

# Fault Detection of Sun Reflection to Increase Estimation Accuracy of Satellite Attitude

Louw UJ<sup>1</sup>, Jordaan HW<sup>2</sup>, Schoeman JC<sup>3</sup>

**Abstract**—Methods are developed to increase the reliability of a Kalman filter when a sun reflection occurs. This is done by omitting the sun sensor reading when a reflection is detected. This method is analysed with a range of accuracies to determine the theoretical prediction accuracy to produce a robust Kalman filter exposed to sun reflections. The supervised learning methods of decision trees and random forests are implemented to predict whether the current state experiences sun reflection or not. The prediction is accompanied with feature extraction to enhance the prediction accuracy. From the analysis it is evident that the prediction accuracy is required to be 99.5%. This is a very challenging accuracy to achieve. From simulations it was found that a system that can predict and isolate sun reflections can improve the robustness and accuracy of the attitude estimate.

## I. INTRODUCTION

For many satellite missions, attitude determination is of high importance. For example, a mission that requires earth following during eclipse and otherwise sun following for solar charging, requires accurate attitude estimation. The estimation however is highly influenced by the sensor readings. The current trend in sensor anomaly detection is to use generic sensor anomalies, such as bias drift, high noise, sudden failure or any drastic change in the behaviour of the sensor. This is thus not based on practical anomalies in specific situations such as sun reflections from solar panels on a sun sensor. An anomaly like sun reflections can be accurately modelled and a binary classification can be developed that can improve the robustness and reliability of the extended Kalman filter, (EKF). In this paper the sun reflection will be modelled and the influence thereof on the attitude determination and control system, (ADCS), will be investigated.

### A. Related Work

Some of the relevant research for sensor anomaly detection was done by Wang and Liang [25], Xiong, Chan, and Zhang [26], Zhou et al. [27] and Nasrolahi and Abdollahi [12]. Wang and Liang [25] proposed an adaptive unscented Kalman filter for sensor fault estimation and isolation. Xiong, Chan, and Zhang [26] provides a fault detection method by using the residuals generated by an unscented Kalman filter to detect anomalies with a threshold based on a confidence level. Zhou et al. [27] implements a fault tolerant federated Kalman filter with three sub-filters for multi-sensor fault estimation. Nasrolahi and Abdollahi [12] provides a fault

detection and recovery method by implementing a non-linear observer to detect anomalies in attitude and rate sensors. For these methods however, the implemented failures are based on severe modelled failures such as a sudden increase in the noise of the sensor, or a sudden constant bias or any large sudden change and are generic modelled anomalies and not specific to the system.

Multiple other methods have been developed for fault detection and recovery of satellites with the focus on reaction wheels, gyroscopes and other probable failures [23, 14, 17, 8, 24, 5, 6]. Methods for sensor failure detection in other dynamical systems have also been developed which include kalman filter methodology [3], isolation forests [10] and using Long Short Term Memory, (LSTM), on sensor data to detect anomalies on engines, vehicles and aircrafts [11].

The research done on anomaly detection is extensive and only a few examples are given. In this paper, research done by Silva et al. [21] on a novel method for feature extraction is adapted and applied to satellite attitude control. An innovative moving average, determined by the error estimated with dynamic mode decomposition, (DMD), and a Kalman filter, is provided as additional input to a predictive model — decision tree. However, the method is adjusted for our use case to be a linear regression model instead of DMD.

The ADCS of the satellite is demonstrated in Figure 1, where the basic ADCS excludes fault detection, isolation and recovery, *FDIR*, and feature extraction. The *FDIR* for sensors receive the sensor measurements and the feature extractions as inputs and outputs the recovery method when required.

The EKF incorporates a physics-based model of the satellite dynamics with sensor fusion and measurement updates to ensure accurate estimation. The sensors that have a modelled vector in the orbit-referenced coordinate, *ORC*, frame, and a measured vector in the satellite body coordinate, *SBC*, frame are used for the measurement update of the EKF. According to the general satellite design, the magnetometer, sun sensor and nadir sensor are used for measurement updates. In addition, the noise of the measurements and the system's noise is incorporated in the EKF model to ensure stability and reliable estimation. The general principle for measurement updates is to update the EKF from the least to the most reliable sensor measurements. Daalen, Jones, and Jaquet [4] explains the derivation of the EKF in detail, with the specific configuration for satellites discussed in Janse van Vuuren [7].

Sun reflection also requires autonomous decision making to ensure sun-facing control. Unfortunately, the ground station cannot do this during orbit unless the control system

<sup>1</sup>Louw UJ is with Faculty of Electrical and Electronic Engineering Department, Stellenbosch University, Stellenbosch Central, Stellenbosch, 7600 louwuj@gmail.com

is dramatically changed after the ground station detects the anomaly. Therefore we aim to design a fault detection, isolation and recovery system for the specific use case of a mission that requires earth following during eclipse and sun following otherwise on a generic small satellite design as seen in Figure 2.

Sun reflection is modelled in Section II and from this modelled vector we implement methods to increase the reliability of the Kalman filter. Different recovery methods are implemented, *EKF-ignore*, *EKF-reset* and *EKF-combination*, and compared in Section V. The prediction accuracies required to increase the reliability of the Kalman filter is also analysed and a *EKF-top2* method is implemented based on a first order filter after sun reflection is predicted. This decreases the required prediction accuracy from 99.5% to 70.0%.

## II. SUN REFLECTION

The reflection anomaly is modelled for the specific shape and design of the CubeSat as shown in Figure 2.

The assumption is that the solar panel can be modelled as a geometric plane. Therefore, light from the solar panel will reflect as from a perfectly smooth mirror. It is further assumed that if the sun sensor detects any reflection from the solar panel, the measured sun vector will default to the reflection ray instead of the direct sun vector. Therefore the intensity of the light vector is disregarded. The reflected sun vector,  $r$ , can be calculated as

$$\mathbf{r} = \mathbf{v} - 2\mathbf{n}^T(\mathbf{v} \cdot \mathbf{n}). \quad (1)$$

where  $\mathbf{v}$  is the incoming sun vector and  $\mathbf{n}$  is the average vector to the plane  $ABCD$  of the solar panel, as seen in Figure 2. To calculate the intersection of the reflected vector with the plane  $XWYZ$  of the sun sensor, the equation of the plane,  $XWYZ$ , the reflected vector,  $r$ , and the point of origin is required. The reflection of the sun vector,  $\mathbf{v}$  is illustrated in Figure 3. The equation for a plane can be denoted as

$$\mathbf{p} = a\mathbf{x} + b\mathbf{y} + c\mathbf{z} + d. \quad (2)$$

where  $x$ ,  $y$  and  $z$  are the dimensions in the SBC frame. The reflected unit vector can also be translated to

$$\begin{aligned} \mathbf{x} &= \alpha t, \\ \mathbf{y} &= \beta t, \\ \mathbf{z} &= \zeta t \end{aligned} \quad (3)$$

Where the coefficients,  $\alpha$ ,  $\beta$  and  $\zeta$  are the values of the reflected unit vector in each respective dimension. Since we can calculate the coefficients for Eq 3 from the reflected vector, we can calculate  $t$ , by substituting  $x$ ,  $y$  and  $z$  into Eq 2. This is possible, because we determine the equation of the plane for the surface  $XYZW$  based on our design.

After that, the intersecting point with the plane  $XYZW$  can be calculated as

$$P(x, y, z) = (o_1 + \alpha t, o_2 + \beta t, o_3 + \zeta t) \quad (4)$$

where  $o_1, o_2, o_3$  is the point of origin. Which in this case is the position of reflection from the solar panel. Therefore,

if the sun vector  $\mathbf{v}$  reflected from the solar panel as  $\mathbf{r}$ , the point of intersection  $Q'$  on Figure 3 can be calculated as

$$Q'(x, y, z) = (Q_x + \alpha t, Q_y + \beta t, Q_z + \zeta t) \quad (5)$$

To model reflection from the solar panels to the sun sensor, only two corners of the solar panel and two corners of the sun sensor are to be considered. From Figure 3 it is evident that if the solar panel reflects on  $Y$  that the reflection will also cover  $X$ . The same is true for corner  $Z$  and  $W$ . Since  $C'$  will be at the same position as  $C$ , which is valid for  $D'$  and  $D$ , the calculation can be omitted. Therefore it is only necessary to calculate the reflected positions  $A'$  and  $B'$ . This simplifies the reflection model significantly.

The reflected position  $A'$  can be calculated as the intersection of the reflected vector  $R$  with plane  $XYZW$  using Eq 4. We also know the position of  $A$ , based on the satellite design and can therefore calculate  $A'$ . The same applies to  $B$  and  $B'$ . To determine whether  $Y$  or  $X$  is within the reflection region, we assume that the plane  $XYZW$  is a 2D plane, and we omit the third dimension. Therefore, the axis changes from  $x, y, z$  to only  $x, y$ . We calculate whether  $x$  is between the lines of  $A'D$  and  $B'C$  and between the lines  $CD$  and  $A'B'$ . By determining the line equation between reflected points in the form

$$y_{A'B'} = mx_{A'B'} + c \quad (6)$$

from the coordinates of  $A'$  and  $B'$ , the corresponding  $X_{A'B',y}$  can be calculated by substituting  $X_x$  into Eq 6. With the same method the coordinates of  $X_{B'C}$ ,  $X_{A'D}$ ,  $X_{A'B'}$  and  $X_{CD}$  can be determined. After that, with logical if statements, it can be determined whether  $X$  is in the reflection zone. If  $X_x$  is to the right of  $X_{B'C,x}$  and to the left of  $X_{A'D,x}$ , as well as  $X_y$  is above  $X_{A'B',y}$  and below  $X_{CD,y}$  then  $X$  is within the reflection zone.

The results for the sun vector with and without reflection is shown in Figure 4. During the modelling of the reflection, the reflection also affects the estimation and, therefore, the attitude control of the satellite. In the figures of this article, the grey zones indicate the eclipse period, as seen in Figure 4.

## III. ANOMALY DETECTION

To recover from sensor anomalies or exclude the sensor from the Kalman filter, the anomaly must be detected, and the sensor from which the anomaly in the data occurs must be classified.

### A. Feature Extraction

The first step to implementing an FDIR for Kalman filter robustness is to detect whether an anomaly has occurred on one of the filters. There are various methods for fault detection, with both supervised and unsupervised methods. However, this study will only focus on a single method proposed by Silva et al. [21] to detect failures in sensors.

The proposed method by Silva et al. [21] uses Dynamic Mode Decomposition (DMD), which was initially developed by Schmid et al. [19] and further expanded to include control by Proctor, Brunton, and Kutz [16], to provide an estimation

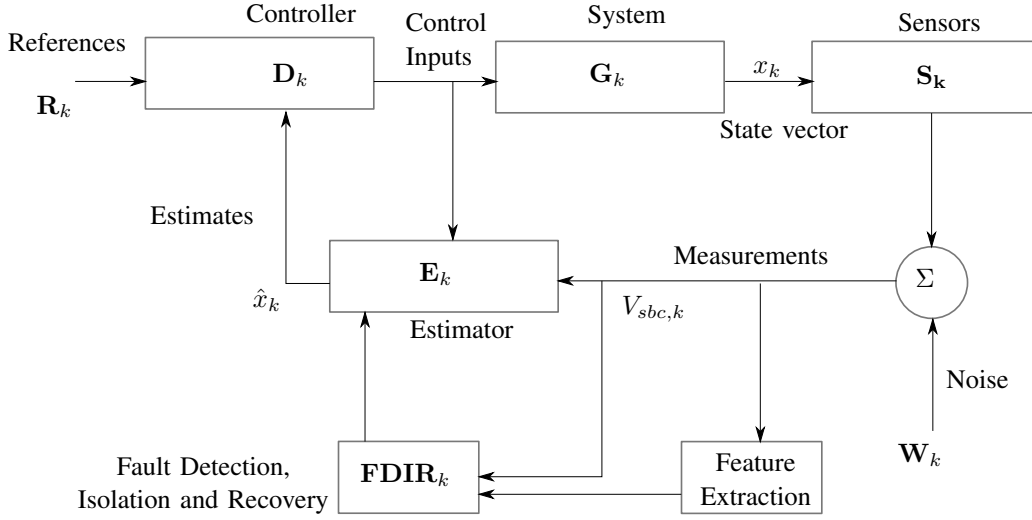


Fig. 1: System Diagram

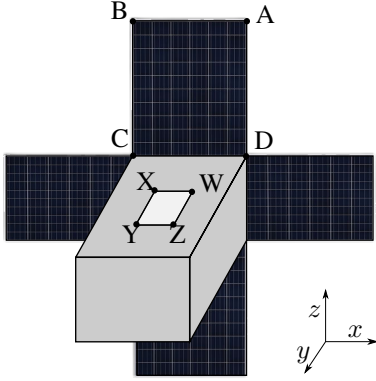


Fig. 2: Cube Sat

of a sensor vector based on the previous measurement of the sensor as well as the measurements of the other sensors in the system. DMD was first developed in the fluids community and constructed a matrix  $\mathbf{A}$  to relate the state vector  $x$  with the following time step of the state vector,  $x_{k+1}$ . The state vector, in our case, will be the measurement vector of the specific sensor that we want to monitor.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \quad (7)$$

Where  $\mathbf{x}_k$  and  $\mathbf{x}_{k+1}$  during a specified number of time steps, will be denoted as  $\mathbf{X}$  and  $\mathbf{X}'$  respectively.

The method of DMD, however, is useful for high order systems where the calculation of  $\mathbf{A}$  is computationally intensive. This is not the case for our system, and using DMD is not justifiable and consequently, a linear regression model is implemented. Therefore with the pseudo-inverse of  $\mathbf{X}$ , denoted as  $\mathbf{X}^\dagger$ , we calculate  $\mathbf{A}$  as

$$\mathbf{A} = \mathbf{X}\mathbf{X}^\dagger \quad (8)$$

This necessitates data of the state vector over time. The article by Silva et al. [21] however includes  $\mathbf{B}$  to relate

the vector measurements of the other sensors to adjust the predicted state,  $\mathbf{X}_{k+1}$  of the monitored sensor.

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{B}\mathbf{Y}_k \quad (9)$$

Where  $\mathbf{Y}_k$  is the other sensor measurements, this is adjusted for our use case, where  $\mathbf{Y}_k$  is the control torques for the magnetorquers and reaction wheels, while  $\mathbf{X}_k$  is all of the sensor measurements. Consequently, the model of Eq 9 denotes the prediction of the sensor measurements at time step  $k + 1$  based on the current sensor measurements and control inputs. Thereafter, as implemented by Silva et al. [21] the model is adjusted with a Kalman Filter. From  $\mathbf{A}$  and  $\mathbf{B}$  the Kalman filter can be implemented to predict  $\mathbf{X}_{k+1}$

$$\hat{\mathbf{X}}_{k+1} = \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{Y}_k + K(\mathbf{X}_k - \hat{\mathbf{X}}_k) \quad (10)$$

where  $K = 0.001$ . After the calculation of  $\hat{\mathbf{X}}_{k+1}$  Silva et al. [21] proposes a moving average of the innovation covariance

$$\mathbf{V}_k = \frac{1}{N} \sum_{i=k-N}^k (\mathbf{X}_i - \hat{\mathbf{X}}_i)(\mathbf{X}_i - \hat{\mathbf{X}}_i)^T \quad (11)$$

where  $N$  is the number of timesteps to account for. The moving average is used as an additional input parameter for the classification of anomalies based on  $\mathbf{X}$ .

### B. Classification

Decision trees and random forests will be implemented to perform binary classification on regular data samples and anomalous data samples. Decision trees [18] and random forests [20, 13, 15] are supervised learning algorithms that classify data based on threshold splitting. Where data samples are split based on a threshold of a specific input parameter. For instance, binary classification can be performed on data samples from a satellite orbit to determine whether the satellite was in an eclipse or not. This would be done by determining whether the magnitude of the sun vector is smaller or greater than 0. Consequently, after two splits it can be determined which data samples are within an eclipse.

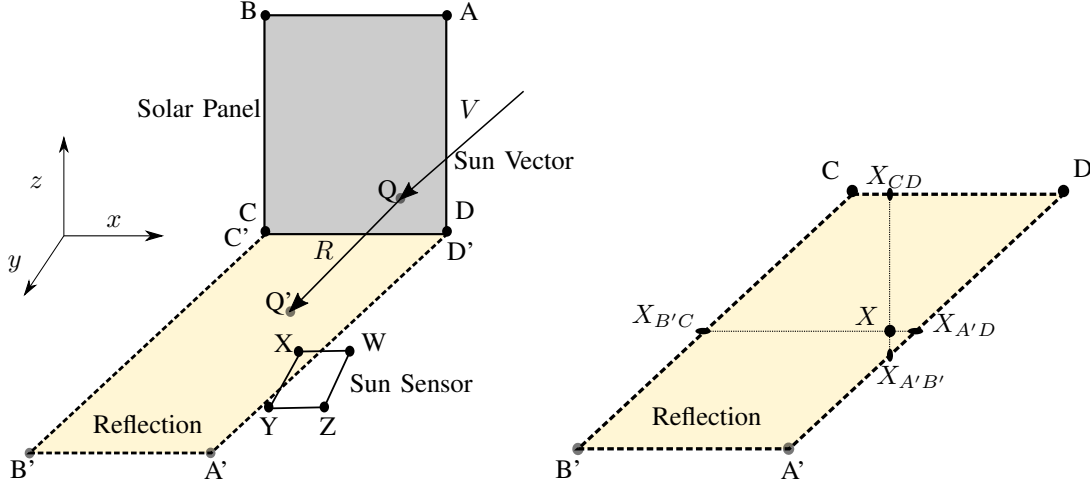
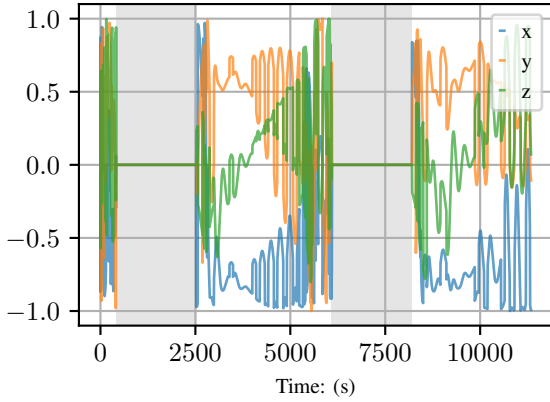
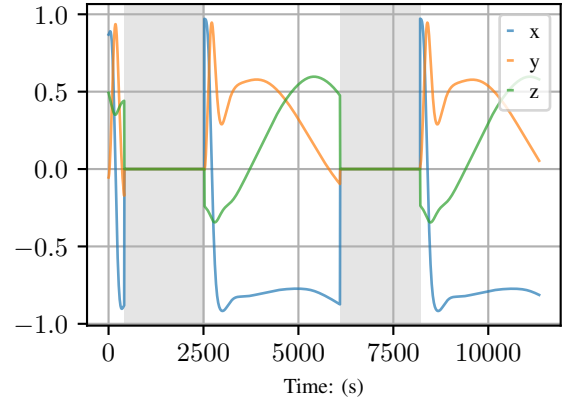


Fig. 3: Reflection



(a) Sun vector with reflection.



(b) Sun vector without reflection.

Fig. 4: Comparison of Sun Vector with and without Reflection

The decision tree determines this split with the classification and regression tree, *CART*, algorithm.

However, to split the data for the anomalies, we need to decide which input parameter will be used to make the first split, the root node. The Gini index measures the probability of a data sample being wrongly classified at a given node. This can be calculated with Eq 12.

$$GI = 1 - \sum_{i=1}^n (P_i)^2 \quad (12)$$

The operator split that produces the lowest Gini index provides the purest split and will be used as the root node. For our use case, the *CART* algorithm will be used to optimise the decision tree, which also considers the most prominent information gain to construct the decision tree. Figure 5 is a graphical representation of the decision tree developed to classify anomalies. The depth of a decision tree determines how many splits occur from the root node to the leaf node the furthest from the first split. If the depth is unspecified, the decision tree will split until all the data

samples are perfectly split into anomalous and normal data samples. However, the larger the depth, the more biased the decision tree is to the training data. This depth can be altered to optimise the efficiency and accuracy of the decision tree.

It is evident in Figure 5 that the most notable splits at the beginning of the tree are the moving average from the feature extraction as well as the sun sensor and angular momentum of the reaction wheels measurements. This makes logical sense since the moving average provides the system with an extracted feature that correlates with the change in measurements. The sun reflection can also be detected on the sun sensor measurements, and changes in the angular momentum of the wheels indicate that there could be significant changes in the sun vector during the sun following period.

Random forests, a method of using the prediction average of randomly sampled decision trees, is also tested, and the results thereof are shown in Section V.

### C. Recovery

Three different methods of recovery are compared. These methods are all focused on ensuring that the sun reflection

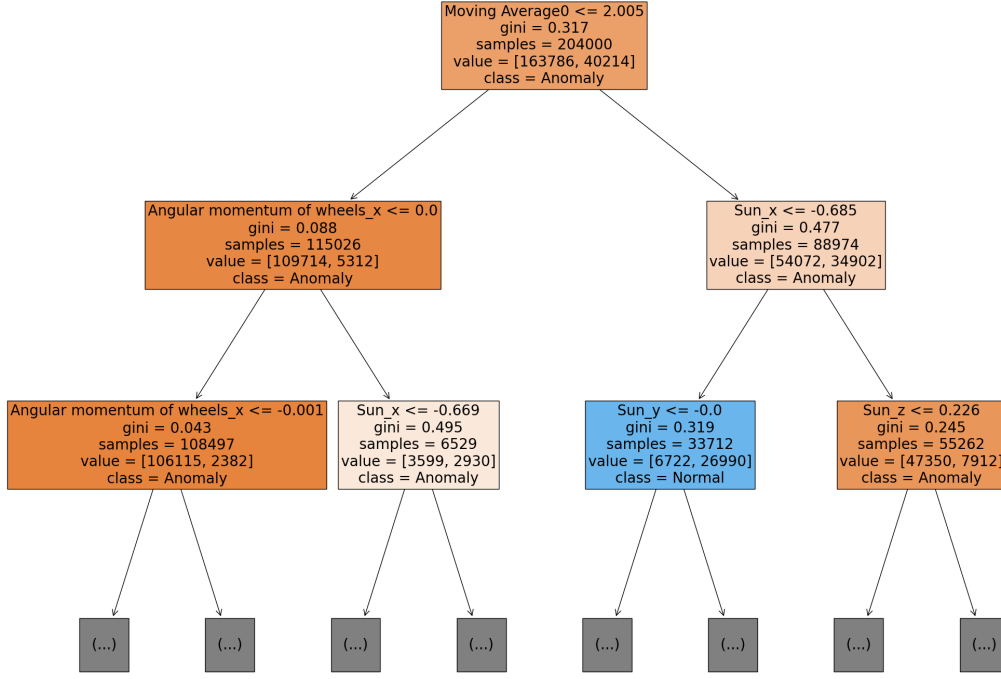


Fig. 5: Decision Tree

does not change the reliability and stability of the EKF.

The *EKF-ignore* method uses the detected sensor that has failed and ignores the sensor measurement from the EKF measurement update. This method is based on the assumption that the EKF estimation is correct up until the moment where the sensor failure is detected. This, however, will highly depend on the accuracy of the anomaly detection method. Since a detection method with low accuracy will create instability of the EKF since many anomalous measurements will be included in the measurement update of the EKF.

The *EKF-reset* method uses a buffer of  $v_{meas,k}$ ,  $v_{model,k}$  and  $\hat{x}_k^+$  and other parameters that are used to update the EKF. If a sensor failure is detected, the sensor is excluded from the EKF, and the EKF is updated with the sensor data in the buffer, excluding the sensor that has failed. The EKF is, therefore, *reset* and updated from timestep  $t_{k-N}$  to  $t_k$ , where  $N$  is the size of the number of timesteps in the buffer.  $N$ , however, must be optimised based on the computational time used to reset the EKF but still ensure convergence of the EKF. If the sensor that was detected to have anomalous behaviour changes back to normal again, the EKF will be reset once again, and the sensor will only be included in the measurement update of  $t_k$  since it was anomalous for timesteps before  $t_k$ .

A backtracking method can be combined with the ignore method, *EKF-combination*. For example, where the backtracking method is implemented only after a specified number of sun reflections are predicted.

Another method implemented and tested continually uses the two sensors measurements that have the smallest mean

squared error between the estimated SBC vector and the actual measured SBC vector. There are, however, setbacks to this method. Firstly, it requires the modelling of the ORC vector and requires the position of the satellite in orbit. Secondly, this method will not work with small drifts in a sensor measurement since the estimator will latch onto the drift in the sensor. The method will only detect sudden changes in the sensor and isolate the sudden change even if it remains stable after the sudden change. This method will only be compared to the other methods as part of the analysis since the method is inherently different. The method is also used to aid the other methods during a period after an anomaly is detected.

#### IV. TESTING SETUP

The *sgp4* simulation environment is used for the positioning in orbit of the satellite to ensure repeatability of the tests conducted in this article. The disturbance torques modelled in this simulation is the aerodynamic disturbances, static and dynamic wheel disturbances, gravity gradient disturbances, and gyroscopic disturbances. The testing for the FDIR methods is done by implementing a reflection model on a CubeSat from the moment of launching the satellite. Therefore the recovery methods are also implemented from the beginning of the satellite orbit. The mission of the ADCS of this specific satellite is to be nadir pointing during eclipse and sun following otherwise.

##### A. Control

The attitude command vector during nadir-pointing in the SBC frame is  $\mathbf{u}_c = [0, 0, 1]$ , since the SBC frame  $z$  coordinate should line up with the ORC frame. During the

sun following phase, the attitude command according to Chen, Steyn, and Hashida [2] can be calculated as

$$\mathbf{u}_c = \frac{\mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o}{\|\mathbf{u}_{sp}^{SBC} \times \mathbf{s}_o\|} \quad (13)$$

Where  $\mathbf{s}_o$  is the measured unit sun vector in ORC, and the main solar panel's position is denoted as a unit vector,  $\mathbf{u}_{sp}^{SBC}$ . The angle between  $\mathbf{u}_{sp}^{SBC}$  and  $\mathbf{s}_o$ ,  $\delta$ , can be calculated with the vector dot-product. The command quaternion  $\mathbf{q}_c$  can then be calculated

$$\mathbf{q}_c = \begin{bmatrix} \mathbf{u}_c \sin(\frac{\delta}{2}) \\ \cos(\frac{\delta}{2}) \end{bmatrix} \quad (14)$$

This can then be used as the reference for the control. The reference  $\omega_b^I$  is always  $[0, 0, 0]$ .

### B. Dimensions of Satellite

The dimensions of the satellite are shown in Table I. The dimensions are provided to ensure the repeatability of the results in this article. The dimensions of the sun sensor are from the Sputnix CubeSat sun sensor model. The sun sensor is also placed next to the solar panel to increase the number of reflections. This is done to produce the worst possible scenario for our FDIR system.

TABLE I: Dimensions of CubeSat

Dimensions	Satellite (m)	Solar Panels (m)	Sun Sensor (m)
<b>x</b>	0.3	0.3	0.028
<b>y</b>	0.3	0.3	0.023
<b>z</b>	0.4	0.002	N/A

### C. Orbit Parameters

The orbit parameters will not significantly affect the results. However, for repeatability, the general parameters for the orbit are given in Table II.

TABLE II: Parameters for CubeSat Orbit

<b>Revolutions per day</b>	15.2355
<b>Inclination</b>	97.4°
<b>Right ascension of the ascending node</b>	275°

### D. Sensors

The measurement update of the Kalman filter will firstly be with the magnetometer, thereafter the nadir sensor and lastly, the sun sensor. This is due to the noise models of the sensors, as all the sensor noise models are based on zero-mean Gaussian random noise. The magnetometer has the most significant standard deviation, and the sun sensor has the smallest standard deviation. There are two sun sensors, a coarse and fine sun sensor, that can both experience sun reflection. The field of view *FOV* of the sun sensors and the nadir sensor are both 180°. There is, however, only a single nadir sensor used in the simulation.

## V. RESULTS

Three scenarios are implemented, a satellite that never experiences reflection, a satellite that experiences reflection without any recovery method and a satellite with a recovery method. The subsets of detecting and recovering from the fault will be isolated and discussed separately. Therefore the results for recovery based on perfect detection can be shown to provide the theoretical possibilities of the recovery method. Please note that the scale of the y-axis for each plot is not the same due to the significant differences between the maximum y-values for each scenario.

The pointing metric is the difference between the command or reference attitude and the actual attitude in degrees. The estimation metric is the difference between the actual and estimated attitudes in degrees.

### A. Perfect Designed Satellite Without Reflection

This test is implemented for the current design, assuming that the sun sensor will never experience sun reflection. This also indicates the best scenario for the ADCS and requires a well-designed satellite. The results for the first 5 orbits are shown to provide a desired result for the recovery methods as shown in Figure 6 and Figure 7.

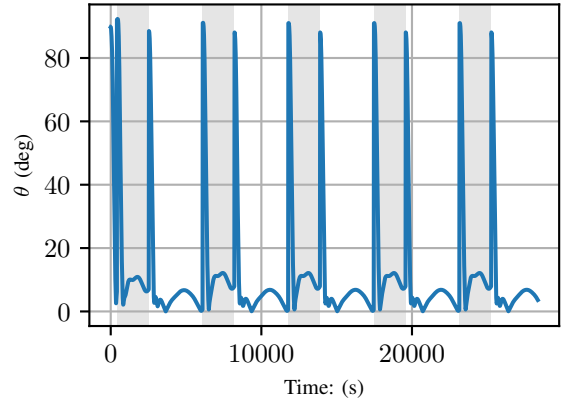


Fig. 6: Pointing Metric for Satellite Without Reflection.

Figure 6 shows the transition of the reference vector between the eclipse and sunlit phases. This is the reason for the spikes in the pointing metric at the beginning of each period. From Figure 7 it is also evident that the estimation is more accurate during the sunlit phase, which correlates with the fact that the sun sensor has a smaller noise.

### B. Satellite With Reflection

If no recovery strategy is implemented for the reflection anomaly, the EKF is unstable, and the estimation is inaccurate. As a bare minimum, the proposed methods should lower the estimation metric as shown in Figure 8 of which the average estimation metric per orbit is also indicated in Figure 9.

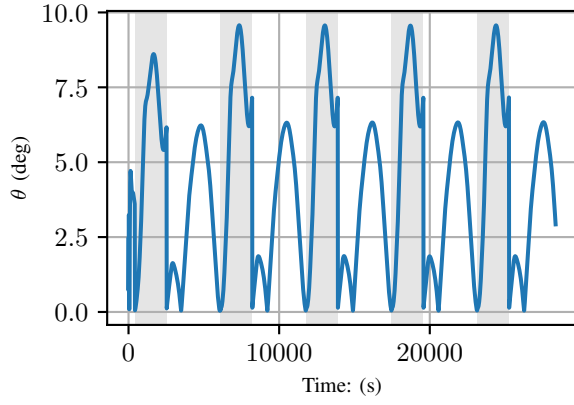


Fig. 7: Estimation Metric for Satellite Without Reflection.

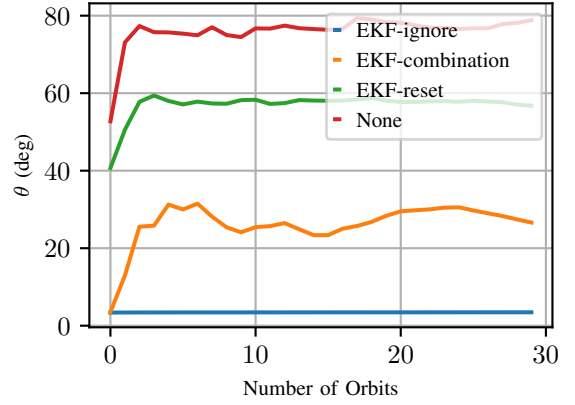


Fig. 9: Estimation Metric for Perfect Detection.

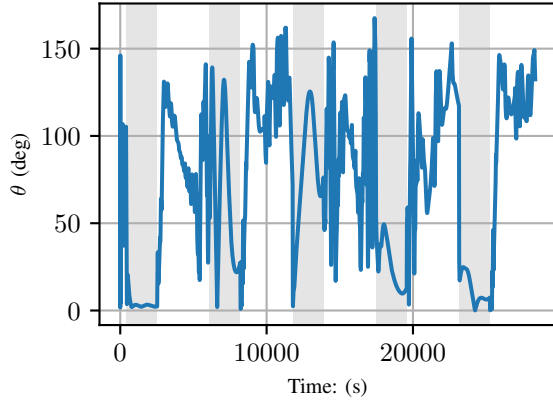


Fig. 8: Estimation Metric for Satellite With Reflection and Without FDIR.

during the first five orbits. Our goal is to introduce detection strategies that produce similar results.

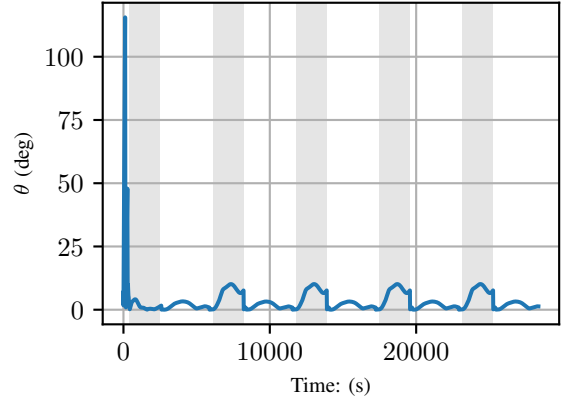


Fig. 10: Estimation Metric for Satellite with Recovery Ignore.

### C. Perfect Detection

This section provides the results for each recovery method, with the assumption that the detection method is perfect. This means that the detection methodology will produce an accuracy of 100%. This is used to determine which method is most suitable for recovery from sensor anomalies since it provides the theoretical best scenario for each method. The mean estimation for each orbit is shown in Figure 9, and it is evident that the EKF-ignore method is the method that reduces the estimation error the most during a period of 30 orbits.

Due to the significant difference in performance between the EKF-ignore and the other recovery methods, only the EKF-ignore method will be further investigated with other detection strategies. The EKF-ignore recovery methods provides an average estimation metric of  $3.43^\circ$  per orbit.

### D. Satellite With Recovery Ignore

To show the promising results of the recovery ignore for perfect detection, Figure 10 shows the estimation metric

To analyse what is required from the detection strategies to produce acceptable results, we perform a few tests with fixed percentage accuracies. This analysis is done by performing uniform randomness with the fixed accuracy percentage. Firstly we test the fixed accuracies by performing prediction on reflection and no reflection time steps. Therefore there can be both false positives and false negatives. The results, therefore, are shown in Figure 11.

It is evident in Figure 11 that the percentage accuracy to produce an estimation metric average of less than  $20^\circ$  per orbit is 99%. This is a challenging task to accomplish. Therefore, we want to determine the effects of perfect prediction during no reflection time steps. We perform a test where the prediction is only performed when reflection occurs, and during no reflection, a 100% accuracy is used. The results, therefore, is shown in Figure 12, which indicates that the most significant effect on the EKF's estimation accuracy is not false negatives, but false positives.



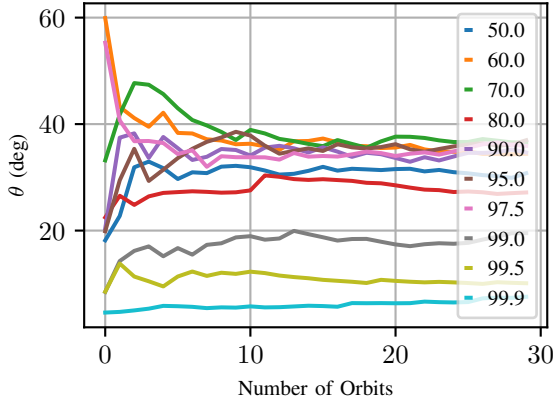


Fig. 11: Estimation Metric of Fixed Percentage Accuracies During Failures and No Failures.

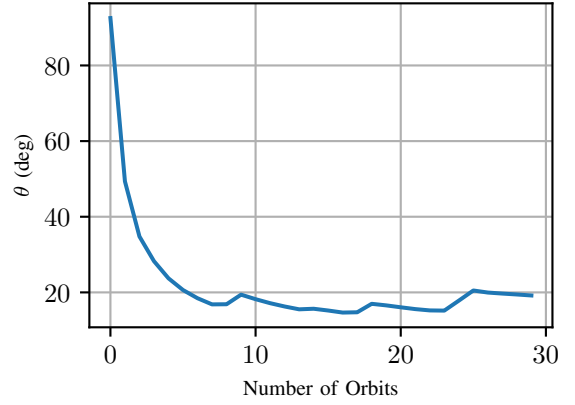


Fig. 13: Estimation Metric for Satellite with Recovery Ignore and Various Detection Methods.

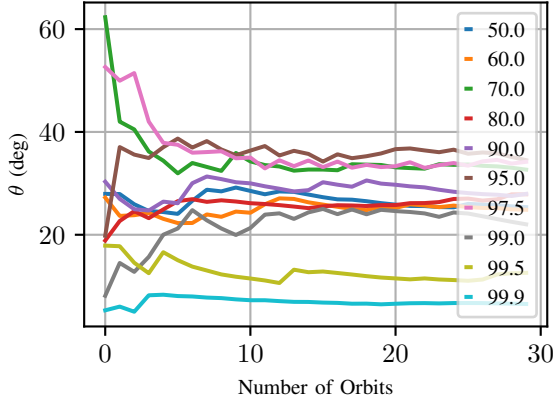


Fig. 12: Estimation Metric of Fixed Percentage Accuracies During Reflection and Perfect Prediction Otherwise.

Therefore, we want to make sure that more failures are accurately predicted. Therefore, we want to implement a first-order filter after a failure is predicted to reduce the number of incorrectly predicted reflections. This, however, cannot be done by just ignoring the sun vector for some time since this will reduce the accuracy.

#### E. First Order Filter

A method that only uses the two sensors whose measurement is the closest to the estimated vectors from the EKF for the measurement update is implemented. This method will be named EKF-top2 for ease of reference. The results of this method are shown in Figure 13. The average estimation error is reduced to less than  $20^\circ$ . This method will consequently be used during a specific period after a failure is predicted. However, it cannot be used continually since the overall accuracy is reduced by only using 2 sensors and assuming that there is always a failure.

Now we want to test what the estimation error is for various fixed accuracy percentages with the EKF-ignore

method and then implement the EKF-top2 method during the buffer period. We firstly conduct the test for various time steps during which the EKF-top2 must be performed at a constant accuracy of 90% for which the results are shown in Figure 14.

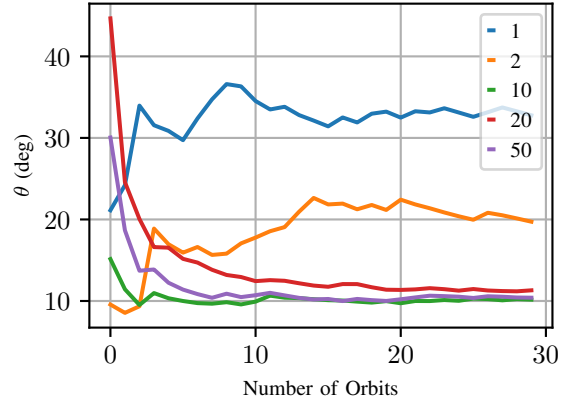


Fig. 14: Estimation Metric of 90.0% Accuracy with Various Time Delays.

From Figure 14 the buffer period is selected as 10 time steps after a failure is detected. The results thereof are shown in Figure 15 for a range of realistic accuracies as expected from the various detection strategies.

The required percentage accuracy is lowered dramatically with the first order filter implementation. A percentage accuracy of 70.0% with the first order filter is performing similarly to a 99.5% accuracy without the buffer after 10 orbits. After we have now analysed the results for various fixed accuracies and the first order filter, we can now implement the detection methods with the recovery methods and determine the efficiency of the detection methods.



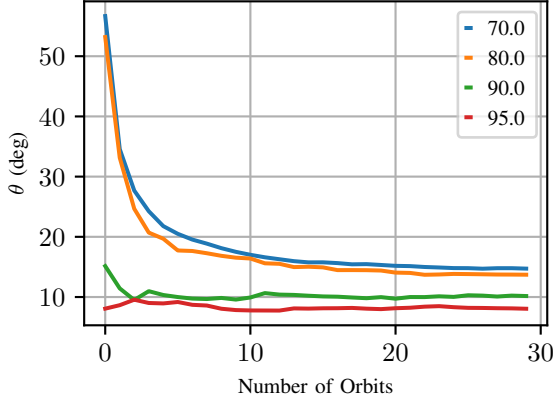


Fig. 15: Estimation Metric of Fixed Percentage Accuracies During Failures and No Failures with Recovery Buffer.

#### F. Detection Methods

Firstly, we want to analyse how the maximum root depth for both random forest and decision trees changes the accuracy of the detection strategies. In Table III the false and true positives and false and true negatives are shown in the confusion matrices for different depths. It is clear from the average of the false positive and false negative that a depth of 20 is the best for the random forest. This is also chosen instead of a 100 since a depth of a 100 is computationally much more expensive than a depth of 20.

For the decision tree the confusion matrices are shown in Table IV. Although the lowest false negatives are at a depth of 5, the false positive is significantly higher than a depth of 10 and 15. This essentially means that there are multiple instances where a failure will be predicted while there is no failure. Comparing the depth of 10 and 20, the sum of the false positive and false negatives are 211 more at a depth of 20. However, false negatives are more concerning than false positives, and from a depth of 10 to a depth of 20, the false negatives decrease by 4494. The depth of 100 has too many false positives for the small decrease of false negatives, and consequently, a depth of 20 is used.

Using a depth of 20 for both random forest and decision trees, we determine the average estimation error for 30 orbits with and without the first order buffer of 10 time steps as shown in Figure 16 and Figure 17. It is clear that the first order buffer implementation increases the estimation error for the decision tree method but reduces the estimation error for the random forest. Comparing the prediction accuracy for both the decision tree and random forest, it is clear that both methods should at least produce better results than the theoretical 80 accuracy. However, it is clear that they have higher estimation errors than the theoretical 80.0%.

To determine the reason for the behaviour of the detection strategies, we plot the prediction accuracy during the first 2 orbits in Figure 19. This is used to determine the time gaps between incorrect predictions, which would vary from

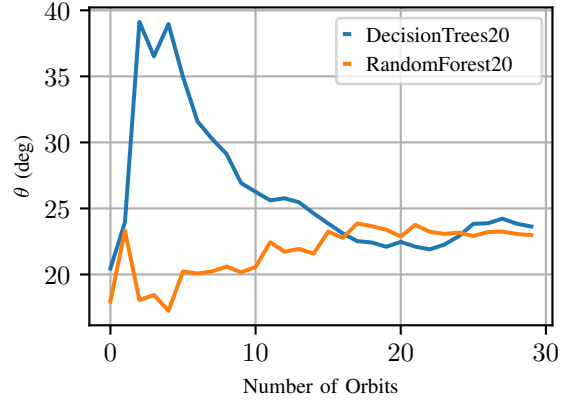


Fig. 16: Estimation Metric of Decision Tree and Random Forest.

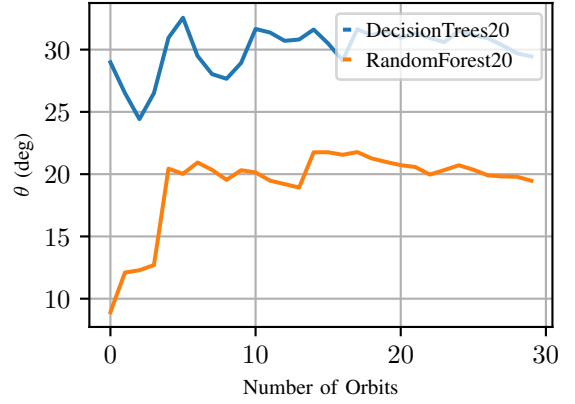


Fig. 17: Estimation Metric of Decision Tree and Random Forest with Recovery Buffer.

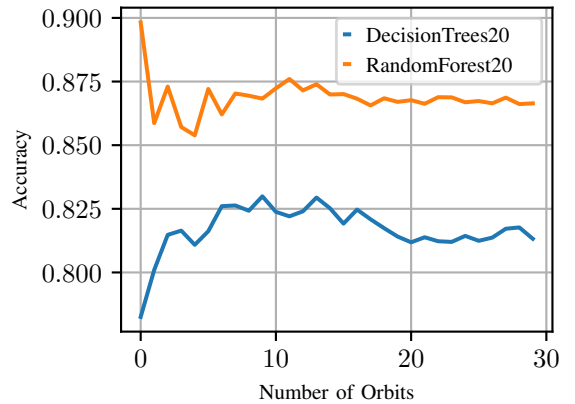


Fig. 18: Prediction Accuracy of Decision Tree and Random Forest with Recovery Buffer.

the fixed accuracies. We isolate the random forest detection

TABLE III: Confusion Matric for RandomForest

Predicted		Predicted		Predicted		Predicted	
Failure	No Failure	Failure	No Failure	Failure	No Failure	Failure	No Failure
90103	11825	88250	7384	91060	6238	89962	6663
19432	48768	16388	58106	16931	55899	16427	57076
(a) Depth of 5		(b) Depth of 10		(c) Depth of 20		(d) Depth of 100	

TABLE IV: Confusion Matric for Decision Trees

Predicted		Predicted		Predicted		Predicted	
Failure	No Failure	Failure	No Failure	Failure	No Failure	Failure	No Failure
95155	3050	84981	15728	86013	11234	86833	9195
31061	40862	12304	57115	17009	55872	25183	48917
(a) Depth of 5		(b) Depth of 10		(c) Depth of 20		(d) Depth of 100	

strategy for this analysis since it significantly outperforms the decision tree method with the first order filter implementation.

From Figure 17 we see that the estimation metric for the first orbit is an average of  $8.9^\circ$ , which increases to  $12.1^\circ$  in the second orbit, and by the  $5^{th}$  orbit, the estimation error has increased to  $20.8^\circ$ . From Figure 19 and Figure 18, it is clear that there are two reasons for this increase in estimation error. Firstly, the average prediction accuracy decreases from 90% during the first orbit to settle at an average of 86%. Secondly, the duration of inaccurate predictions increases, as seen after 7500s. These extensive durations of constant inaccurate predictions will increase the instability and decrease the accuracy of the EKF. These inaccurate predictions are due to the change of the satellite behaviour during recovery, which, in turn, influences the prediction. It is thus recommended to implement non-linear machine learning methods such as support vector machines which will not produce single thresholds for each parameter but a kernel function that splits the data with a n-dimensional plane.

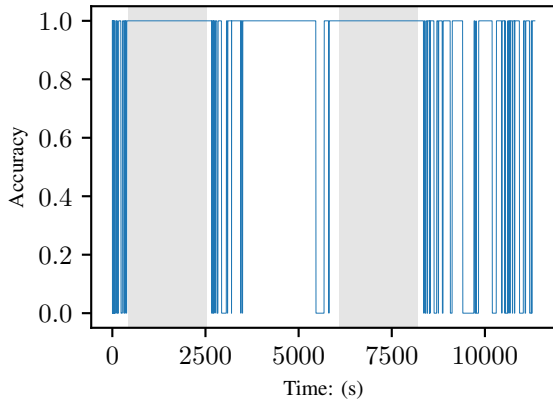


Fig. 19: Prediction Accuracy of Decision Tree and Random Forest with Recovery Buffer.

## VI. CONCLUSIONS

It is clear from the analysis of the results that sun reflection recovery is possible with the proposed EKF-ignore method. It is also clear that the percentage accuracy for the detection is drastically decreased with a first-order buffer and the EKF-top2 method. However, both the decision tree and random forest method do not produce the desired results due to the decreased inaccuracy of the prediction after the recovery method is implemented and large durations of inaccurate predictions. Therefore, it is recommended to further the testing with methods such as support vector machines.

## REFERENCES

- [1] Jacoba Auret. “Design of an aerodynamic attitude control system for a CubeSat”. PhD thesis. Stellenbosch: Stellenbosch University, 2012.
- [2] Xiaojiang Chen, Willem Steyn, and Yoshi Hashida. “Ground-target tracking control of earth-pointing satellites”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2000, p. 4547.
- [3] O Ciftciogl. “SENSOR FAILURE DETECTION IN DYNAMICAL SYSTEMS BY KALMAN FILTERING METHODOLOGY Service unit General Services”. In: March (1991).
- [4] C van Daalen, T Jones, and D Jaquet. “Advanced Estimation 813 :” in: (2017).
- [5] Sara K Ibrahim et al. “Machine learning methods for spacecraft telemetry mining”. In: *IEEE Transactions on Aerospace and Electronic Systems* 55.4 (2018), pp. 1816–1827.
- [6] Sara K Ibrahim et al. “Machine learning techniques for satellite fault diagnosis”. In: *Ain Shams Engineering Journal* 11.1 (2020), pp. 45–56.
- [7] Gerhard Hermann Janse van Vuuren. “the Design and Simulation Analysis of an Attitude Determination and Control System for a Small”. In: March (2015).

- [8] Jaehyun Jin, Sangho Ko, and Chang-Kyung Ryoo. "Fault tolerant control for satellites with four reaction wheels". In: *Control Engineering Practice* 16.10 (2008), pp. 1250–1258.
- [9] Hendrik Willem Jordaan. "Spinning Solar Sail: The Deployment and Control of a Spinning Solar Sail Satellite". In: March (2016).
- [10] Desheng Liu et al. "Sensors Anomaly Detection of Industrial Internet of Things Based on Isolated Forest Algorithm and Data Compression". In: *Scientific Programming* 2021 (2021). ISSN: 10589244. DOI: 10.1155/2021/6699313.
- [11] Pankaj Malhotra et al. "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection". In: (2016). arXiv: 1607.00148. URL: <http://arxiv.org/abs/1607.00148>.
- [12] Seiiid Saeed Nasrolahi and Farzaneh Abdollahi. "Sensor fault detection and recovery in satellite attitude control". In: *Acta Astronautica* 145. February (2018), pp. 275–283. ISSN: 00945765. DOI: 10.1016/j.actaastro.2018.01.002. URL: <https://doi.org/10.1016/j.actaastro.2018.01.002>.
- [13] Angshuman Paul et al. "Improved Random Forest for Classification". In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 4012–4024. ISSN: 10577149. DOI: 10.1109/TIP.2018.2834830.
- [14] Barbara Pilastre et al. "Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning". In: *Signal Processing* 168 (2020), p. 107320. ISSN: 01651684. DOI: 10.1016/j.sigpro.2019.107320. URL: <https://doi.org/10.1016/j.sigpro.2019.107320>.
- [15] Rifkie Primartha and Bayu Adhi Tama. "Anomaly detection using random forest: A performance revisited". In: *Proceedings of 2017 International Conference on Data and Software Engineering, ICoDSE 2017* 2018-January (2018), pp. 1–6. DOI: 10.1109/ICODSE.2017.8285847.
- [16] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. "Dynamic mode decomposition with control". In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161.
- [17] Afshin Rahimi and Atilla Saadat. "Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning". In: *Aerospace Systems* (2020), pp. 1–8.
- [18] Matthias Reif et al. "Anomaly detection by combining decision trees and parametric densities". In: *Proceedings - International Conference on Pattern Recognition* January (2008). ISSN: 10514651. DOI: 10.1109/icpr.2008.4761796.
- [19] Peter J Schmid et al. "Applications of the dynamic mode decomposition". In: *Theoretical and Computational Fluid Dynamics* 25.1 (2011), pp. 249–259.

- [20] Tao Shi and Steve Horvath. “Unsupervised learning with random forest predictors”. In: *Journal of Computational and Graphical Statistics* 15.1 (2006), pp. 118–138. ISSN: 10618600. DOI: 10.1198/106186006X94072.
- [21] Brian M. de Silva et al. “Physics-informed machine learning for sensor fault detection with flight test data”. In: (2020), pp. 1–21. arXiv: 2006.13380. URL: <http://arxiv.org/abs/2006.13380>.
- [22] Herman Steyn. July 2021.
- [23] N. Tudoroiu and K. Khorasani. “Satellite fault diagnosis using a bank of interacting Kalman filters”. In: *IEEE Transactions on Aerospace and Electronic Systems* 43.4 (2007), pp. 1334–1350. ISSN: 00189251. DOI: 10.1109/TAES.2007.4441743.
- [24] Alexandra Wander and Roger Förstner. *Innovative fault detection, isolation and recovery strategies on-board spacecraft: state of the art and research challenges*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2013.
- [25] Mao Wang and Tiantian Liang. “Adaptive Kalman filtering for sensor fault estimation and isolation of satellite attitude control based on descriptor systems”. In: *Transactions of the Institute of Measurement and Control* 41.6 (2019), pp. 1686–1698.
- [26] K. Xiong, C. W. Chan, and H. Y. Zhang. “Detection of satellite attitude sensor faults using the UKF”. In: *IEEE Transactions on Aerospace and Electronic Systems* 43.2 (2007), pp. 480–491. ISSN: 00189251. DOI: 10.1109/TAES.2007.4285348.
- [27] Jun Zhou et al. “A scheme of satellite multi-sensor fault-tolerant attitude estimation”. In: *Transactions of the Institute of Measurement and Control* 38.9 (2016), pp. 1053–1063. ISSN: 01423312. DOI: 10.1177/0142331216642838.