TRUE}. A finite-memory version of the RLS procedure allows for tracking time-dependent parameters $A^{\mathrm{TRUE}}(t)$. Note that for a constant time step Δt, $t=k\,\Delta t$, and $y(t)$, $V(t)$ turn into $y(k)$, $V(k)$.

To adopt this method for our needs, we define the input vector as a concatenation of vectors containing system inputs and outputs

$$V(k) = [u_1(k), u_2(k), \dots u_m(k), x_1(k), x_2(k), \dots x_n(k)]^{\mathrm{T}} \quad (1)$$

where

$k = 1,2, \dots N$ is the discrete-time index,

$u_i(k)$ – i-th input $i \in 1..m$

$x_j(k)$ – j-th output $j \in 1..n$

For our flight model, we can choose to monitor all available flight data including: roll, pitch, yaw, elevator, throttle, rudder, aileron, and GPS location.

The *N*-th step of RLS procedure is outlined in Figure 1.

1. Initialization of $A_0, P_0, N = 1$
2. Acquisition of $y(N), V(N)$
3. Define Kalman Gain
$$K_N = P_{N-1}V(N)\left(\mathrm{I} + V(N)^T P_{N-1} V(N)\right)^{-1}$$
4. Update estimated parameters
$$A(N) = A(N-1) + K_N[y(N) - V(N)^T A(N-1)]$$
5. Computer matrix P for the next iteration
$$P_N = (\mathrm{I} - K_N V(N)^T) P_{N-1}$$
6. N=N+1, return to step 2

Figure 1. Recursive Least Squares procedure implementation

where:

A_0 represents the initial values of the estimated parameters (nominal parameter values are utilized).

P_0 is the covariance matrix of vector *V(t)*

I is the identity matrix

K_N represents the Kalman Gain

In the following sections, we will show how the RLS method can be applied to UAV parameter estimation. We present the application of RLS to difference equation parameter estimation and to state space parameter estimation.

B. Difference Equation Parameter Estimation

The majority of commercially available inexpensive autopilots use a Proportional-Integral-Derivative (PID) controller for flight control and stabilization [16]. A PID controller operates by calculating the error, between a set reference and the output of a plant or process. It then attempts to minimize this error by modifying controller outputs. A PID controller block diagram is illustrated in Figure 2.

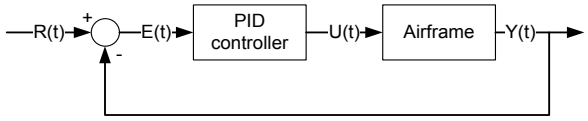

Figure 2. PID controller

where:

R(t) represents the reference signal or desired position and location of the aircraft.

E(t) represents the reference error.

U(t) represents the control signals generated by the PID controller .

Y(t) represents the current position and location of the aircraft.

A PID control law is defined as follows:

$$U(t) = a_p E(t) + a_i \int_o^t E(\tau)d\tau + a_d \frac{dE(t)}{dt} \quad (3)$$

Taking the Laplace Transform of (3) and expressing the result as a difference equation yields:

$$U(t) = a_1 E(t) + a_2 E(t-1) + a_3 E(t-2) + b_1 U(t-1) + b_2 U(t-2) \quad (4)$$

Applying the RLS procedure, displayed in Figure 1, leads to

$y(t)=U(t)$, $A=[a_1, a_2, a_3, b_1 b_2]^{\mathrm{T}}$,

$V(t)=[E(t), E(t-1), E(t-2), U(t-1), U(t-2)]^{\mathrm{T}}$,

and to the estimation of parameter values *A* based on the actual UAV data. Note that statistically significant difference between the estimated and nominal parameter values would manifest a successful cyber attack on the UAV computer. differ from their nominal values.

C. State space parameter estimation

A dynamic system, such as an UAV, can be described in the matrix-vector format [17]. The following constitutes an open-loop system description that represents the propertied of the airframe (no feedback controller in the loop).

$X(t+1)=QX(t)+BU(t)$ (5)

$Y(t)=CX(t)$ (6)

where:

t=1,2,3,... is discrete time index,

$X(t) = [x_1(t)\ x_2(t)\ x_3(t)\ ...\ x_m(t)]^T$ represents the state vector of the system comprising m state variables,

$Y(t) = [y_1(t)\ y_2(t)\ y_3(t)\ ...\ y_n(t)]^T$ represents the output vector of the system,

$U(t) = [u_1(t)\ u_2(t)\ u_3(t)\ ...\ u_k(t)]^T$ represents an input vector containing control efforts and external forcing functions.

Q is a $m \times m$ matrix and is known as the fundamental matrix of the system, representing system inertia and interactions between particular state variables. In our case, matrix Q represents the unique dynamic properties of the airframe reflecting its inertia and interaction with the environment. Matrix B has dimension $m \times k$ and describes the effect of forcing functions on individual state equations. One can realize that such factors as ice buildup, mechanical failures and battlefield damage can be detected and assessed in flight only by the estimation of matrix Q and matrix B of the airframe.

C is a $n \times m$ matrix providing the definition of the system's output variables through the state variables. In our system we assume that $n = m$ and matrix C is an identity matrix of $n \times n$.

While (5), (6) represent of the UAV airframe, a complete description of UAV dynamics includes a controller capable of automatically stabilized flight that follows waypoints and avoids obstacles. This state space description is referred to as the closed-loop system as there is a feedback controller in the loop.

The matrix vector description of the controlled airframe and its state variable feedback controller can be written as follows:

$X(t+1)=QX(t)+BU(t)$ (7)

$Y(t)=CX(t)$ (8)

$U(t)=R(t)-FX(t)$ (9)

where:

F is a matrix containing the parameters of the state-variable controller,

R is the reference signal for the control loop, consistent with desired flight path or aircraft position,

The fundamental matrix for the closed-loop system in that case can be written as:

$Q^{CL}=Q - BF$ (10)

Using the RLS method enables us to estimate matrix Q and matrix B of the airframe, matrix Q^{CL} and finally matrix F of the state-variable controller.

Consider j-th state equation of (5):

$x_j(t+1)=\sum_{i=1}^{m} q_{ji}x_i(t)+\sum_{i=1}^{k} b_{ji}u_i\ (t)$,

assume

$y(N)=x_j(t+1)$,

$V(N)=[x_1(t),x_2(t),...,x_m(t),u_1(t),u_2(t),...,u_k(t)]^T$,

$A=[q_{j1},q_{j2},...,q_{jm},\ b_{j1},b_{j2},...,b_{jk}]^T$

Now, Fig.1 explains how the row number j of the matrix Q of the open-loop system could be estimated. This procedure, repeatedly utilized for every $j=1,2,..,m$ would result in the estimates $Q(N)$ and $B(N)$ at the every step of the RLS procedure.

A similar approach is applied for the estimation of the fundamental matrix of the closed-loop system, Q^{CL}. Indeed, assume

$y(N)=x_j(t+1)$,

$V(N)=[x_1(t),x_2(t),...,x_m(t),r_1(t),r_2(t),...,r_k(t)]^T$,

$A=[q^{CL}_{j1},q^{CL}_{j2},...,q^{CL}_{jm},\ b_{j1},b_{j2},...,b_{jk}]^T$

where

$r_i(t)$, i=1,2,...are component of the vector $R(t)$,

q^{CL}_{ij} are appropriate elements of matrix Q^{CL},

then every N-th step of the RLS procedure results in estimated matrix $Q^{CL}(N)$. Thus while estimate $Q(N)$ also exists, matrix F of the state-variable controller could be also estimated as

$F(N)=B(N)^{-1}[Q(N)-Q^{CL}(N)]$ (11)

This result is based on a realistic assumption that the number of reference signals $r_i(t)$ is equal to the number of state variables $x_i(t)$ and B is a square nonsingular matrix. It should be noted that (11) does not represent the solution of the controller design problem, instead, it is an attempt

to deduct the controller parameter values from the observed behavior of the UAV. This task facilitates the detection of significant discrepancies between the actual and nominal parameter values that could be attributed to a cyber attack.

D. Statistical Significance

The application of the RLS procedure to flight data yields a sequence of parameter estimates that fluctuate with time due to measurement noise and a number of unaccounted factors. These estimates are randomly distributed around some unknown "true" values that represent the immediate status of the UAV. Our effort is aimed at the detection and validation of the deviations between the immediate (estimated) and nominal parameter values known to UAV designers.

First, recall that for all practical purposes after a sufficient number of steps RLS estimation becomes equivalent to Least Squares estimation, and as such results in a minimum variance unbiased estimates of unknown parameters. The "true" value of the estimated parameter, ρ^{TRUE}, is defined as the mean value of the obtained estimates, $\rho(t)$, t=1,2,3,.... .

The distance between an RLS estimate of unknown parameter $a_i(N)$ and its "true" value a_i^{TRUE} assuming that N>>1 [24] can be characterized as:

$$\left|a_j(N)-a_j^{TRUE}\right| \le t(\alpha,\infty)S_E\sqrt{K_{jj}} = \Delta_j(\alpha) \quad (12)$$

where:

$t(\alpha,\infty)$ is Student distribution defined for the chosen level of significance α and infinite number of degrees of freedom,

S_E is standard deviation of the modeling error

$$E(N)=y(N)-A(N)^T V(N)$$

and K_{jj} is the j-th diagonal element of the covariance matrix of the 'input" vector $V(t)$.

One can conclude that a deviation between a nominal value of a system parameter a_j^{NOM} and its immediate value represented by estimate $a_j(N)$ is statistically significant, and as such manifests an anomaly, if

$$\left|a_j(N)-a_j^{NOM}\right| \ge \Delta_j(\alpha) \quad (13)$$

In other words, the occurrence of condition (13) can be interpreted as an indicator of a systematic change to the monitored parameters, caused by a cyber attack or physical changes in the airframe. The confidence level of such a conclusion is (1-2α).

V. IMPLEMENTATION

A. Hardware Health Monitor

We propose to embed within the UAV a hardware health monitoring system, shown in Figure 3, capable of detecting various types of cyber attacks as well as hardware failures. The monitoring system will be responsible for ensuring safe flight and completion of the flight mission

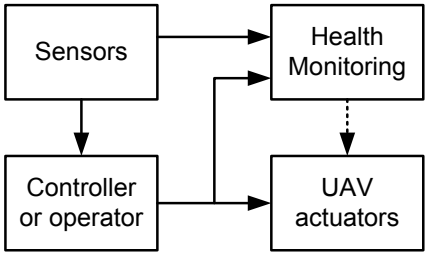

Figure 3. Hardware Health Monitoring System

The Health Monitor will take all available UAV data including sensor data, control surface positions and other control efforts to determine UAV health, which will be placed into one of the following categories:

- UAV operating normally
- UAV hardware failure
- UAV controller failure

The UAV health state can be determined by the implementation of the described approach to find statistically significant deviations from nominal UAV hardware and control parameters. If a significant deviation is detected, the failsafe procedures are executed. The failsafe protocol may depend on the severity of the detected failure e.g. the UAV may attempt to return back home or to execute an emergency landing.

VI. EXPERIMENTS

For experimental verification of our approach, we chose to use the ArduPlane platform. ArduPlane is an open source plane autopilot system developed by the DIY Drones community and is flashed onboard an Arduino micro controller board. The ArduPlane open source platform is ideal for experimentation as it allows for the modification of source code and the uploading of custom firmware to the pilot. ArduPlane is able to connect to any GCS which supports the MAVlink protocol. This allowed us to run fully scriptable missions in addition to real-time waypoint control and constant two-way communication to monitor the flight.

A. Data Collection

We chose to use a Hardware in the Loop (HIL) setup to run our experiments in the configuration displayed in Figure 4.

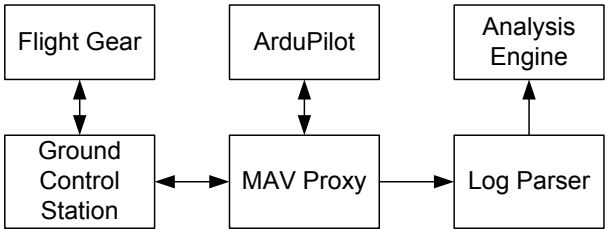

Figure 4. Simulation setup

Flight Gear, an open source flight simulator provides all flight data, including location, wind, and air pressure to the ground control station. which in turn relays it to MAV Proxy. MAV Proxy then sends this data to the ArduPilot, which generates control surface responses to maintain flight according to the plan generated by the GCS. The ArduPilot was configured to report all measured or controlled variables during the flight. This data is forwarded by MAV Proxy to the Log Parser and Analysis Engine. This arrangement allowed us to gather the necessary data for our experiments such as roll, pitch, and yaw, as well as the servo motor control efforts such as aileron, ruder, elevator, and throttle.

When running a flight simulation, environmental variables such as flight location, wind, time of day, and atmospheric pressure can affect collected measurements. Therefore, environment can also affect estimated parameters by introducing systemic errors. To keep the effect of the environmental conditions on our experiments minimal, we ran all experiments under the same environmental conditions and all of the test flights flew along the same path pictured in Figure 5.

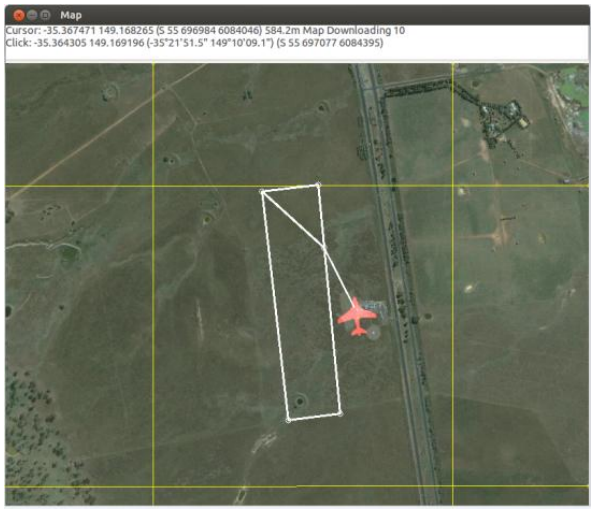

Figure 5. Flight path

B. Experimental Results

The experimental results were collected to empirically verify our contributions i.e. the ability to detect and interpret changes in airframe and control parameters. The first part of this section will demonstrate the performance of the parameter identification engine. The second part will show that we were able to detect significant changes in the UAV control parameters.

1) Identification of airframe and controller state space parameters

Before takeoff, we loaded our UAV controller with known nominal parameters for the UAV airframe to be flown. Then the UAV was flown along the

path shown in Figure 5 for 15 minutes while we collected UAV state data such as roll, pitch, yaw and servo responses. The measured airplane attitude data is presented in Figure 6.

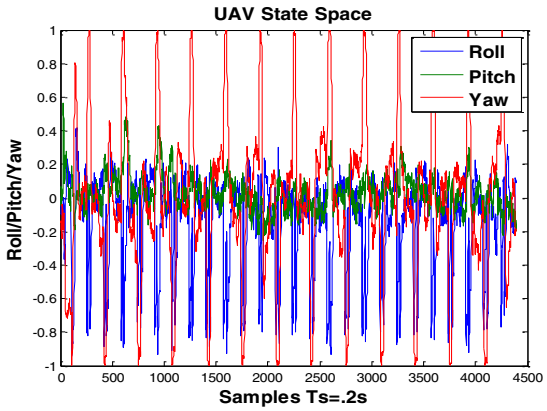

Figure 6. UAV state vector plot

Table 1. Identified UAV state space parameters (Normal Flight)

	normal flight 0		normal flight 1	
Parameter	OL	CL	OL	CL
1	1.02	1.01	1.02	1
2	0.04	-0.14	0.02	-0.15
3	0	0	0	-0.01
4	0.28	2.63	0.28	2.71
5	0.01	-2.49	0.01	-2.59
6	-0.3	-0.13	-0.3	-0.12
7	0.02	-0.03	0.02	-0.03
8	0.96	0.86	0.96	0.87
9	0.01	0	0	-0.01
10	0.06	-0.31	0.05	-0.39
11	-0.2	0.39	-0.19	0.44
12	0.11	-0.08	0.12	-0.06
13	-0.02	-0.01	0	-0.01
14	-0.04	-0.04	0.05	-0.02
15	1	1.01	1	1.01
16	-0.01	-0.02	0.01	-0.1
17	0.02	-0.45	-0.02	-0.4
18	-0.01	0.47	0.01	0.5

Each UAV state space data point was forwarded to our parameter estimation engine depicted in Figure 1. The parameter estimation engine produced eighteen estimates for the airframe and controller state space parameters. The parameters estimated for the closed loop (CL) and open loop (OL) systems are seen in Tables 1.

The estimated parameters in Table 1 cover 15-minute flights during which the simulated UAV executed about 6 loops and 24 turns. Table 1 demonstrates that the parameters remain constant between flights if the airframe and control law were not altered. Consider parameter 4 for the open loop, it has not changed remaining constant at 0.28. For the closed loop, parameter 4 changed negligibly. The same can be said about all other parameters, all parameters show little or no change.

Table 2. Identified UAV state space parameters (Altered Flight)

	altered flight 0		altered flight 1	
Parameter	OL	CL	OL	CL
1	0.87	1.05	0.87	1.05
2	-0.43	-0.93	-0.48	-0.93
3	0.02	0	0	0
4	0.17	5.43	0.17	5.58
5	-0.02	-4.79	0.02	-4.91
6	-0.21	-0.64	-0.23	-0.64
7	0	-0.05	-0.01	-0.05
8	0.79	0.8	0.77	0.78
9	0	-0.01	0	-0.01
10	0	-0.4	0	-0.42
11	-0.02	0.54	-0.02	0.51
12	0.02	-0.14	0.02	-0.09
13	0	0	0	-0.01
14	0	-0.04	-0.02	-0.06
15	1	1.01	1	1.01
16	0	-0.04	0	-0.08
17	-0.01	-0.48	0	-0.44
18	0.01	0.51	0	0.52

2) Identification of parameter alteration in state space

To test the ability of our system in detecting control parameter changes, we loaded the UAV controller with a different set of control law parameters and ran the same flight plan as presented in Figure 5 for ten minutes. According to our computations the estimated control law parameters did reflect the applied change in the

control law. When viewing the flight in the simulation environment we noticed that it was not as smooth when compared to normal flight.

We applied the same analysis method to the flight data collected with an altered flight controller. The estimates for the airframe parameters and for the flight controller are presented in Table 2.

The Table 2 clearly shows that the estimated parameters have changed noticeably from those in Table 1. This change is caused by the UAV being initialized with different controller parameters. For example, parameter 4 has changed from 2.71 to 5.58 due to change in the control law. The same effect can be observed for a number of estimated parameters.

To compare the parameters identified in each of the flights, we used Normalized Root Mean Square Deviation (NRMSD).

$$E(A, B) = \sqrt{\sum_{i=0}^{n}([a_i] - [b_i])^2 / var(a_i)} \quad (14)$$

where:

A, B are the flights being compared

$[a_i]$ and $[b_i]$ are the mean values for the i-th estimated parameters in flights A and B.

$var(a_i)$ variance of the i-th estimated parameter in flight A

The closer two sets of parameters are to each other, the lower the NRMSD will be. Table 3 compares deviations for flights in Tables 1 and 2 using NRMSD.

Table 3. Flight parameters comparison (Closed Loop)

	normal flight 0	normal flight 1	altered flight 0	altered flight 1
normal flight 0	0	0.77	139.38	147.81
normal flight 1	0.75	0	129.43	137.72
altered flight 0	7.85	7.40	0	0.22
altered flight 1	9.16	8.47	0.23	0

There was not much of a difference in the NRMSD value between flights when parameters were kept consistent. When parameters were changed e.g., between normal flight 0 and altered flight 0, the NRMSD takes a noticeably larger value. This result will allow us to build an anomaly detection system that detects parameter deviations.

The same method can be applied to the open loop system. Changes in the identified open loop system parameters may indicate physical damages to the airframe or flight equipment.

C. Results summary

Our experimental results successfully support our claim that it is possible to automatically identify airframe and control parameters of the UAV. We also confirmed that any significant changes to UAV control law can be detected. We believe that hardware degradation or damage can be identified with the same method.

VII. DISCUSSION

This paper presents a method for the automatic detection of anomalous status of a UAV that might be the result of cyber attack or mechanical failure. This method is designed to address particular types of anomalies associated with adulteration of control laws and degradation of the properties of the airframe. This method has limitations outside of its application area.

A. Limitations

The cyber attack against the UAV may have many different implementations and could cause failures at multiple levels. Our method reliably detects the changes in the UAV dynamics which is important for flight safety. Our method is not designed to and will not be able to detect attacks against the GPS system or the waypoint scheduling system.

The results presented in this paper were obtained in an advanced simulation environment which might not completely capture the complexity of the real world. However, we believe that all significant aspects of UAV flight were successfully modeled by our simulation.

VIII. RELATED WORK

The UAV security can be thought of as a special security domain that embraces various security related problems arising from the fusion

of multiple technologies that make UAVs fly. The body of work in communication security [18], host based security [19], and software security [20] can often be directly applied to UAVs. However, UAVs pose specific problems that cannot be addressed by simply applying known solutions [6, 11, 14].

There is not much work published in the field of UAV security to date. In this section we cover to the best of our knowledge the majority of work that has been published about UAV security.

Kim et al. [21] proposed a method to classify cyber attacks, hardware attacks, sensor spoofing attacks, and wireless attacks. Hardware attacks occur when the attacker has access to the UAV controller and components directly. Wireless attacks occur when attacks are carried out via a wireless communication channel. Sensor spoofing is defined as when an attacker passes false data through the onboard sensors, such as GPS, to the pilot. In addition, Kim et al. ran a number of experiments showing that UAV flight and behavior can be altered by carrying out varying attack types including gain scheduling, sensor fuzzing, and digital update delays. Using these attack vectors and a simulation environment, they showed that UAV flight and behavior can be altered. They also laid out a general framework for UAV anomaly detection but did not provide any implementation solutions.

Shepard et al. [11] showed that civilian, non-encrypted GPS signals can be hijacked by a spoofing device. The spoofer utilizes GPS observables and incorporates a necessary user controllable offset to generate fake GPS coordinates. The spoofed signal is then broadcast to the UAV, slowly drifting from the true GPS signal eventually leading to the GPS device being fully controlled by the spoofer. The developed GPS spoofer can alter a UAV's perceived location potentially leading to mid air collisions with other UAVs or buildings. Shepard et al. were even able to successfully maintain a short-term 3-dimensional control of a UAV. Shepard et al. proposes that long-term UAV control could be achieved using a medium-sized radar system in conjunction with a control loop feedback system.

In the event of GPS failure, Conte and Doherty [22] demonstrated that using onboard accelerometers, a gyroscope and a barometer combined with a camera, one can determine UAV position. Internal sensors alone cannot be used to determine position due to sensor drift. They proposed a navigation architecture which fuses information from inertial sensors, a video camera, and a barometric sensor. Sensor information is fused together using a Kalman filter to estimate UAV position. The video camera provides a visual odometer by comparing pictures of the terrain to calculate speed and distance traveled. Using a Geo-reference image database, the system is able to match database images with flight data providing an additional means of calculating position. Their results showed only 3 meters of drift from the true GPS position at the end of a 1 km flight using the developed navigation system.

Mitchell and Chen [23] proposed an adaptive behavioral specification based on an Intrusion Detection System (IDS). Their IDS uses exhaustive specification for all possible UAV states with respect to its equipment and its position in the air. If the UAV leaves the specified safe states it can be concluded that an attack was carried out or that the UAV has misbehaved.

The authors used a Monte Carlo simulation to model their system and to determine how accurately they can detect a statistically significant UAV safe or unsafe state. The system was not tested on actual or simulated UAV flights.

The approach proposed by Mitchel and Chen has one serious drawback that might prevent it from operating in real UAVs as an intrusion detection platform. Safety conditions often depend on the environment where the UAV flies. For example, the altitudes between 1000 and 2000 feet might be safe in flat areas but can be unsafe in mountainous regions. Therefore, additional safe corridors will need to be specified, extending the specification. There might also be some other conditions such as lead to rain or wind that might affect the safety specifications. These issues the uncontrollable number of possible states and the inevitable impossibility to cover all of them.

IX. CONCLUSION

The increasing widespread use of Unmanned Aerial Vehicles (UAVs) raises a host of new security concerns. We put forth a novel process for defining UAV flight behavior. Our proposed technology allows us to detect a cyber attack against UAV electronic hardware. It also can detect UAV hardware degradation and failure.

Our research resulted in a prototype UAV monitoring system, which captures flight data and estimates airframe and controller parameters. Estimated parameters are constantly compared to previously known parameters. If a significant change in parameters is observed, the system alerts the ground control station and may execute predefined actions.

We adapted the Recursive Least Squares method to implement an estimator for UAV parameters. This estimator is used in conjunction with the state space model parameters and difference equation parameters. The vectors of estimated parameters are accumulated and subjected to confidence interval testing. If the monitored parameters fall off the corresponding confidence interval, the statistically significant deviation is detected. We also show that simpler normalized root squared deviation can be used to detect anomalous parameter change.

We tested our system with a set of experiments based on a state of the art flight simulator. Our experiments demonstrated that it is possible to automatically establish UAV flight parameters and detect significant deviations.

Our results establish that achieving efficient anomaly detection in flight is possible through the intelligent application of statistical methods to system behavioral profiling.

ACKNOWLEDGMENT

This project became possible due to the support of graduate student research at Binghamton University by the Air Force Office of Scientific Research. In addition, the authors would like to thank Dr. George Siouris for his expertise and valuable suggestions.

REFERENCES

[1] Online, Amazon Prime Air, http://www.amazon.com/b?node=8037720011, accessed on 13 Feb 2014

[2] Online, UAV Roundup 2013, http://www.aerospaceamerica.org/Documents/AerospaceAmerica-PDFs-2013/July-August-2013/UAVRoundup2013t-AA-Jul-Aug2013.pdf, accessed on 13 Feb 2014

[3] Online, Federal Aviation Administration selects Rutgers for drone research program, http://www.dailytargum.com/news/federal-aviation-administration-selects-rutgers-for-drone-research-program/article_403dc68a-7888-11e3-a0c2-0019bb30f31a.html, accessed on 13 Feb 2014

[4] Online, Unmanned Aircraft Systems: PercePtions & Potential http://www.aia-aerospace.org/assets/AIA_UAS_Report_small.pdf, accessed on 13 Feb 2014

[5] Online, FAA Selects Six Sites for Unmanned Aircraft Research, http://www.faa.gov/mobile/index.cfm?event=news.read&update=75399, accessed on 13 Feb 2014

[6] Online, Rawnsley, A. "Iran's Alleged Drone Hack: Tough, but Possible," http://www.wired.com/dangerroom/2011/12/iran-drone-hack-gps, accessed on 13 Feb 2014

[7] Gorman, Siobhan, Yochi J. Dreazen, and August Cole. "Insurgents hack US drones." The Wall Street Journal 17 (2009), Dow Jones and Company, Inc.

[8] Online, Case Study: Rockwell Collins demonstrates damage tolerant flight controls and autonomous landing capabilities. http://www.rockwellcollins.com/sitecore/content/Data/Success_Stories/DARPA_Damage_Tolerance.aspx, accessed on 13 Feb 2014

[9] Nhan T. Nguyen and Kalmanje Krishnakumar. "Hybrid Intelligent Flight Control with Adaptive Learning Parameter Estimation", Journal of Aerospace Computing, Information, and Communication, Vol. 6, No. 3 (2009), pp. 171-186.

[10] Online, http://www.nytimes.com/2014/01/15/us/nsa-effort-pries-open-computers-not-connected-to-internet.html?_r=1, accessed on 13 Feb 2014

[11] Shepard, Daniel P., et al. "Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks." Proceedings of the ION GNSS Meeting. 2012.

[12] Online, SkyJack: Hacker-drone that can wirelessly hijack & control other drones,

http://rt.com/news/hacker-drone-aircraft-parrot-704/, accessed on 13 Feb 2014

[13] Dennis K. Nilsson, Ulf Larson, "A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure", Journal of Networks, 4 (7) pp. 552-564

[14] Javaid, A.Y.; Weiqing Sun; Devabhaktuni, V.K.; Alam, M., "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," Homeland Security (HST), 2012 IEEE Conference on Technologies for , vol., no., pp.585,590, 13-15 Nov. 2012

[15] Spripada, N.R., Fisher, D.G., "Improved Least Squares Identification," Int. J. Control, Vol. 46, No. 6, 1987, pp. 1889-1913.

[16] Zul Azfar, A.; Hazry, D., "A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control," Signal Processing and its Applications (CSPA), 2011 IEEE 7th International Colloquium on , vol., no., pp.28,32, 4-6 March 2011

[17] George M. Siouris, Missile Guidance and Control Systems, Springer; 2004

[18] Timothy X. Brown and Amita Sethi. 2008. Potential cognitive radio denial-of-service vulnerabilities and protection countermeasures: a multi-dimensional analysis and assessment. Mob. Netw. Appl. 13, 5 (October 2008)

[19] Cowan, C.; Wagle, P.; Pu, C.; Beattie, S.; Walpole, J., "Buffer overflows: attacks and defenses for the vulnerability of the decade," DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings , vol.2, no., pp.119,129 vol.2, 2000

[20] A. Dolgikh, T. Nykodym, V. Skormin, and Z. Birnbaum, "Using Behavioral Modeling And Customized Normalcy Profiles As Protection Against Targeted Cyber-Attacks", Proceedings MMM-ACNS October 17, 2012, St. Petersburg, Russia

[21] Kim, Alan, et al. "Cyber Attack Vulnerabilities Analysis for Unmanned Aerial Vehicles." The American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2012.

[22] Conte, Gianpaolo, and Patrick Doherty. "An integrated UAV navigation system based on aerial image matching." Aerospace Conference, 2008 IEEE.

[23] Mitchell, R.; Chen, I.-R., "Adaptive Intrusion Detection of Malicious Unmanned Air Vehicles Using Behavior Rule Specifications," Systems, Man, and Cybernetics: Systems, IEEE Transactions on , vol.PP, no.99, pp.1,1, 0.

[24] Hazewinkel, Michiel, ed. (2001), "Confidence estimation", Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4