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

Louw UJ[1], Jordaan HW[2], Schoeman JC[3]

Abstract— The Kalman Filter is a state estimator that is often used in attitude determination of satellites. A Kalman filter is highly sensitive to anomalies that occur in sensors. A good example of this is the reflection of a solar panel on a sun sensor that changes the perceived sun vector. This in term influences the estimation of the attitude by the kalman filter and consequently the control of the satellite. Detecting anomalies in sensors and omitting the sensor reading from the measurement update of the Kalman Filter increases the stability and reliability of the Kalman filter for satellite attitude determination. However detecting when an anomaly occurs is difficult and an accuracy of more than 99% is required to produce satisfactory results.

keywords — anomaly detection, Kalman filter, machine learning, sun reflection, satellites

I. INTRODUCTION

For many satellite missions the attitude determination is of high importance. A mission that requires earth following during eclipse and otherwise sun following for solar charging requires accurate attitude estimation. The current state vector of the system can not be determined with only the use of models or sensors. Since the sensors contains noise and the mathematical model does not include certain disturbances of the actual system. Therefore a probabilistic approach can be used to determine the attitude from both the model and the sensors. This probabilistic approach is done with the use of an extended kalman filter (*EKF*).

The focus of this article is the attitude determination and control system, *ADCS*, of the satellite. This system is demonstrated in Figure 1, where the basic ADCS excludes the 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 if it is required.

The EKF is a method which incorporates a physics based model of the satellite dynamics as well as using sensor fusion and measurement updates to ensure accurate estimation. The sensor measurements that are used for the measurement update, are the sensors that provide a modelled vector in the orbit-referenced coordinate, *ORC*, frame as well as a measured vector in the satellite body coordinate, *SBC*, frame. The sensors used for the measurement update according to the general satellite design is a magnetometer, sun sensor and nadir sensor. The noise of the measurements and the noise of the system is incorporated in the EKF model to ensure

stability and reliable estimation. The general principal for measurement updates, is to update the EKF from the least to the most reliable sensor measurements. The error between the modelled and measured vectors are used to update the EKF. The EKF and the specific configuration thereof for satellites can be researched further [2].

The problem with an EKF is when the sensors do not follow their modelled vector. Slight deviations thereof won't have significant effects, but anomalies such as failed sensors can cause the EKF to become unstable. Consequently, we want to be able to recover from failed sensors. The frequency of the anomaly occurrence can also influence the stability of the Kalman filter. Therefore we opted to use sun reflection from the solar panel on sun sensors as our modelled anomaly. This is because of the accurate modelling for sun reflection. This is a real problem in the satellite industry that can be isolated with changes in the satellite design to ensure that sun reflection does not occur on the sun sensor.

This anomaly also requires autonomous decision making to ensure sun facing control. This cannot be done by the ground station during orbit, unless the control system is dramatically changed after the anomaly is detected by the ground station. 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.

A. Related Work

Sun reflection is not a new problem and has consequently been researched thoroughly. Hardware solutions to sun reflection have been developed with the use of digital sun sensors that can discriminate between direct sunlight and reflected sunlight. These digital sun sensors however, are not as accurate as many other analog sun sensors. There is also a long list of research that has been done with regards to sensor fault detection (as seen in [7]). One of the most relevant articles is done by Wang and Liang [7], and proposed a adaptive unscented Kalman filter for sensor fault estimation and isolation. The results thereof seems promising and are based on dramatical changes in the measurements. However, the research is not based on failures such as sun reflection, the results are based on drastic modelled failures. Therefore, future work will be to use the algorithm developed by Wang and Liang [7] to test the response thereof on sun reflection, noting that sun reflection occurs regularly during the sunlight phase of the orbit.
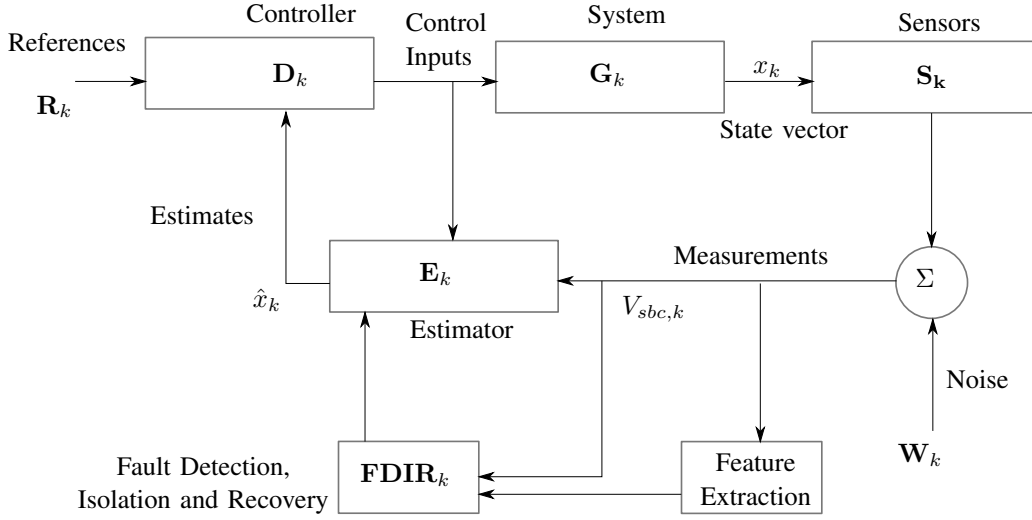
Fig. 1: System Diagram

Other research that is used for feature extraction in this article is based on research done by Silva et al. [6]. Fault prediction is done by implementing an innovative moving average, determined by the error estimated with dynamic mode decomposition, *DMD*, and a kalman filter, as an input to a predictive model — decision tree. The method however is adjusted for our use case to be a linear regression model instead of DMD.

Discuss the following article: [3], [9], [8].

B. Preliminaries

The details of satellite dynamics will not be discussed in this article, however it must be noted that the orbit-reference coordinate and satellite body coordinate frame will be referred to as *ORC* and *SBC* respectively. General notation of this article is matrices in upper case and bold, vectors in lower case and bold and scalars as lower or upper case but not in bold as illustrated below.

- Matrix **A**
- Vector $\mathbf{a} = \begin{bmatrix} x & y & z \end{bmatrix}$
- Scalar a or A

This will be the notation throughout this article unless specified otherwise.

II. REFLECTION

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

The assumption is made 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 also assumed that if any reflection from the solar panel is detected by the sun sensor, 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

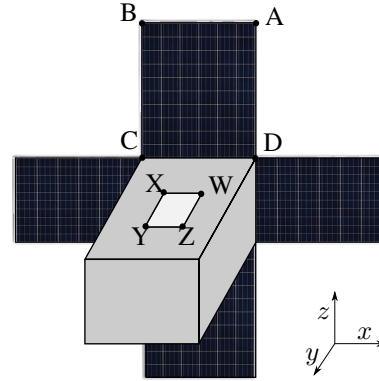$$\mathbf{r} = \mathbf{v} - 2\mathbf{n}^T(\mathbf{v} \cdot \mathbf{n}). \tag{1}$$

Fig. 2: Cube Sat

Where **v** is the incoming sun vector and **n** is the normal 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 equation for a plane can be denoted as

$$\mathbf{p} = ax + by + cz + d \tag{2}$$

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

$$\begin{aligned} x &= \alpha t \\ y &= \beta t \\ z &= \zeta t \end{aligned} \tag{3}$$

where the coefficients, α, β and ζ 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.
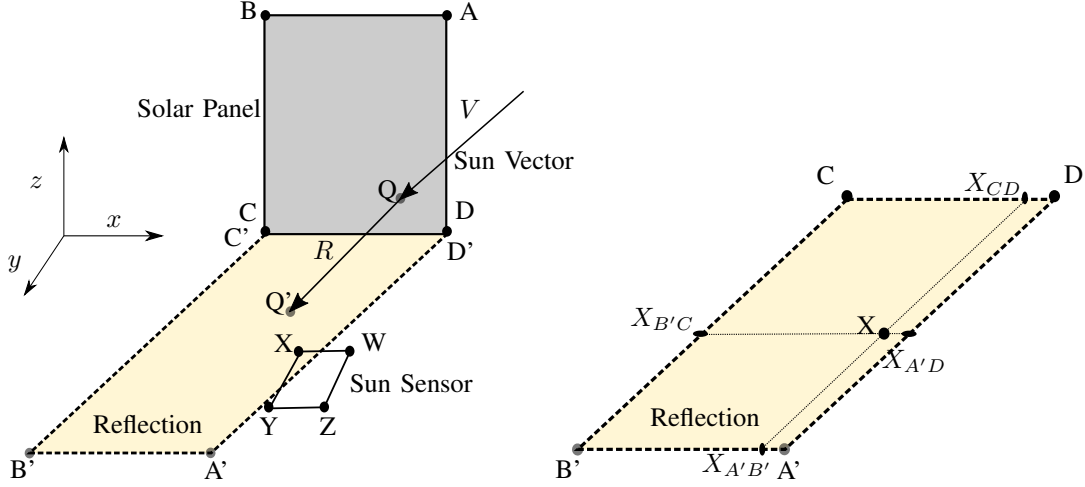
Fig. 3: Reflection

Thereafter, 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 taken into account. 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 exact same position as C, which is also true for D' and D, the calculation thereof 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 then determine whether Y or X is within the region of reflection, we assume that the plane $XYWZ$ 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 $B'C$ as well as between the lines CD and $A'B'$. By determining the line equation between reflected points in the form

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

the corresponding x or y coordinates can be calculated by substituting either X_y or X_x in Eq 6. With this the coordinates of $X_{B'C}$, $X_{A'D}$, $X_{A'B}$ and X_{CD} can be determined. Thereafter 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 effects the estimation and therefore also 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 be able to recover from sensor anomalies or to 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 a FDIR for kalman filter robustness is to detect whether an anomaly has occured on one of the filters. There are various different 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. [6] to detect failures in sensors.

The proposed method by Silva et al. [6] uses Dynamic Mode Decomposition (DMD), which was originally developed by Schmid et al. [5] and further expanded to include control by Proctor, Brunton, and Kutz [4], to provide an estimation of a sensor vector based on the previous measurement fo the sensor as well as the measurements of the other sensors in the system. DMD was first developed in the fluids community and constructs 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.
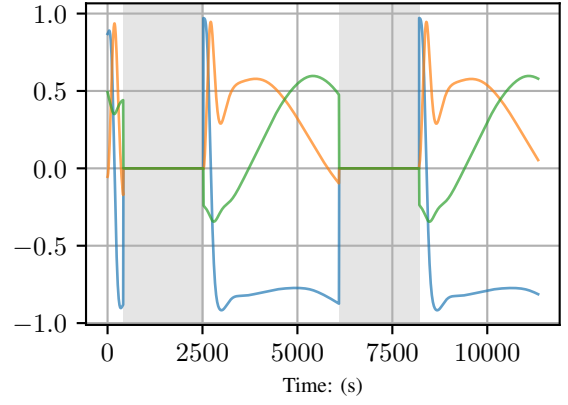
$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \quad (7)$$

Where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ over a time period 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
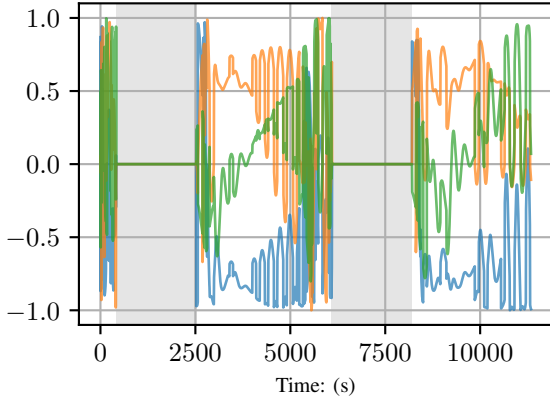
(a) Sun vector with reflection.

(b) Sun vector without reflection.

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

intensive. This is not the case for our system and using DMD is not justifiable. Therefore with the pseudo-inverse of $\mathbf{X}$, denote it as $\mathbf{X}^{\dagger}$, we calculate $\mathbf{A}$ as

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

This necessitates the required data for the state vector. The article by Silva et al. [6] however includes the $\mathbf{B}$ to relate the vector measurements of the other sensors to adjust the predicted state, X_{k+1} of the monitored sensor.

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

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 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. [6] the model is adjusted with a Kalman Filter. From $\mathbf{A}$ and $\mathbf{B}$ the Kalman filter can be implemented to predict $\mathbf{X}_{k+1}$

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

where $K = 0.001$. After the calculation of $\hat{\mathbf{X}}_{k+1}$ Silva et al. [6] proposes a moving average of the innovation covariance

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

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

The first step of FDIR is to detect whether an anomaly exists in the current sensor data with binary classification. For the proposed method, decision trees and random forests will be implemented to classify anomalies. A decision tree is a classification method that splits data samples 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 simple be done by determine whether the magnitude of the sun vector is equal to 0. The decision tree determines this split with the 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, root node. The Gini index provides a measure of the probability of a data sample being wrongly classified at a given node. This can be calculated with Eq 12.

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

The operator split that produces the lowest Gini index, provides the most pure split and will therefore be used as the root node. For our use case the CART algorithm will be used to optimize the decision tree, which also takes into account the largest 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 bias the decision tree is to the training data. Therefore, the depth is specified to 10 for our use case. This can be altered to optimize the decision tree, but we do not want to create an optimal solution for the simulation environment that will not behave well in a real environment.

It is evident in Figure 5 that the largest splits at the beginning of the tree is 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 the change in measurements.
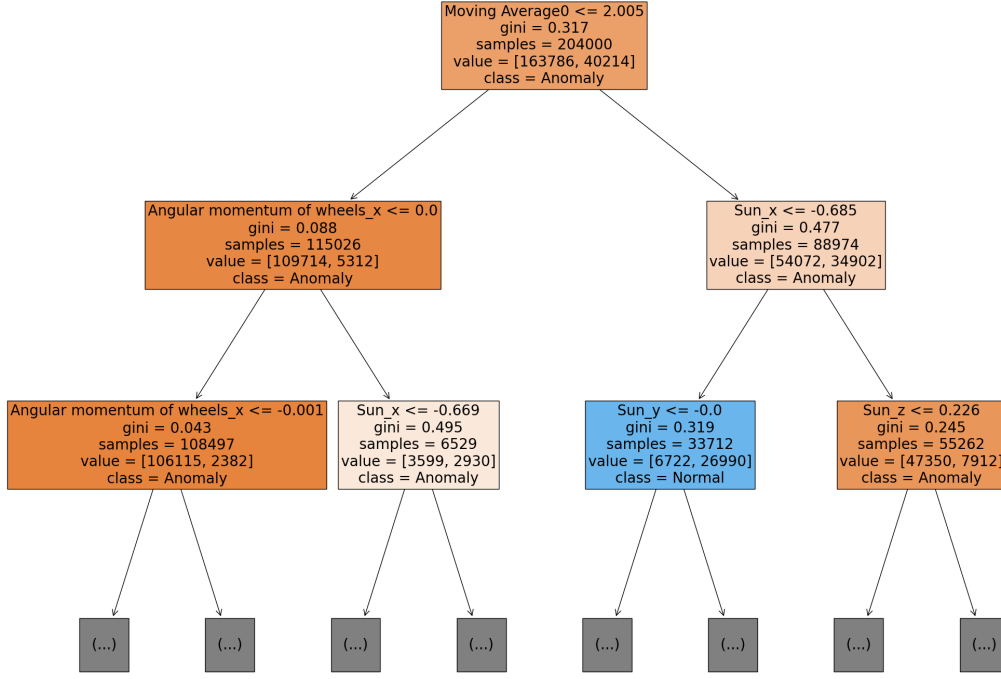
Fig. 5: Decision Tree

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 large 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 shown in section V.

C. Recovery

Three different methods of recovery are compared. These methods are all focused on ensuring that the sun reflection does not change the reliability and stability of the EKF.

The 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 backtrack 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 optimized 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 backtrack method can be combined with the ignore method. Where the backtrack method is implemented only after a specified number of sun reflections are predicted.

Another method implemented and tested always uses the two sensors measurements that has the smallest mean squared error between the estimated SBC vector and the actual measured SBC vector. There are 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 unto the drift in the sensor. The method will only detect sudden changes in the sensor, but will isolated the sudden change even if it remains stable after the sudden change. This method is will only compared to the other methods as part of the analysis, since it the method is inherently different.

IV. TESTING SETUP

To ensure repeatability of the tests conducted in this article, the sgp4 simulation environment is used for the position in orbit of the satellite. 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

Quaternion-feedback control with momentum dumping only during eclipse is implemented. 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 [1] can be calculated as

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

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 $\mathbf{s}_o$, δ, 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 ω_b^I is always $[0, 0, 0]$.

B. Dimensions of Satellite

The dimensions of the satellite are shown in Table I. The dimensions are shown to ensure the repeatability of the results in this article. The dimensions of the sun sensor are from the Sputnix cubesat sun sensor model.

TABLE I: Dimensions of Cubesat

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

C. Orbit Parameters

The orbit parameters will not significantly effect the results, however, for repeatability the general parameters for the orbit is given in Table II.

TABLE II: Parameters for Cubesat Orbit

Revolutions per day	15.2355
Inclination	97.4^o
Right ascension of the ascending node	275°

D. Sensors

The measurement update of the Kalman filter will firstly be with a magnetometer, 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 Guassian random noise and the magnetometer has the largest standard deviation and the sun sensor has the smallest standard deviation. There are two sun sensors, a coarse and fine sun sensor and both of them can 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 the fault 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 large 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 attitude and the estimated attitude in degrees.

A. Perfect Designed Satellite Without Reflection

This test is implemented for the current design with the assumption 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.
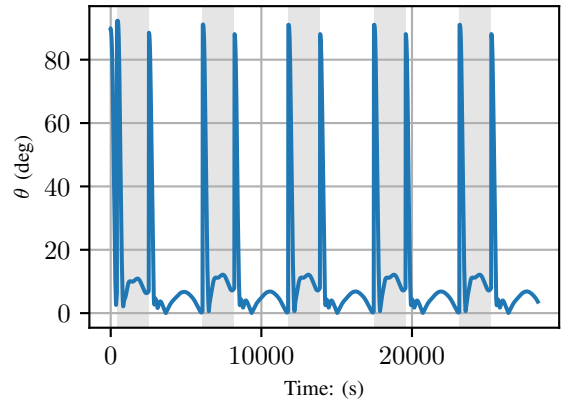

Fig. 6: Pointing Metric for Satellite Without Reflection.

Figure 6 evidently 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.
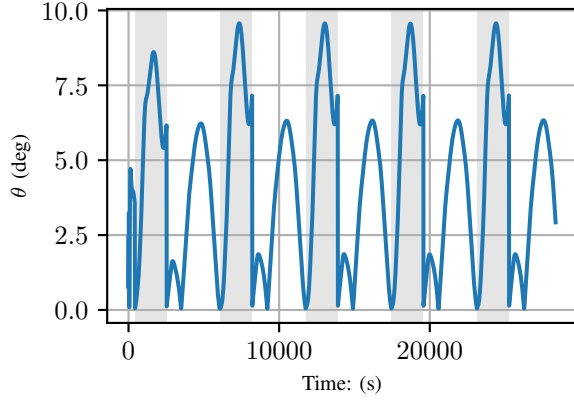
Fig. 7: Estimation Metric for Satellite Without Reflection.

Fig. 9: Estimation Metric for Perfect Detection.

B. Satellite With Reflection

If no recovery strategy is implemented for the reflection anomaly, the EKF is very 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 is also indicated in Figure 9. It is clear that the ADCS is unstable due to sun reflection.
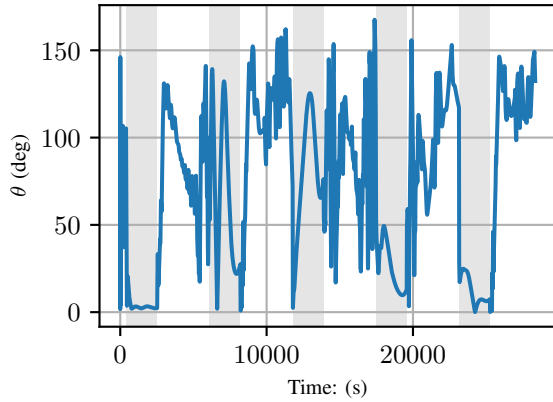

Fig. 8: Estimation Metric for Satellite with Reflection and without FDIR.

C. Perfect Detection

This section indicates 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.
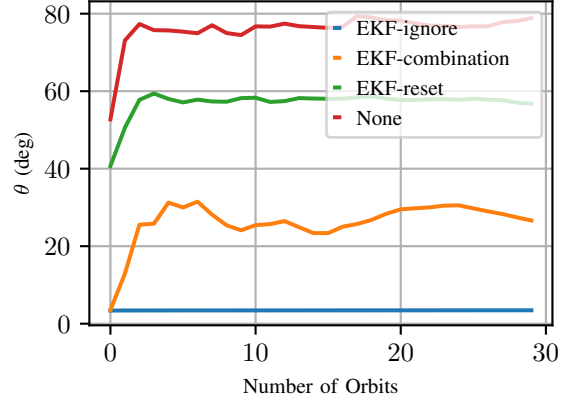
Due to the significant difference in performance between the EKF-ignore and the other recover methods, only the EKF-ignore method will be further investigated with other detection strategies. The EKF-ignore results on average 3.43

D. Satellite With Recovery Ignore

To show the promising results of the recovery ignore for perfect detection Figure 10 shows the estimation metric during the first 5 orbits. Our goal is to introduce detection strategies that produce similar results.
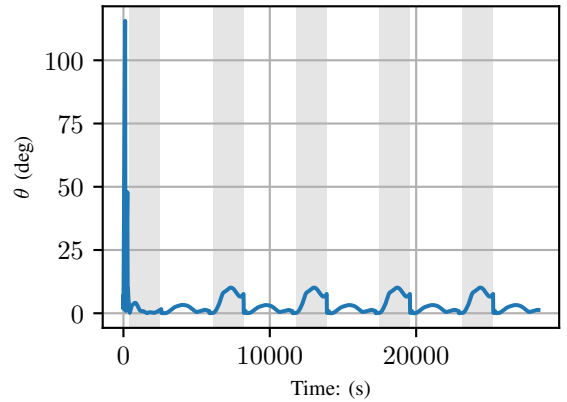

Fig. 10: Estimation Metric for Satellite with Recovery Ignore.

To perform an analysis of 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 a 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 is shown in Figure 11.

It is clear Figure 11 that the percentage accuracy to produce an estimation metric average of less than 20° per
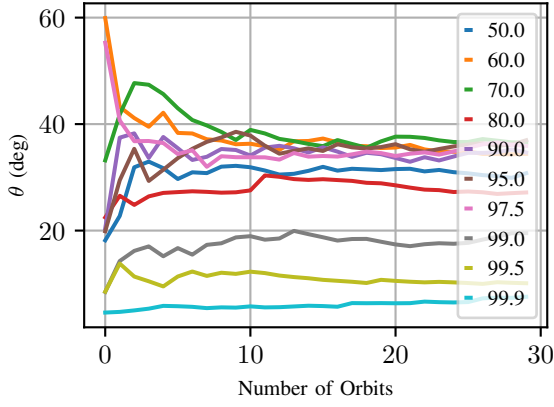
Fig. 11: Estimation Metric of Fixed Percentage Accuracies During Failures and No Failures.

orbit is 99%. This is a very difficult task to accomplish, therefore we want to determine what the effects of perfect prediction during no reflection time steps are. We perform a test where the prediction is only performed when reflection is implemented and during no reflection a 100% accuracy is used. The results therefore is shown in Figure 12, which clearly indicates that the largest effect on the EKF's estimation accuracy is not whether the no reflection time steps are predicted accurately, but rather that the reflection time steps are predicted correctly.
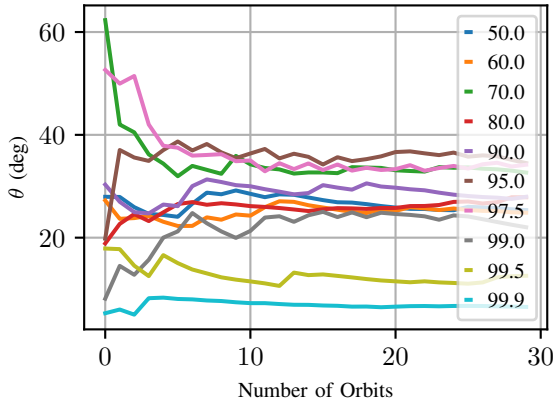

Fig. 12: Estimation Metric of Fixed Percentage Accuracies During Failures and Perfect Prediction During No Failures.

We therefore want to somehow make sure that more failures are accurately predicted. We therefore want to implement a first order filter after a failure is predicted to reduce the number of reflections that are incorrectly predicted. This however cannot be done by just ignore the sun vector for a period of time, since this will reduce the accuracy. This is then implemented in section V-E.

E. First Order Filter

A method that only uses the two sensors who's measurement is the closest to the estimated vectors from the EKF and the modelled ORC vectors is implemented. This method will be named EKF-top2 for ease of reference. This results of this method are shown in Figure 13. It is evident that the average estimation error is reduced to less than 20°. This method will consequently be used during a certain period after a failure is predicted. It cannot be used continually, since the overall accuracy is reduced by only using 2 sensors and assuming that there is always a failure.
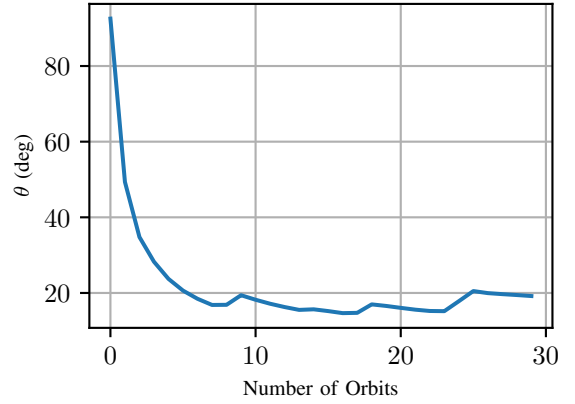

Fig. 13: Estimation Metric for Satellite with Recovery Ignore and Various Detection Methods.

Now we want to test what the estimation error is for various fixed accuracy percentages with the EKF-ignore method and then implementing 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.
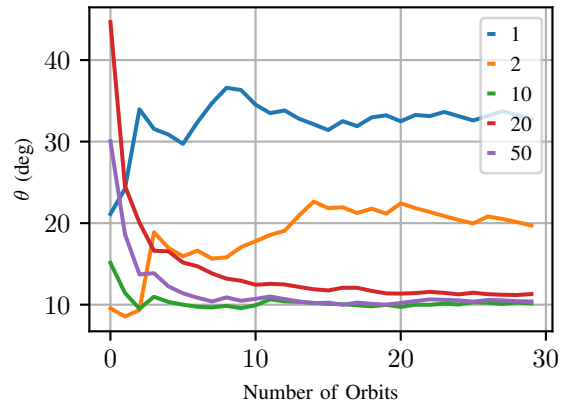

Fig. 14: Estimation Metric of 90.0% Accuracy with Various Time Delays.

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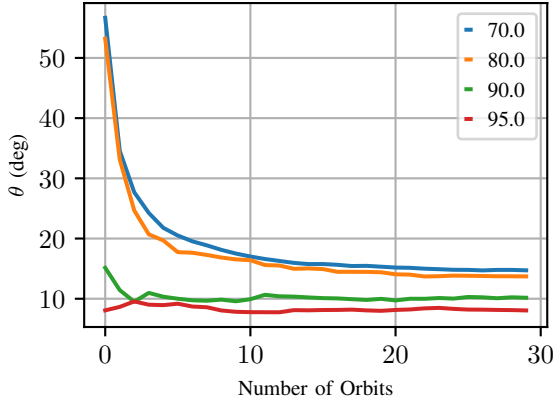

Fig. 15: Estimation Metric of Fixed Percentage Accuracies During Failures and No Failures with Recovery Buffer.

It is clear that the required percentage accuracy is lowered dramatically with the buffer implementation. A percentage accuracy of 70.0% with the buffer is performing similarly to a 99.5% accuracy without the buffer after 10 orbits. After we've now analysed the results for various fixed accuracies and various deviations of the the recovery methods we can now implement the detection strategies with the recovery methods and determine the efficiency of the

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 rather chosen than the depth 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 negative 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, the depth of 20 is used.

Using the 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 has a negative impact on the estimation error for the Decision Tree method, but slight decrease 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, but it is clear that they have higher estimation errors.
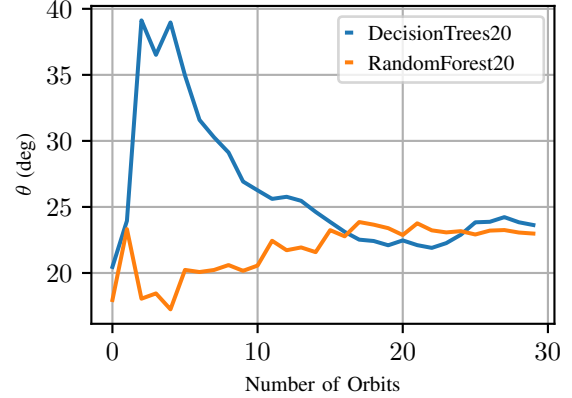

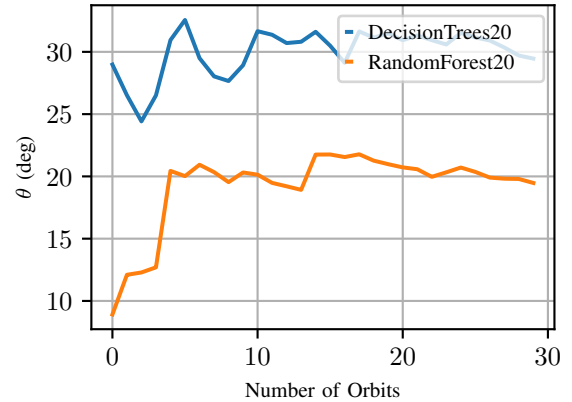Fig. 16: Estimation Metric of Decision Tree and Random Forest.

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

To determine the behaviour of the detection strategies we plot the prediction accuracy during the first 5 orbits, but we do not plot the mean, but the actual prediction accuracy. This is used to determine the time gaps between incorrect predictions, which would vary from the fixed accuracies. For this analysis we isolate the Random Forest detection strategy, since it out significantly out performs the Decision Tree method.

From Figure 17 we see that the estimation metric for the first orbit is an average of 8.9^o, which increases to 12.1^o in

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	

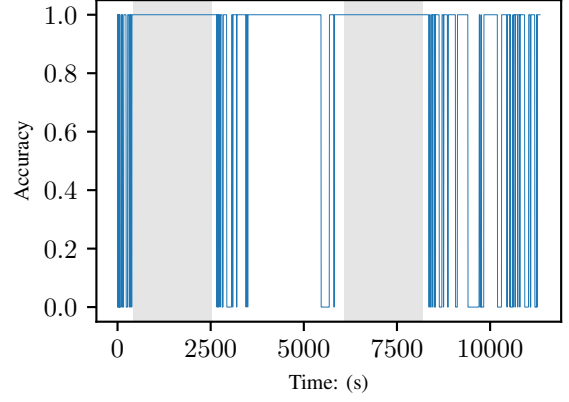Fig. 18: Prediction Accuracy of Decision Tree and Random Forest with Recovery Buffer.

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

the second orbit and by the 5th orbit the estimation error has increased to 20.8^o. 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 86% accuracy. Secondly, the duration of inaccurate predictions increases as seen in the horizontal line at 0. This large durations of constant inaccurate prediction will increase the instability and decrease the accuracy of the EKF. The reason for this inaccurate predictions is due to the change of the satellite behaviour during recovery and this 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 based on n-dimension planes.

VI. CONCLUSIONS

It is clear from the results from the analysis that recovery of sun reflection 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 the decrease in accuracy of the prediction after the recovery method is implemented. It is therefore recommend to further the testing with methods such as Support Vector Machines.
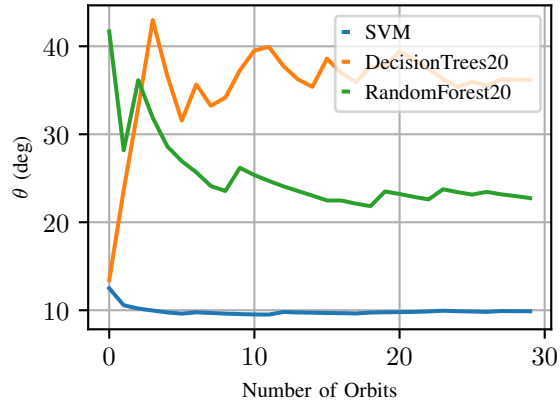
Fig. 20: Prediction Accuracy of Decision Tree and Random Forest with Recovery Buffer.
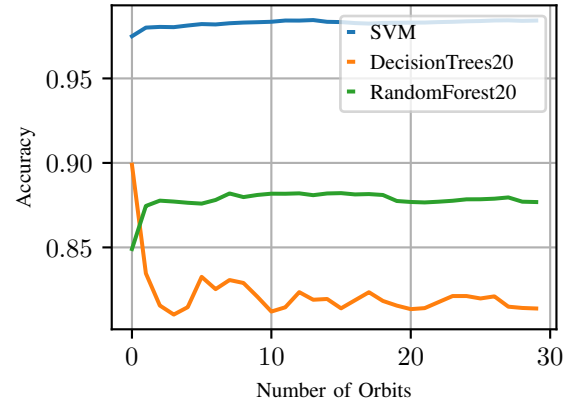

Fig. 21: Prediction Accuracy of Decision Tree and Random Forest with Recovery Buffer.

REFERENCES

[1] 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.

[2] Gerhard Hermann Janse van Vuuren. "the Design and Simulation Analysis of an Attitude Determination and Control System for a Small". In: March (2015).

[3] Seiied 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.

[4] 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.

[5] Peter J Schmid et al. "Applications of the dynamic mode decomposition". In: *Theoretical and Computational Fluid Dynamics* 25.1 (2011), pp. 249–259.

[6] 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.

[7] 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.

[8] 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.

[9] 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.