False data injection in Kalman Filters in an aerospace setting; radar distance estimated from ADS-B data with simulated noise

Mark Patrick Roeling*[1]

[1]CDT Cybersecurity, Department of Computer Science, University of Oxford

June 23, 2016

Miniproject 1. May-June 2016
Scripts can be retrieved from `https://github.com/mproeling/Kalman-Filter`

Contents

*mark.roeling@wolfson.ox.ac.uk

1 Introduction

Kalman Filters (KF) are recursive state estimation algorithms capable of combining and weighting different variables to estimate the real latent state of a system (Kalman, 1960). In this context, recursive reflects the property that not all previous data has to be kept in storage but every iteration incorporates information from previous observations and predictions (Maybeck, 1979). This made the KF widely applicable resulting in its implementation across various settings, including aerospace, submarines, and the estimation of missile trajectories (Grewal & Andrews, 2010). Given the importance and KFs across settings and systems there is growing interest from security researchers to understand the robustness of KFs under different adversarial models.

2 Related work

In the broader context of machine learning, taxonomies have been proposed to categorize attacks on learning algorithms (Barreno, Nelson, Joseph, & Tygar, 2010). According to those taxonomies, false data injection can be classified as a causative attack where the attackers aim to influence the learning process by affecting the training data. Several studies have addressed the effects of false data injection, mainly in the context of cyber physical systems such as power systems (see Yang et al., 2014), network coordinate systems (Chan-Tin, Feldman, Hopper, & Kim, 2009), and spam filters (Lowd & Meek, 2005; Huang et al., 2011). Although estimating effective attack vectors for the measurement * state matrix is computationally intensive, brute force attacks are still feasible. However, one option is to increase the resilience of the system by relaxing the constraints on brute force attacks, and installing redundant measurement sensors (Mo & Sinopoli, 2010). Also, if the attacker knows the input data and the system, this could allow him to add a vector to the original measurement $z_a = z + x$ instead of true measurement z. Resulting in the attack vector to become a linear combination of the vectors of the (column vectors of the) measurement * state matrix, letting the L_2 norm of the measurement residual of z_a equal z, passing the detection (Lui, Ning, & Reiter, 2009). Other suggestions that help preventing false data injection including schemes to protect measurements (Bobba, Rogers, Wang, Khurana, Nahrstedt, & Overbye, 2010) and detect the attack (Kosut, Jia, Thomas, & Tong, 2010; Pasqualetti, Carli, & Bullo, 2011), as well as algorithms to select the optimal subset of measurements to protect (e.g. through encryption; Kim & Poor, 2011).

A typical method to investigate the robustness of state estimation systems and the capacity of detection methods is the Frog Boiling method. This method words by gradually and episodically injecting data to attack the system, in order not to be detected. A study by Chan-Tin and colleagues (2009) showed that in network coordinate systems, the frog boiling attack was just as effective as a random attack, leading to the assumption that KFs will not be effective outlier detectors. This assumption was tested by Mo and Sinopoli in 2010 who

provided proof that the KF estimates could indeed be destabilized with false data injection, despite several failure detectors.

False data injection in KFs has been studied in the context of SCADA systems (Yang, Chang, & Yu, 2013) and secure estimation methods on simulated UAV data (Chang, Hu, & Tomlin, 2013). Yang and colleagues (2013) investigated the robustness of state estimates by evaluating an innovation factor in five attack models; maximum magnitude-based, wave-based, positive deviation, negative deviation, and mixed. In the maximum magnitude-based attack the adversary tries to achieve the maximum deviation of original measurements that equals to the maximum magnitude of the attack vector. In the wave-based attack, the malicious measurements are the reverse direction of injected attack data. In the positive and negative deviation attack, the adversary tends to achieve the maximum deviation of original measurements along with the direction of increase. Finally, the mixed attack can be a combination of the latter four attack models in consecutive time points (e.g. positive deviation at $t+1$, wave-based at $t+2$). Chang et al. (2013) combined the KF with secure estimation and showed that applying the KF after data were run through a secure estimation algorithm yielded more secure output than applying the algorithm or filter alone. Finally, one of the reasons why the KF itself was not robust against attacks was that the manipulated data violated the KF assumption of Gaussian distributed noise. This observation adds to the overall impression that KFs are inherently insecure and vulnerable to data manipulation.

Aircraft position estimation has historically relied on the availability and interpretation of radar data. Recently, a new system has been developed called Original Automatic Dependent Surveillance - Broadcast (ADS-B) that is to replace primary and secondary surveillance radar technologies by 2017. ADS-B is based on the Global Navigation Satellite System (GNSS) and relies on onboard navigation systems that retrieve GPS data, determine the aircraft position, and forward these data to ground stations[1]. Researchers (e.g., Strohmeier, Lenders, & Martinovic, 2013) and hackers (e.g., Haines, 2012) have already identified several vulnerabilities in the ADS-B infrastructure. The main problem is the absence of encryption of ADS-B message content, resulting in the possibility that adversaries can eavesdrop on messages sent out by aircraft. Other vulnerabilities include the injection of ADS-B messages to create ghost aircraft, jam the signal to make aircraft disappear, or replace aircraft by replacing the identifier of the ADS-B message with modified data. While the technical details of the on-board aircraft position estimation in the ADS-B infrastructure are difficult to come by, integration and combination of raw satellite data to derive an accurate GPS position is likely based on the Kalman Filter. Also Kalman Filters can be used for the combination of ADS-B with radar data (Dunstone, 2014).

Given the outlined vulnerabilities of both the Kalman Filter algorithm and the ADS-B infrastructure, this study aims to investigate the effects of false data injection in the Kalman Filter (both linear and non-linear), by replicating the

[1] https://en.wikipedia.org/wiki/Automatic_dependent_surveillance_-_broadcast

Figure 1: Plotted flightpath of flight OHY925 based on the ADS-B GPS data.

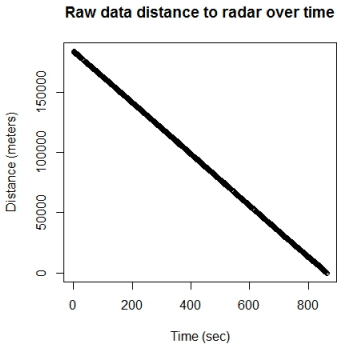
study of Yang et al. (2013) in ADS-B data.

3 Methods

OpenSky ADS-B data

Original Automatic Dependent Surveillance - Broadcast (ADS-B) data were obtained through the OpenSky platform (Strohmeier, Martinovic, Fuchs, Schfer, & Lenders, 2015). In short, OpenSky consists of various off-the-shelf sensors distributed over Europe capturing more than 40% of Europes commercial air traffic. Aircraft use on-board satellite navigation systems (e.g. GPS) to retrieve their own position and velocity, which is broadcasted twice per second to ATC stations on the ground and other aircraft. Exploration of the data from a subset of flights revealed that the data were already filtered upon collection as most commercial GPS systems have built in Kalman Filtering (Labbe, 2015). Filtering ADS-B output again is unlikely to be beneficial, because it violates the time-independency assumption of KFs: GPS data are time dependent as the filter bases its current estimate on the recursive estimates of all previous measurements. For this project, a representative subset of data from flight OHY925 from Antalya (Turkey) to Amsterdam (The Netherlands; see Figure 1) were used.

(a) Figure 2a



(b) Figure 2b

3.1 Linear unidimensional model

Transformation of ADS-B to radar data

Latitude and longitude were included in the ADS-D data. Latitude of a point on the Earth's surface is the angle between the equatorial plane and the straight line that passes through that point and through (or close to) the centre of the Earth. Longitude of a point on the Earth's surface is the angle east or west from a reference meridian to another meridian that passes through that point[2]. For the linear model, I decided to transform the GPS data to position data, allowing simulation of radar distance signals. For every timestep in the ADS-B data, I calculated the great-circle distance, which is the shortest distance between two points on the surface of a sphere, with the Vincenty method (Vincenty, 1975) as implemented in the R package Geopshere (Hijmans, 2015). Because the flight occurred above central Europe, reference measures were used from the European Terrestrial Reference System 1989 (ETRS89). ETRS89 is an earth-centered, earth-fixed geodetic Cartesian reference frame, in which the Eurasian Plate as a whole is static. The equatorial axis of ellipsoid is 6378137, the polar axis of ellipsoid is 6356752.31414, and the inverse flattening of ellipsoid = 298.257222101.

The standard Kalman Filter can only be applied to linear states. To simulate a linear radar system, a subset of the first 500 ADS-B coordinates was selected that consisted of the flightpath between east Aue (Germany) and Flinsberg (Germany; Figure 2a). The last point of the ADS-B flight data was set as radar station (longitude = 10.24911, latitude = 51.31821). With the Vincenty method, the metric distance was calculated for every ABS-B data point and the virtual radar station. The largest distance (beginning of the flightpath) to the radar station was 184092.35 meters (m), the smallest distance (end of the flightpath) to the radar station was 0.28 m. Normally, ADS-B is forwarded twice a second, but the distance between forwards revealed that, assuming a

[2]Wikipedia, Geographic Coordinate System `https://en.wikipedia.org/wiki/Geographic_coordinate_system`

Table 1: Parameter definition

$\bar{u}_t$ = Predicted state estimation
A_t = Matrix of n*n that describes how the state evolves from t-1 to t
B_t = Matrix of n*l that describes how the control u_t changes the state
$\overline{\Sigma}_t$ = Predicted process covariance matrix
Q_t = Process noise covariance matrix
K_t = Kalman gain
C_t = m-dimensional measurement matrix
R_t = Measurement noise
u_t = Current state estimation
y_t = real noisy measurement
y_{noise} = observation errors in mechanism (eg. electronic delays)
z_t = Imported measurement
Σ_t = Updated process covariance matrix
I = Identity Matrix
w = Gaussian white noise

constant velocity in flight, the forwarding occurred with large instability. Also, 46.8% of the forwards contained identical GPS information to the previous forward. Given that ADS-B data is forwarded from the aircraft, filtering of the raw satellite data has already been conducted and ADS-B data is very smooth. This supports the decision to simulate additional error. Noise in radar systems can vary from 5 to 300 m. Given the speed of the aircraft in flight (250 m/s at 900 km per hour) noise was simulated by creating a random normal distribution (N = 100.000) with mean zero and a standard deviation of 250 m. At every iteration of the model, one sample was independently drawn from this distribution and added to the raw radar distance.

If the aircraft has a constant velocity (which is to be expected in this part of the flight), the variance around the distances between updates should be minimal. To verify this, I calculated how much steps were identical between every step in GPS coordinates and divided the metric distance between GPS coordinates by the amount of identical steps. As a result, the differences will average out and every update includes a distance that is standardized for time (one second), yielding a linear model in which the aircraft approached the virtual radar every timestep (Figure 2b). Verification showed substantial variability of the distance between updates, following a normal distribution with mean -213.56 m/s and standard deviation (SD) 101.47 m. These distances were used as the velocity of meters/second to model acceleration in the dynamical model. The noise simulation was identical to the linear model.

Kalman Filter model

The Kalman Filter for the linear model followed the structure as outlined in Welch & Bischop (2006) with parameter definition in Table 1, with the pre-

diction step defined as:

$$\bar{u}_t = A_t \bar{u}_{t-1} + B_t u_{t-1} + w_t \tag{1a}$$

$$z_t = C_t y_t + y_{noise} \tag{1b}$$

$$\overline{\Sigma} = A_t \Sigma_{t-1} A_t^T + Q_t \tag{2}$$

And the filtering step defined as:

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + R_t)^{-1} \tag{3}$$

$$u_t = \bar{u}_t + K_t[z_t - C_t \bar{u}_t] \tag{4}$$

$$\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \tag{5}$$

The dynamical model, or state transition function, followed a standard radar-aircraft model with the innovation function $\left(\begin{smallmatrix} 1 & \triangle T \\ 0 & 1 \end{smallmatrix}\right)$ with $\triangle T$ being the timestep (1 second) and the state change estimator $\left(\begin{smallmatrix} \frac{1}{2}\triangle T^2 \\ \triangle T \end{smallmatrix}\right)$ $[acceleration]$ (e.g., Wendel, Schlaile, Trommer, 2001). Labbe (2015) presents the explanation for the innovation function, with state space matrix $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)\left(\begin{smallmatrix} x \\ \dot{x} \end{smallmatrix}\right)$ where x is the position and $\dot{x}$ is the velocity. The F matrix is $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ and the following Taylor series expansion linearises the equation at t:

$$\Phi(t) = e^{F_t} = I + F_t + \frac{(F_t)^2}{2!} + \frac{(F_t)^3}{3!} + ... + \frac{(F_t)^n}{n!} \tag{6}$$

resulting in $F^2 = \left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, so with all higher powers of F equal 0: $\Phi(t) = I + F_t + 0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right) + \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)t = \left(\begin{smallmatrix} 1 & t \\ 1 & 0 \end{smallmatrix}\right)$. Acceleration was defined as the velocity for every $\triangle T$, which could increase and decrease and the value in the model was updated with the velocity. Starting values of the model were taken from the data, position equal to the largest distance from the aircraft to radar (184092.35 m) and velocity equal to the mean of the velocity in the data (-213.56 m/s). Other starting parameters were uncertainty in measurement (500 m), process covariance matrix (position 1km, velocity 300 m/s), and observation error (position 1km, velocity 250 m/s).

There are different methods to formulate the process covariance matrix (Salau, Trierweiler, Secchi, & Marquardt, 2009) varying from a face value definition to the Autocovariance Least-Squares technique (Rajamani, 2007). I decided to use a process noise covariance model that assumes that the acceleration is constant for the duration of each time period (in line with the standardization), but differs for each time period (in line with the variance in distance), and each of these are uncorrelated between time periods (time independence), outlined in Labbe (2015), shortly defined as $Q_t = \left(\begin{smallmatrix} \frac{\triangle T^4}{4} & \frac{\triangle T^3}{2} \\ \frac{\triangle T^3}{2} & \triangle T^2 \end{smallmatrix}\right)[\sigma_v^2]$ where $[\sigma_v^2]$ is the variance of the velocity.

Illustration 1. One of the radar disks of the Air Operations Control Station Nieuw-Milligen[3].

3.2 Non-linear multidimensional model

Adding noise to ADS-B data

For the non-linear model, a subset of the ADS-B GPS data with altitude was selected that consisted of the flightpath between east Flinsberg (Germany) and Schiphol (the Netherlands, Figure 3). Given that ADS-B data are already filtered, noise was added to the data (Figure 4). First, for every GPS coordinate, the longitude estimate was incremented with 0.000001, while keeping latitude identical. The Vincenty Method was used to calculate the pairwise distance on every iteration, which stopped if the distance reached 100 meters. Typically, satellite estimates are accurate, especially within Europe where continental drift effects are minimal. Given the speed of the aircraft (900 km/h = 250 m/s) and the observation that ADS-B normally forwards twice a second, I used 100 meters as noise threshold. Then, the same procedure was repeated, decreasing the longitude until the distance again reached 100 m. Finally, this iterative process was repeated for latitudes in both directions, while keeping the longitude identical. This resulted in a range for longitude and a range for latitude (Figure 5), that was used to create a list of all possible intermediate GPS values (resolution = 0.00001) from which a random GPS coordinate was

[3] https://www.defensie.nl/binaries/large/content/gallery/defensie/content-afbeeldingen/organisatie/luchtmacht/radar-van-de-luchtmacht-op-air-operations-control-station-nieuw-mill jpg

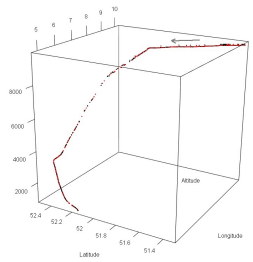


Figure 3: Flightpath (black) and noise (red) starting at the upper right corner. Axes are Longitude, Latitude, and Altitude.

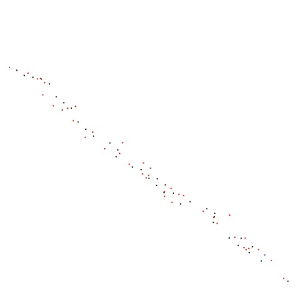


Figure 4: Zoomed-in part of the flightpath, raw data in black, simulated noise in red.

Figure 5: One GPS position (middle) with four 100 m deviations (up, down, left, right) and the yellow noise window.

sampled. This noisy data was used for all subsequent analyses.

Figure 3 shows the flightpath following a non-linear model. As the aircraft slowed down in the descent towards the airport, the mean of the velocity (-122.424 m/s) and with a larger variability (SD = 371.073 m) than the linear flightpath (SD = 101.47 m). These distances were used as the velocity of meters per second.

Extended Kalman Filter model

The Extended Kalman Filter (EKF) is the most commonly used state estimation algorithm for non-linear processes (Nadella, 2015). Following Yang, Chang, and Yu (2013) I used the computation method described in Welch and Bishop (2006), which is largely identical to the non linear Kalman Filter in that is it still defined as a linear model but uses local linearisation to approximate the slope at the point of measurement. This local linearisation occurs in the estimation of the dynamical model, so that the estimated state is system function (f) that takes three parameters $\bar{u}_t = f(\bar{u}_{t-1}, u_{t-1}, w_t)$ with output function $z_t = h(\bar{u}_t, v_k)$, where $\bar{u}_t$ and z_k are the state variable vector and measurable output at time t, respectively. Parameter u_t is the measurable input, w_t is the process noise (White Gaussian), and v_k is the measurement noise (White Gaussian). Calculation of the Jacobians was conducted with the package numDeriv[4] available in R.

The dynamical model that has to be linearised was identical to the model used in the linear model but extended to six variables. Giving the following innovation function $\begin{pmatrix} 1 & 0 & 0 & \triangle T & 0 & 0 \\ 0 & 1 & 0 & 0 & \triangle T & 0 \\ 0 & 0 & 1 & 0 & 0 & \triangle T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} longitude \\ latitude \\ altitude \\ velocity_{longitude} \\ velocity_{latitude} \\ velocity_{altitude} \end{pmatrix}$ and state change estimator

[4] https://cran.r-project.org/web/packages/numDeriv/numDeriv.pdf

$$\begin{pmatrix} \frac{1}{2}\triangle T^2 & 0 & 0 \\ 0 & \frac{1}{2}\triangle T^2 & 0 \\ 0 & 0 & \frac{1}{2}\triangle T^2 \\ \triangle T & 0 & 0 \\ 0 & \triangle T & 0 \\ 0 & 0 & \triangle T \end{pmatrix} \begin{pmatrix} accelaration_{longitude} \\ accelaration_{latitude} \\ accelaration_{altitude} \end{pmatrix}.$$

The extended kalman filter formulation is as follows:
The prediction step:

$$\bar{u}_t = f(\bar{u}_{t-1}, u_{t-1}, w_t) \tag{7a}$$
$$z_t = h(y_t, u_{t-1}, y_{noise}) \tag{7b}$$

$$\overline{\Sigma} = A_t \Sigma_{t-1} A_t^T + L_{t-1} Q_t L_{t-1}^T \tag{8}$$

Where A_t = and L_{t-1} = and the filtering step is defined as:

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + M_t R_t M_t^T)^{-1} \tag{9}$$

$$u_t = \bar{u}_t + K_t[z_t - h(\bar{u}_t, 0)] \tag{10}$$

$$\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \tag{11}$$

Where $C_t = \frac{\partial h}{\partial x}(\bar{u}_t, 0)$ and $M_t = \frac{\partial h}{\partial v}(\bar{u}_t, 0)$.

Starting parameters were uncertainty in measurement (1 km, equal to a bidirectional deviation of 0.0015 longitude and 0.0010 latitude), process covariance matrix (500 m with start error of 300 m/s), and observation error (position 1 km, velocity 250 m/s).

3.3 Attack models

Anomaly detection in Kalman Filter

This method follows the paper from Yang, Chang, and Yu (2013). After each prediction step the innovation factor (v_t) is calculated, which is equal to the difference between the prediction and the actual measurement:

$$v_t = z_t - C_t \bar{u}_t \tag{12}$$

With z_t being the original measurement and $C_t \bar{u}_t$ being the predicted state. The innovation factor can be approximated by a white Gaussian process. To enhance interpretation of the innovation factor, it is standardized:

$$\lambda_t = v_t / \rho_t \tag{13}$$

$$\rho_t = \sqrt{(C_t \overline{\Sigma}_t C_t^T + R_t)} \tag{14}$$

Where C_t is a m-dimensional measurement matrix, $\overline{\Sigma}_t$ is the updated process covariance matrix, and R_t is the measurement noise. A detailed description of the steps involved in anomaly detection is published elsewhere (Yang, Chang, and Yu, 2013). In short, anomalies are detected by comparing the absolute

value of the standardized innovation factor $|\lambda_t|$ against a predefined threshold λ_{max}. Given the two tailed distribution of the standardized innovation factor.

Sophisticated data attacks use an effective non-zero attack vector c_t, in the anomaly detection algorithm: $\frac{z_t - C_t \bar{u}_t}{\rho_t} \leq \lambda_{max}$ so that the range of z_t van be obtained by:

$$C_t \bar{u}_t + \lambda_{max} \rho_t \geq z_t \geq C_t \bar{u}_t - \lambda_{max} \rho_t \tag{15}$$

In other words, the malicious measurement z_t should be a value that is derived from the boundaries depending on the measured state, the predicted state, and the (standardized) innovation factor threshold, in order not to be detected by the anomaly detection threshold. The attack vector c_t can be obtained by subtracting y_t (observed noisy estimate) from z_t (predicted measurement). I assume the attacker knows the anomaly detection algorithm and the predefined threshold λ_{max}. Other parameters ρ_t and $C_t \bar{u}_t$ can be derived between $t-1$ and t. Since the state prediction is conducted at the very beginning of the KF procedure, and adopts the value of the previous state estimation after the first iteration, z_t can be derived as soon as the previous iteration is completed, which is before t.

Maximum magnitude-based attack

In this attack, the adversary tries to achieve the maximum deviation of the original measure. That is, the maximum deviation that is allowed within the anomaly detection threshold, by estimating the maximum attack vector $|c_t|$ that achieves the maximum manipulation of the received measurement z_t from the original measurement y_t by inserting false data. The adversary acquires the parameters at time $t-1$, computes the predicted measurement h_i, λ_{max}, and ρ_t. For the next timestep $(t+1)$, the original measurement y_t is retrieved and the innovation vector v_t is calculated. Depending on the evaluation of the innovation vector, $\lambda_{max} \rho_t$ is added to $(v_t < 0)$ or subtracted from $(v_t \geq 0)$ h_i. This attack can be expressed as follows:

$$\text{if } v_t \geq 0 : h(\bar{u}_t, 0) - \lambda_{max} \rho_t \tag{16a}$$

$$\text{if } v_t < 0 : h(\bar{u}_t, 0) + \lambda_{max} \rho_t \tag{16b}$$

With attack vector c_t:

$$\text{if } v_t \geq 0 : c_t = z_t - y_t = -v_t - \lambda_{max} \rho_t \tag{17a}$$

$$\text{if } v_t < 0 : c_t = z_t - y_t = -v_t + \lambda_{max} \rho_t \tag{17b}$$

Giving:

$$|c_t| = |z_t - y_t| = |v_t| + \lambda_{max} \rho_t \tag{18}$$

Wave-based attack

This attack is computationally identical to the maximum magnitude-based attack, but the injected attack data will be in the opposite direction of the estimated state. This translated into the formulas below, with opposite conditions

on v_t:

$$\text{if } v_t < 0 : h(\bar{u}_t, 0) - \lambda_{max}\rho_t \tag{19a}$$
$$\text{if } v_t \geq 0 : h(\bar{u}_t, 0) + \lambda_{max}\rho_t \tag{19b}$$

With attack vector c_t:

$$\text{if } v_t < 0 : c_t = z_t - y_t = -v_t - \lambda_{max}\rho_t \tag{20a}$$
$$\text{if } v_t \geq 0 : c_t = z_t - y_t = -v_t + \lambda_{max}\rho_t \tag{20b}$$

Giving:

$$|c_t| = |z_t - y_t| = \lambda_{max}\rho_t - |v_t| \tag{21}$$

Positive deviation attack

In aim of this attack is to achieve the maximum deviation (maximum value of z_k) of original measurements along with the direction of increase, independent of the direction of the innovation factor. This attack can be formulated as:

$$z_k = h(\bar{u}_t, 0) + \lambda_{max}\rho_t \tag{22}$$

With attack vector c_t:

$$c_t = z_t - y_t = -v_t + \lambda_{max}\rho_t \tag{23}$$

Giving:

$$|c_t| = |z_t - y_t| = \lambda_{max}\rho_t - v_t \tag{24}$$

Negative Deviation attack

The negative deviation attack is identical to the positive deviation attack, but here, z_t is always the minimum of the range of its possible value:

$$z_k = h(\bar{u}_t, 0) - \lambda_{max}\rho_t \tag{25}$$

With attack vector c_t:

$$c_t = z_t - y_t = -v_t - \lambda_{max}\rho_t \tag{26}$$

Giving:

$$|c_t| = |z_t - y_t| = \lambda_{max}\rho_t + v_t \tag{27}$$

3.4 State deviation under attack

Linear model

This section explains how and where in the KF procedure, data are injected. This study assumed that attacks had full knowledge about the system, the incoming data, the anomaly detection algorithm, and the implementation of the state estimation algorithm(s). The Kalman filter is defined below:
The linear model:

$$\bar{u}_t = A_t\bar{u}_{t-1} + B_t u_{t-1} + w_t \tag{28}$$
$$\overline{\Sigma} = A_t\Sigma_{t-1}A_t^T + Q_t \tag{29}$$

And the filtering step defined as:

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + R_t)^{-1} \tag{30}$$

$$u_t = \bar{u}_t + K_t[z_t - C_t \bar{u}_t] \tag{31}$$

$$\Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \tag{32}$$

The attack vector was defined previously as $c_t = z_t - y_t$, and the attack vector errors are obtained by multiplying them with the Kalman Gain:

$$a_t = K_t c_t \tag{33}$$

Typically, the attack vector and its errors are matrices $m * 1$ matrices, with m being the number of variables or dimensions in the model, necessitating the definition of an attack parameter c_t for every dimension (the errors a_t can be derived from the Kalman gain and attack vector). These attack parameters are added to the state estimation model parameter $\bar{u}_t$. Hence, both the observed measurement y_t and the predicted state $\bar{u}_t$ are respectively manipulated by the attack vector c_t and a_t:

$$z_{t+1} = y_{t+1} + c_{t+1} \tag{34a}$$

$$\bar{u}_t^+ = \bar{u}_t + a_t \tag{34b}$$

The attack vectors c_t and a_t are injected in the state estimation formula, resulting in a manipulated state estimation:

$$u_{t+1}^+ = [A_t(\bar{u}_{t-1} + a_t) + B_t u_{t-1} + w_t] + K_{t+1}[(y_{t+1} + c_t) - C_t[A_t(\bar{u}_{t-1} + a_t) + B_t u_{t-1} + w_t]] \tag{35}$$

Equal to:

$$u_{t+1}^+ = \bar{u}_t^+ K_{t+1}[z_{t+1} - C_t \bar{u}_t^+] \tag{36}$$

Where u_t^+ is the state estimation after the attack. Since the state estimation is defined or acknowledged to be the moment t, the attack occurs between $t-1$ and t.

Non-linear model

Data injection is largely identical in the non-linear model, with a_t identical to formula 33. Consequently, the state estimation model is as follows:

$$u_{t+1}^+ = f(\bar{u}_t^+, 0) + K_{t+1}[z_{t+1} - h(f(u_t^+, 0), 0)] \tag{37}$$

Where z_{t+1} is the received measure at $t+1$ and $z_{t+1} = y_{t+1} + c_{t+1}$.

3.5 Evaluating KF performance

There are numerous ways to investigate the performance of the KF. In general, relying on visual inspection of the plotted KF estimation can be intuitive but is not always valid. Labbe (2015) describes how to use check the KF residuals and compare these residuals against 95% confidence intervals. To understand

the accuracy of the parameters in the KF model, there is a widely used performance index (J_t; see Yang, Chang, and Yu (2013) that evaluates the ratio of [estimated measurement - true vector of measurements] versus [real (noisy) measurement - true vector of measurement]. Ideally, the ratio approaches unity, as an indication of optimal performance:

$$J_t = \frac{\Sigma|u_t - \bar{u}_t|}{\Sigma|y_t - \bar{u}_t|} \tag{38}$$

4 Results

4.1 Linear model

The normal (not attacked) linear model is visually presented in Figure 6, which shows a very good fit of the Kalman Filter on the data.

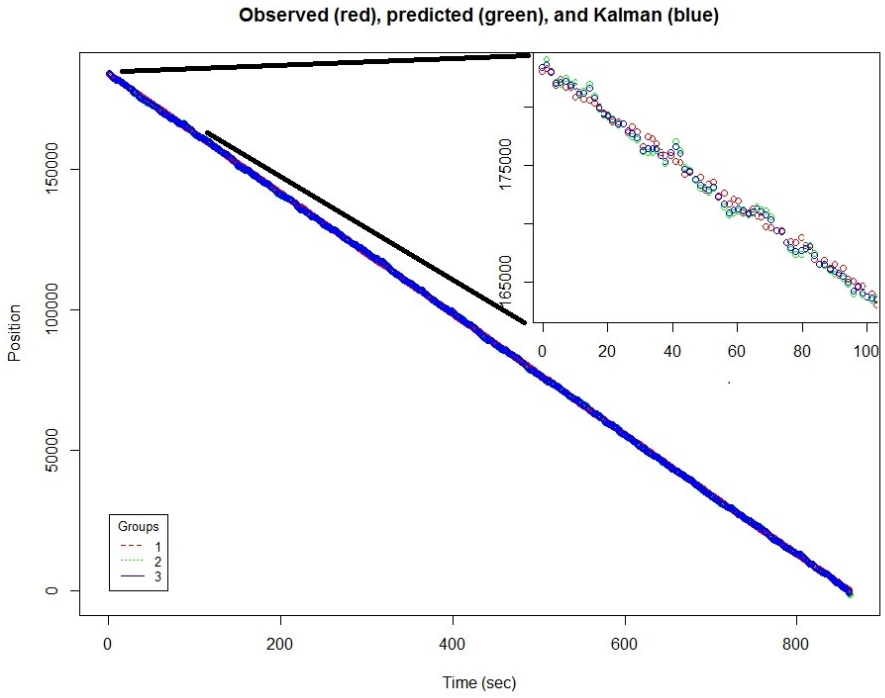
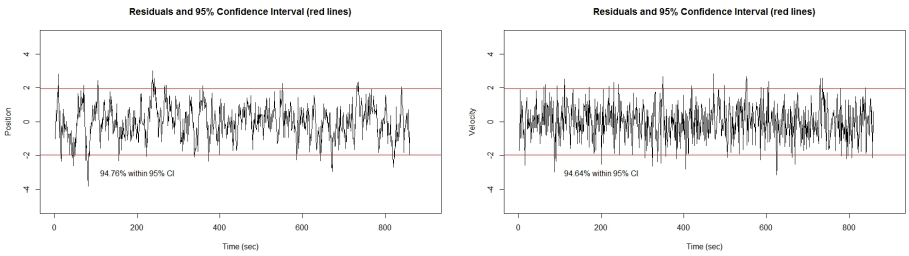


Figure 6. Plot of the linear model noisy data (red), the predicted state (green) and the current state Kalman Filter (blue).

Evidence for good fit of the data can also be inferred from the residuals, where 94.76% of the position estimates (Figure 7a) and 94.64% of the velocity estimates (Figure 7b) fall within the 95% confidence interval.

Figure 8 presents (a part of) the trajectory plots for the four attack models. The maximum magnitude based attack shows considerable deviation from the



(a) Figure 7a. Residuals of the position.

(b) Figure 7b. Residuals of the velocity.

trajectory, especially when compared to the wave based (opposite direction of the prediction), where deviation of the predicted state and current state (Kalman) look consistently smaller, owing to the fact that they are in the oppo-

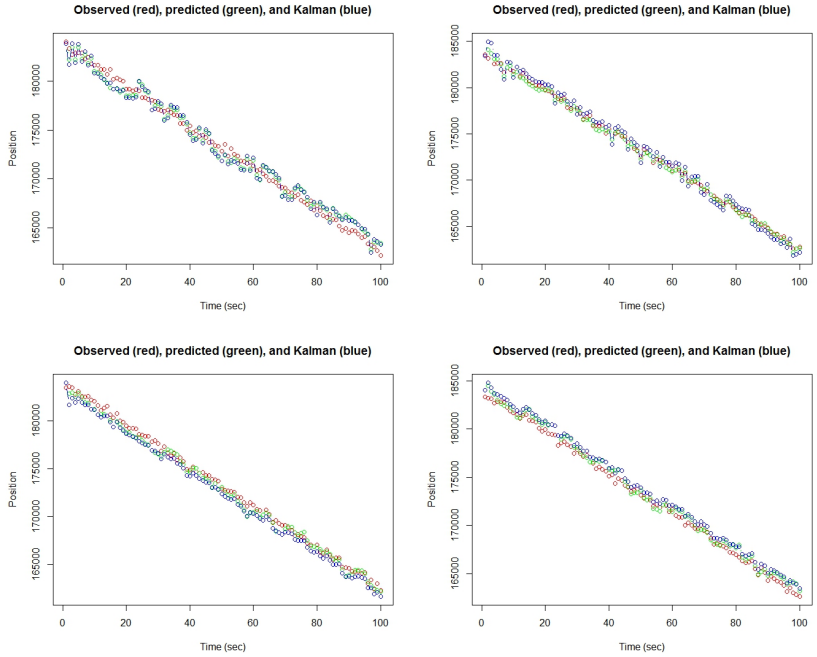


Figure 8. Scatterplots of the noisy data (red), predicted state (green) and Kalman state estimate (blue), for the maximum magnitude (upper left), wave-based (upper right), positive deviation (lower left) and negative deviation (lower right) attacks.

site direction, and therefore could level out strong deviations in the real (noisy) state. The positive and negative deviation attacks show similar patterns, with the positive deviation yielding a uniform deviation to lower positions; pulling the Kalman estimate of position closer to the radar. In contrast, the negative

deviation attack provides a persistent overestimation of the position of the aircraft.

Figure 9 displays the performance estimates of the different models and reveals, although the effect is small, that the positive deviation attack and maximum magnitude data injection provide the worst performance from the Kalman Filter in a linear model. Whereas the negative deviation and wave based attack model have less impact on model performance.

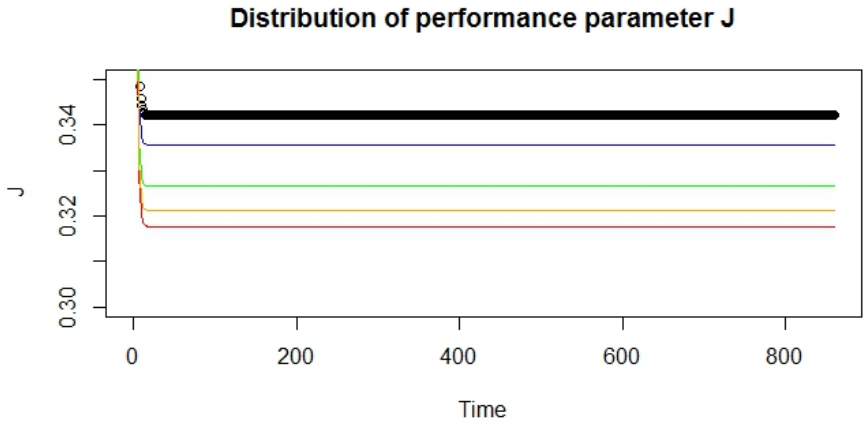
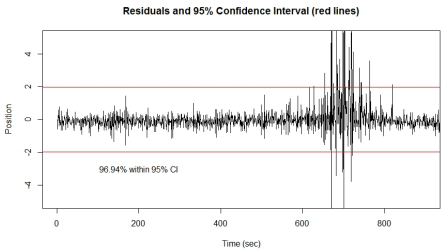


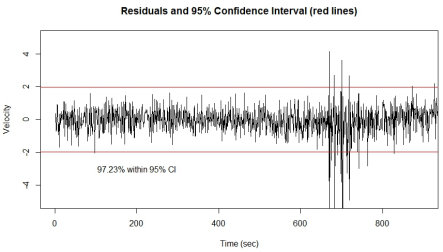
Figure 9. Plot of the performance index J for all timepoints, for the model without attack (black), maximum magnitude attack (orange), wave based attack (blue), positive deviation (red), and negative deviation (green) attack.

4.2 Non-linear model

The baseline (not attacked) non-linear multidimensional model is visually presented in Figure 11, which shows the trajectory of the aircraft, coming in at an altitude of 10 km, slowly decreasing in altitude for landing. During it's descent, there are several manoeuvres, mainly to the left (please note that the ADS-B data used here is not forwarded twice a second resulting in a pattern that is not really time-scaled). We can clearly see that in stable flight, there is a very good convergence of the predicted state and current (Kalman) estimates. When the aircraft decreases in altitude (and velocity; not in model) the predicted estimates slightly diverge, but the Kalman state estimation remains relatively close to the measured position. Especially during sudden manoeuvres, the residuals (Figure 9a-c) show slight model divergence, which is resolved after a 20-50 seconds, illustrating typical Kalman Filter behavior. In normal

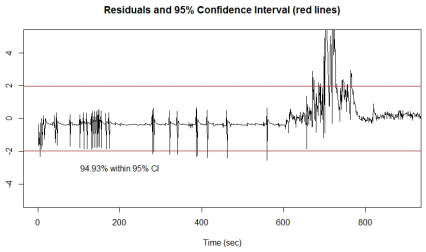


(a) Figure 10a. Residuals of longitude.



(b) Figure 10b. Residuals of latitude.

flight, the residuals remain well within the 95% Confidence Interval boundaries, but the signal crosses the boundaries during fast and unpredicted changes of longitude, latitude, and altitude of the aircraft. These kinds of deviations in multidimensional systems are well known (Labbe, 2015).



(c) Figure 10c. Residuals of altitude.

Figures 12-15 present the trajectory plots for flight under data injection (please see the Appendix for larger versions). Again, the maximum magnitude based attack (Figure 12) shows considerable deviation from the trajectory, exceeding the deviation of the wave based attack (Figure 13).

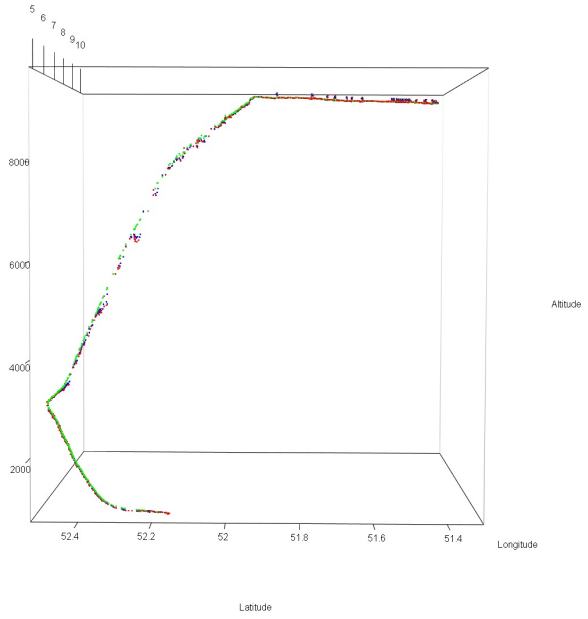


Figure 11. 3d plot of the flight trajectory without data injection, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.

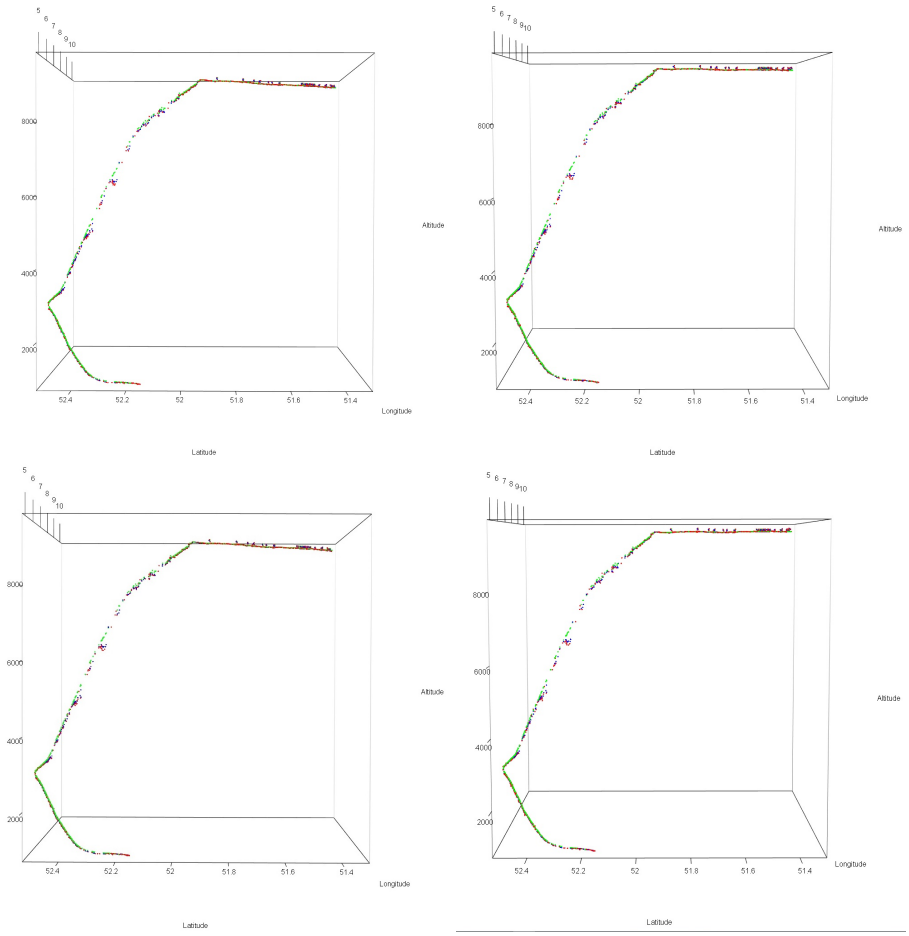


Figure 12. 3d plots of the flight trajectory under maximum magnitude (upper left), wave based (upper right), positive deviation (lower left), and negative deviation (lower right) attacks, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.

The performance indices for the different attack models are presented in Figures 13a and b. Which both show strong differences between the different attack models. In short, the positive deviation attack and maximum magnitude data injection result in the worst deviation from the baseline (no attack) model. The negative deviation and wave based attack model have less impact on model performance.

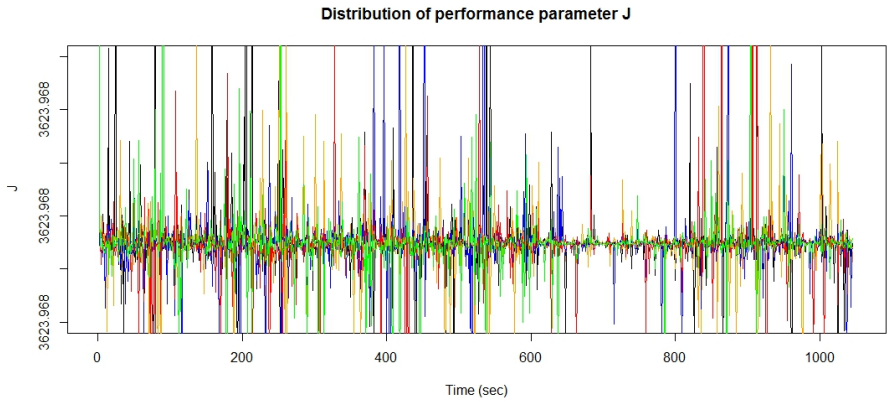


Figure 13a. Performance estimates for longitude for the various attack models (normal = black, maximum magnitude = orange, wave based = blue, positive deviation = red, and negative deviation = green).

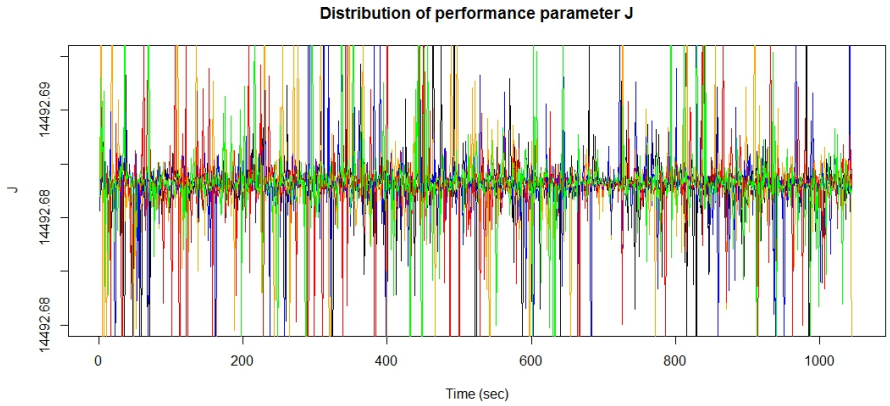


Figure 13b. Performance estimates for latitude for the various attack models (normal = black, maximum magnitude = orange, wave based = blue, positive deviation = red, and negative deviation = green).

5 Discussion

The overarching aim of this study was to test the effects of false data injection on state estimation with Kalman Filters in an aerospace setting. I replicated the findings from Yang, Chang, and Yu (2013), confirming that the wave-based attack had the least impact. This finding may be influenced by the variability in the data, as the wave based attack is an injection of signal in the opposite direction of the innovation factor. Although the innovation factor follows a normal distribution with zero mean, strong deviations from the expected result could be counterbalanced by data injection in the wave based attack. In

contrast, attacks that manipulate the signal to the furthest possible (within the anomaly detection threshold λ_{max}) extremes have more impact on state estimation, especially when the injection is in the same direction as the predicted state. Kalman filter behavior was in line with descriptions and reports from others (Jwo et al., 2009; Wendel, Schlaile, & Trommer, 2001; Romaniuk & Gosiewski, 2014; Labbe, 2015). Arguably, one of the reasons why the KF and EKF models fitted provided a good fit to the data was that the data, even in the non linear model, did not contain very large non-linear patterns. The fit of the multidimensional model could be improved by weighting the individual velocity estimates in the dynamical model for the speed estimate forwarded from the aircraft, which was not done in this study because of the apparent complexity of modelling speed in miles per hour and longitude/latitude estimates. Indeed, in sudden manoeuvres the EKF diverged, only to return to normal after a few seconds. This EKF divergence problem could intensify if data would be highly non-linear (Bizup & Brown, 2003; Jansberg, 2010). I am aware of the advantages for model convergence and computational stability of other filters like the unscented KF or the enhanced EKF, but current evidence does not show that these versions of the KF algorithm are more stable or secure when attacked with false data (see Yang, Chang, and Yu, 2013).

The addition of noise to the ADS-B GPS coordinates may have provided an overestimation of the variation in ADS-B data, causing the innovation factor distribution to be too wide. In data with small variability, the anomaly detection threshold could be more stringent because large deviations are not expected, allowing early detection of deviations (indicating attacks). However, even without noise added to the data, the anomaly detection window will likely to be large because it has to allow manoeuvres from the aircraft. Although velocity and altitude are major predictors for the probability that an aircraft makes a rapid manoeuvre. The anomaly detection threshold is assumed to be static, meaning it does not depend on these variables. Therefore, λ_{max} has to present an equilibrium that allows manoeuvring of the aircraft within its flight envelope, but stringent enough to detect anomalies. Consequently, it is likely that the frog boiling attack would still have been successful, even without simulated noise, because λ_{max} allows significant deviation from the predicted signal.

This work also confirms that the frog boiling method can be successfully used to attack state estimation systems, even with anomaly detection. It is unclear whether state estimation systems are currently equipped with anomaly detection algorithms. However, the severity of the attack (as modelled here) is limited by the innovation factor v_t calculated from the measured state z_t and the predicted state $\bar{u}_t$. As the measured state is updated on every iteration of the model and is not infinite in its error (given the requirement of Gaussian noise), the effect of the frog boiling attack could be smaller than observed by others (Chan-Tin et al., 2009), since its impact is bounded by the anomaly detection system. This could also explain why our model deviated from its original state, but did not diverge as a result of the attack, which contrast the results from (Mo & Sinopoli, 2010). Moreover, the innovation vector v_t is es-

timated for every variable that contributes to the state estimation. This study used the same standardized innovation vector threshold λ_{max} across variables, but variable-specific thresholds could easily be implemented to make the system more robust against attacks. Nevertheless, frog boiling remains successful, even with an anomaly detection system implemented.

Countermeasures

Following the apparent vulnerabilities of the KF, studies have proposed several countermeasures to mitigate or reduce the impact of false data attacks. Yang, Chang, and Yu (2013) proposed to multiply the measurement noise with an exponential, which leads to a decrease of the Kalman Gain so that it favours prediction over measurement. They also proposed temporal-based detection, using the nonparametric cumulative sum (CUSUM) algorithms to detect change in the observations as early as possible. Another proposed countermeasure resembles more general machine learning anomaly detection and involves comparing state estimates to distributions based on historical data (Yang et al., 2014).

Attack models

One of the limitations of the models used in this project (and others) could be the assumptions made about the attacker. This study assumed that attackers had full knowledge about the system, the incoming data, the anomaly detection algorithm, and the implementation of the state estimation algorithm(s). It is unclear to what extend these assumptions are valid. Maybe these assumptions stem from cryptography, a field where it is common practice to assume the attacker has knowledge about the technical and computational details of the cryptographic protocols (Schneier, 1996). It is difficult to prove these assumptions are wrong, but it could be a good idea to formulate general rules and best practice guidelines that can be used to formulate attack model assumptions. This could aid generalization of theoretical problems. Also, however worrisome the effects of false data injection are, the exact implications of false data injection attacks are unknown. In industrial systems, training data are often not available to attackers and the data-driven thresholds used in detection system (e.g., weights of words in spam filters) are not continuously updated with every new email (datapoint) but based on large amounts of historical data that have been screened intensively. Also, most learning processes are inherently robust against direct data injection attacks. Given the amount of data that is typically used in industrial data driven detection algorithms it is almost impossible for a single attacker to immediately change the underlying distribution of a detection algorithm. To prevent rejection of the injection as an outlier, one has to model the data injection careful to allow subtle deviation of the original signal. Hence, the outcomes of this project favour the use of data-driven security thresholds in state estimation systems. With data-driven thresholds, the best achievable scenario could be one where the variation of the distribution increases, broadening the boundaries for attack messages that will ultimately fall within the distribution.

6 Conclusion

This study reports the effects of false data injection on ADS-B derived position estimates of aircraft position. Data were injected in a linear model (Kalman Filter), investigating the change of radar-distance position, and in a non-linear model (Extended Kalman Filter) with the ADS-B GPS coordinates with simulated noise. For both models, the positive deviation attack and maximum magnitude data injection provide the worst performance of the filtered state estimation model. Whereas the negative deviation and wave based attack model have less impact on model performance.

References

[1] Barreno, M., Nelson, B., Joseph, A.D., & Tygar, J.D. (2010). The security of machine learning. Machine Learning, 81, 121-148.

[2] Bizup, D.F., & Brown, D.E. (2003). The over-extended Kalman filterdon't use it!. In: Proceedings of the Sixth International Conference on Information Fusion, Cairns, Qld., Australia, University of New Mexico, pp227-233.

[3] Bobba, R.B., Rogers, K.M., Wang, Q., Khurana, H., Nahrstedt, K., & Overbye, T.J. (2010). Detecting false data injection attacks on DC state estimation. Proc. Preprints of the First workshop Secure Control Systems.

[4] Chan-Tin, E., Feldman, D., Hopper, N., & Kim, Y. (2009). The frog-boiling attack: limitations of anomaly detection for secure network coordinate systems. Security and Privacy in Communication Networks, 19, 448-458.

[5] Chang, Y.H., Hu, Q., & Tomlin, C.J. (2015). Secure estimation based Kalman Filter for cyber-physical systems against adversarial attacks. Retrieved online February 12th from http://arxiv.org/abs/1512.03853

[6] Dunstone, G. (2014). ADS-B in a radar environment. Retrieved online from http://www.icao.int/APAC/Meetings/2014%20ADSBSITF13/SP06_AUS%20-%20ADSB%20in%20radar%20environments.pdf

[7] Grewal, M.R., & Andrew, A.P. (2010). Applications of Kalman Filtering in Aerospace 1960 to the present. IEEE control systems magazine, June 2010, 69-78.

[8] Haines B. [RenderMan]. (2012, july). DEFCON 20: Hacker + Airplanes = No Good Can Come Of This. Retrieved from https://www.youtube.com/watch?v=CXv1j3GbgLk

[9] Hijmans, R.J. Introduction to the geosphere package. Retrieved online 22nd of May from https://cran.r-project.org/web/packages/geosphere/vignettes/geosphere.pdf

[10] Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I.P., & Tygar, J.D. (2011). Adversarial Machine Learning . Proceedings of the 4th ACM Workshop on Artificial Intelligence and Security, 43-58.

[11] Jansberg, R. (2010). Tracking of an Airplane using EKF and SPF (Master thesis). Retrieved online 14th June from https://www.duo.uio.no/bitstream/handle/10852/10968/TrackingxofxanxairplanexusingxEKFxandxSPFx-xJansberg.pdf?sequence=16&isAllowed=y

[12] Jwo, D-J., Chen, M-Y., Tseng, C-H., & Cho, T-S. (2009). Adaptive and Nonlinear Kalman Filtering for GPS Navigation Processing. In V.M. Moreno and A. Pigazo (Eds.). Kalman Filter: Recent Advances and Applications. Retrieved online from http://cdn.intechopen.com/pdfs/6336.pdf

[13] Kalman, R.E. (1960). A New approach to linear filtering and prediction problems. ASME Trans. J. Basic. Eng. 82, 95-108.

[14] Kim, T.T., & Poor, H.V. (2011). Strategic protection against data injection attacks on power grids. IEEE Transactions on smart grid, vol 2, 2.

[15] Kosut, O., Jia, L., Thomas, R.J., & Tong, L. (2010). On malicious data attacks on power system state estimation. Proc 45th International Univ. Power Eng. Conf. (UPEC 2010).

[16] Labbe, R. (2015). Kalman and Bayesian Filters in Python. Retrieved online 12th of May from https://drive.google.com/file/d/0By_SW19c1BfhTHRXWFJ1RUtvaDQ/view?usp=sharing

[17] Llowd, D., & Meek, C. (2005). Adversarial Learning. KDD.

[18] Lui, Y., Ning, P., & Reiter, M.K. (2009). False data injection attacks against state estimation in electric power grids. Proceedings of the 16th ACM conference on Computer and communications security, 21-32.

[19] Maybeck, P.S. (1979). Stochastic models, estimation, and control. New York: Academic Press.

[20] Mo, Y., & Sinopoli, B. (2010). False data injection attacks in control systems. Proc. Preprints of the first workshop secure control systems.

[21] Nadella, S.D. (2015). Use of Extended Kalman Filter in Estimation of Attitude of a Nano-Satellite International Journal of Electronics and Electrical Engineering, 3(1), 38-43.

[22] Pasqualetti, F., Carli, R., & Bullo, F. (2011). A distributed method for state estimation and false data detection in power networks. Proc. IEEE International Conf. Smart Grid Comm.

[23] Rajamani, M.R. (2007). Data-based techniques to improve state estimation in model predictive control. Retrieved online 8th June from http://jbrwww.che.wisc.edu/theses/rajamani.pdf

[24] Romaniuk, S., & Gosiewski, Z. (2014). Kalman Filter realization for oriental and position estimation on dedicated processor. Retrieved online 12th June from http://www.actawm.pb.edu.pl/volume/vol8no2/06_2014_004_ROMANIUK_GOSIEWSKI.pdf

[25] Salau, N.P.G., Trierweiler, J.O., Secchi, A.R., & Marquardt, W. (2009). A new process noise covariance matrix tuning algorithm for Kalman based state estimators. Advanced Control of Chemical Processes, Volume 7, Part 1, 572-577

[26] Schneier, B. (1996). Applied Cryptography Protocols, Algorithms, and Source Code in C. John Wiley & Sons.

[27] Strohmeier, M. Lenders, V., & Martinovic, I. (2013). Security of ADS-B: State of the art and beyond. 2013, arXiv:1307.3664v1.

[28] Strohmeier, M., Martinovic, I., Fuchs, M., Schfer, M., & Lenders, V. (2015). OpenSky: A Swiss army knife for air traffic security research. IEEE 34th Digital Avionics Systems Conference, 4A1-1 - 4A1-14.

[29] Vincenty, T. (1975). Direct and Inverse solution of geodesics on the ellipsoid with application of nested equations. Survey Review, XXIII (176), 88-93.

[30] Welch, G., & Bishop, G. (2006). An Introduction to the Kalman Filter. Department of Computer Science, University of North Carolina at Chapel Hill, Retrieved online 12th May from https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[31] Wendel, J., Schlaile, C., T., & rommer, G.F. (2001). Direct Kalman Filtering of GPS/INS for Aerospace Applications. International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS2001), Canada.

[32] Yang, Q., Chang, L., & Yu, W. (2013). On false data injection attacks against Kalman filtering in power system dynamic state estimation. Security and Communication Networks, 9(9), 833-849.

[33] Yang, Q., Yang, J., Yu, W., An, D., Zhang, N., & Zhoa, W. (2014). On false data injection against power system state estimation: modeling and countermeasures. IEEE Transactions on parallel and distributed systems, 25(3), 717-729.

7 Appendix

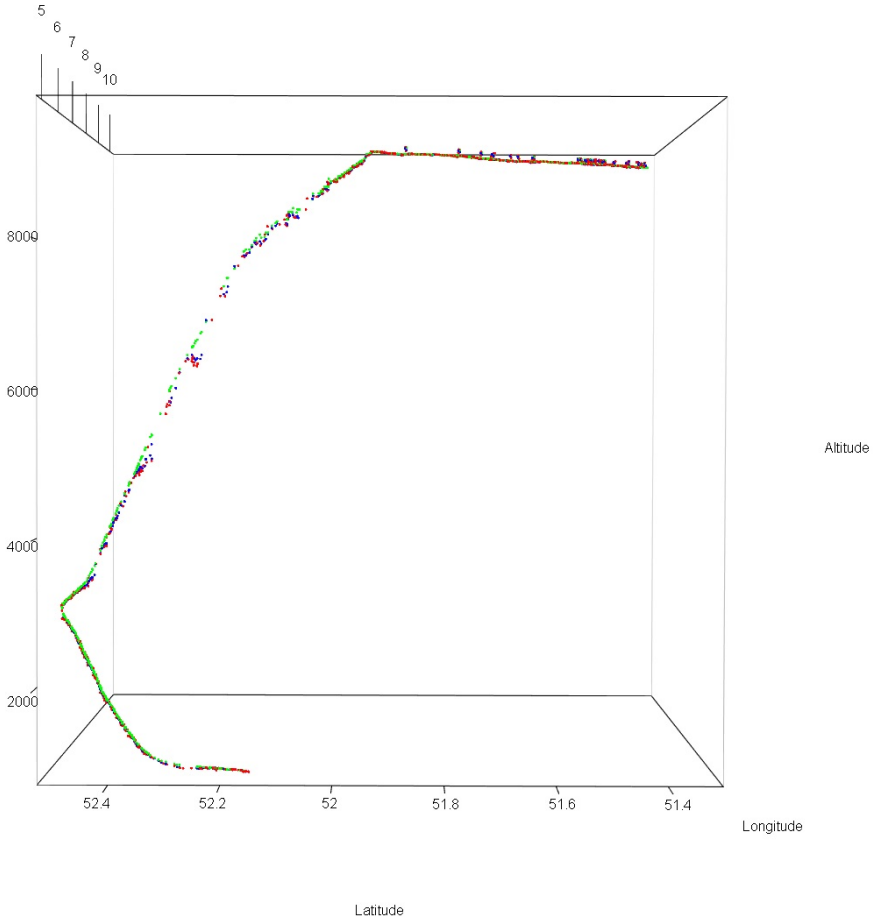


Figure A1. 3d lot of the flight trajectory under maximum magnitude based attack, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.

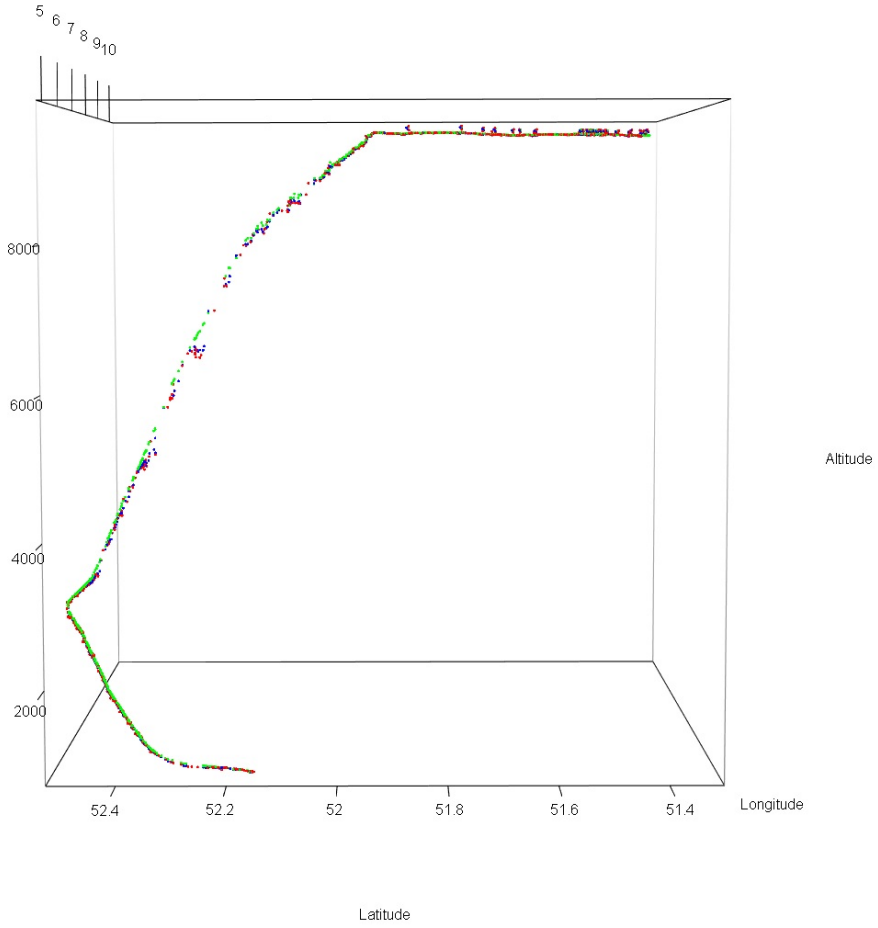


Figure A2. 3d lot of the flight trajectory under wave based attack, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.

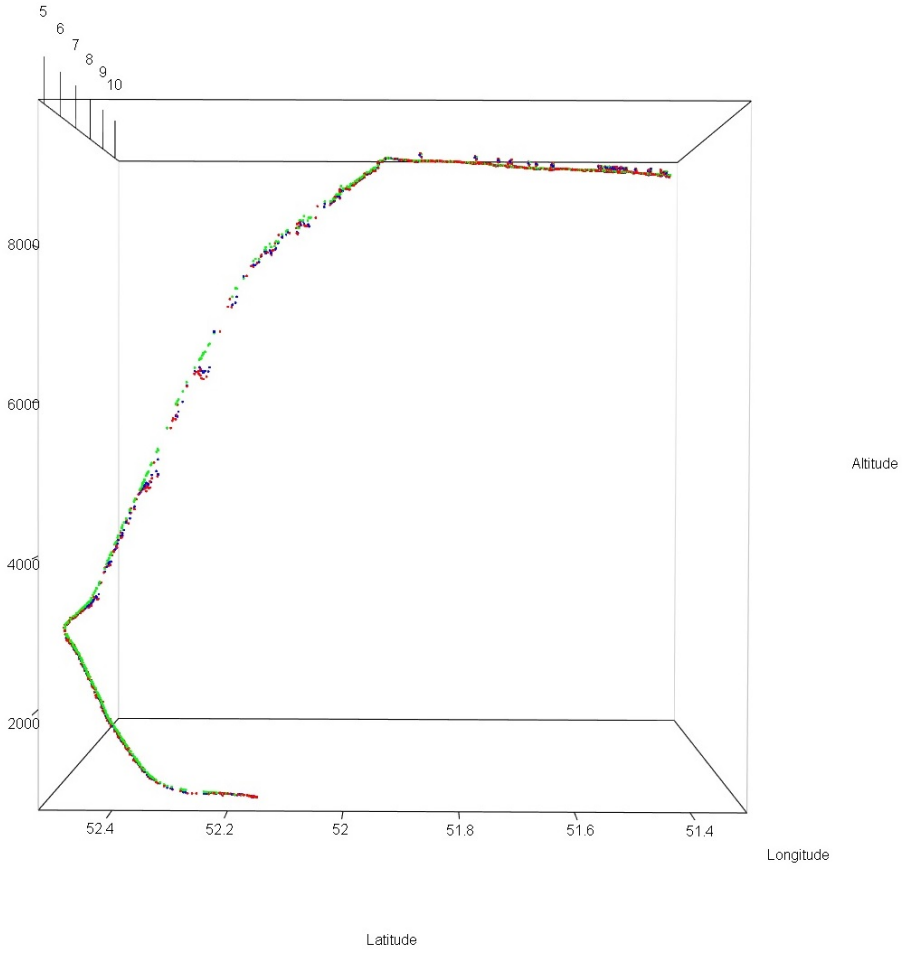


Figure A3. 3d plot of the flight trajectory under positive deviation attack, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.

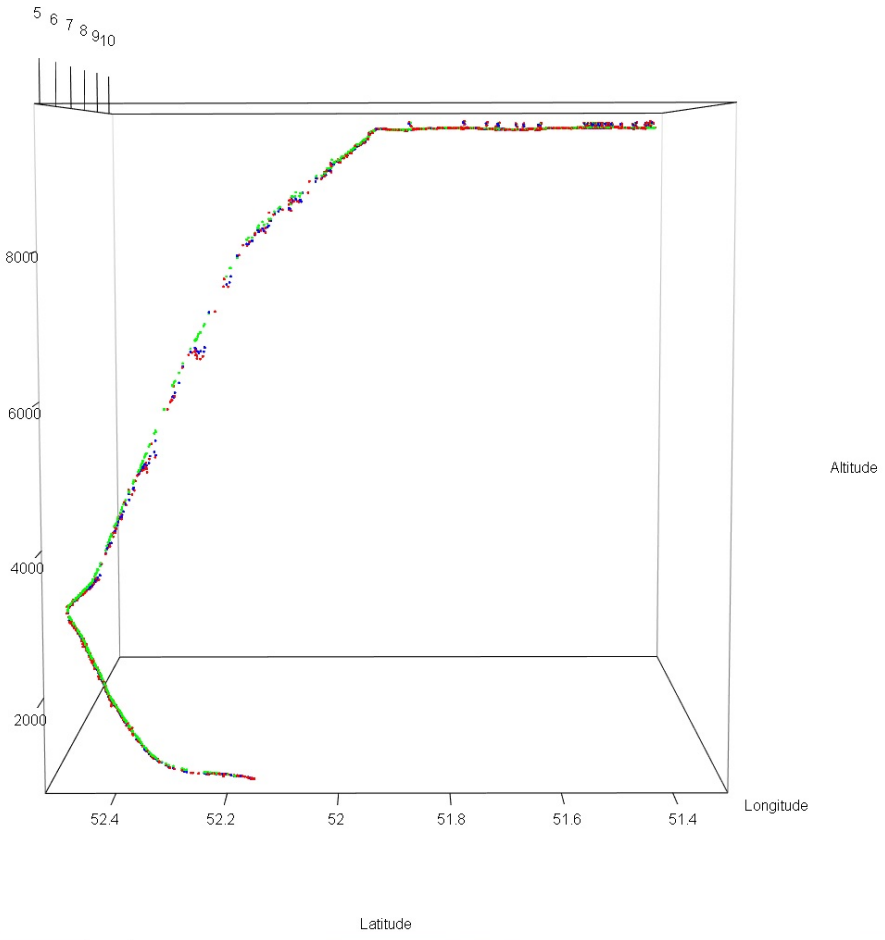


Figure A4. 3d plot of the flight trajectory under negative deviation attack, showing the aircraft in landing. Axes are longitude, latitude, and altitude. Colors are noisy (real) state in green, predicted state in red, and Kalman estimates in blue.