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.**

***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 in the actual system. Therefore a probabilistic approach can be used to determine the state vector from both the model and the sensor. 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 such as the 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 measurements. The error between the modelled and measured vectors are used to update the EKF estimation. 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 determine 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 if the anomaly is detected by the ground station. Therefore we aim to design a fault detection, isolation and recovery system specifically. The specific use case will be 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 [6]). One of the most relevant articles is done by Wang and Liang [6], 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 does not seem to include sensor fusion, but only requires the true estimation from the single occurence of an anomaly in a sensor measurement. Therefore, future work might be to use the algorithm developed by Wang and Liang [6] to test the response thereof on sun reflection, noting that sun reflection occurs regularly during the sunlight phase of the orbit as well as the fact that normal satellite operations use sensor fusion to create an accurate estimation.

Other research that is used for feature extraction in this article is based on research done by Silva et al. [5]. Fault

*This work was not supported by any organization

[1]Louw UJ is with Faculty of Electronic & Electrical Engineering, Electronic System Laboratory, University of Stellenbosch, Stellenbosch Central, Stellenbosch, 7600 louwuj@gmail.com

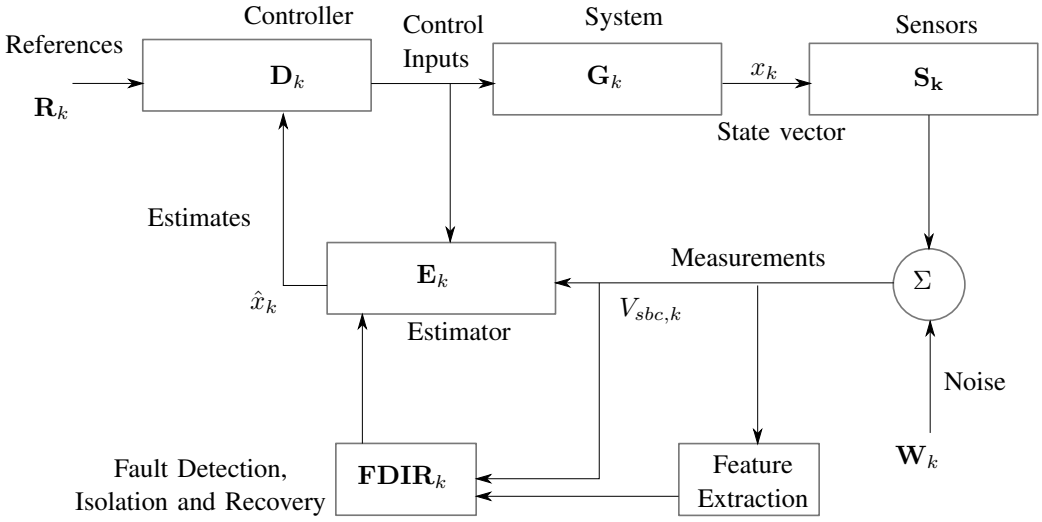

Fig. 1. System Diagram

prediction is done by implementing an innovative moving average, determined by the error estimated with dynamic mode decomposition, *DMD*, and a kalman filter, as a 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.

B. Preliminaries

The details of satellite dynamics will not be discussed in this article, however it must be noted that 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 $\mathbf{A}$
- 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 any dimensions but with 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 plane. Therefore light from the solar panel will reflect similarly to 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

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

Where $\mathbf{v}$ is the incoming sun vector and $\mathbf{n}$ is the normal vector to the plane $ABCD$ of the solar panel as seen in

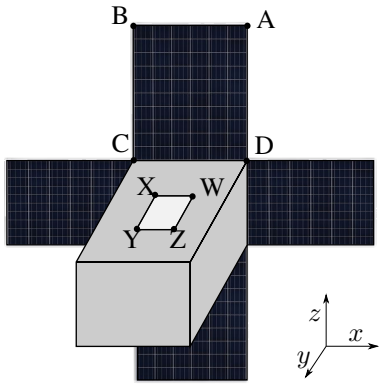

Fig. 2. Cube Sat

Figure 2. To calculate the intersection of the reflected vector with the plane $XWYZ$ of the sun sensor the intersecting point the equation of the plane, reflected vector 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 $x, yandz$ 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.

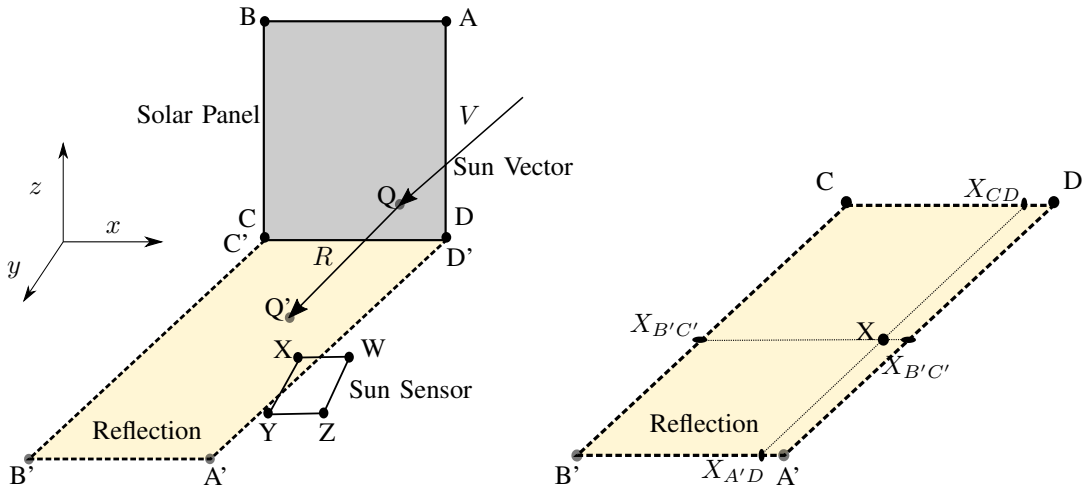

Fig. 3. Reflection

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

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

where o_1, o_2, o_3 is the position 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' can be calculated as

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

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

$$y_{A'B'} = mx_{A'B'} + c \tag{6}$$

the corresponding x or y coordinate 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. This is demonstrated with the position X. 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. For modelling purposes, the reflection in this example has no influence on the estimation and control of the satellite.

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. [5] to detect failures in sensors.

The proposed method by Silva et al. [5] uses Dynamic Mode Decomposition (DMD), which was originally developed by Schmid et al. [4] and further expanded to include control by Proctor, Brunton, and Kutz [3], 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.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \tag{7}$$

Where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ over a time period will be denoted as $\mathbf{X}$ and $\mathbf{X}'$ respectively.

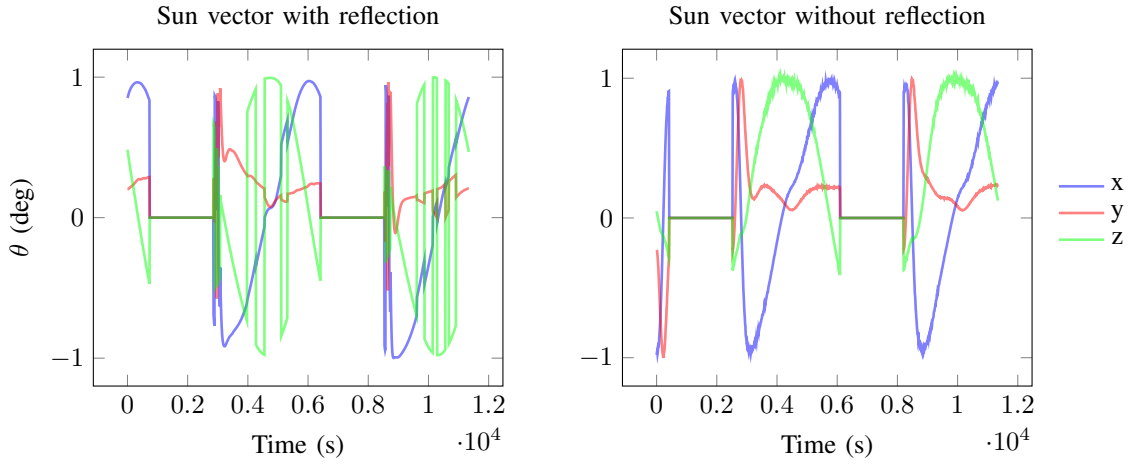

Fig. 4. Sun vector without reflection.

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. Therefore with the pseudo-inverse of $\mathbf{X}$, denote it as $\mathbf{X}^{\dagger}$, we calculate $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{X}\mathbf{X}^{\dagger} \tag{8}$$

This necessitates the required data for the state vector. The article by Silva et al. [5] 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.

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

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

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

where $K = 0.001$. After the calculation of $\hat{\mathbf{X}}_{k+1}$ Silva et al. [5] proposes a moving average of the innovation covariance

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

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 being wrongly classified at a given node. This can be calculated with Eq 12.

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

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 III-B 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.

It is evident in Figure III-B that the best splits at the beginning of the tree is the sun sensor measurements as well as the moving average from the feature extraction. This makes logical sense, since the sun reflection should firstly be detected on the sun sensor measurements and the moving average indicates the changes in the linear regression model that correlates the sensors measurements.

Random Forests, an method of using the prediction average of randomly sampled decision trees, is also tested and

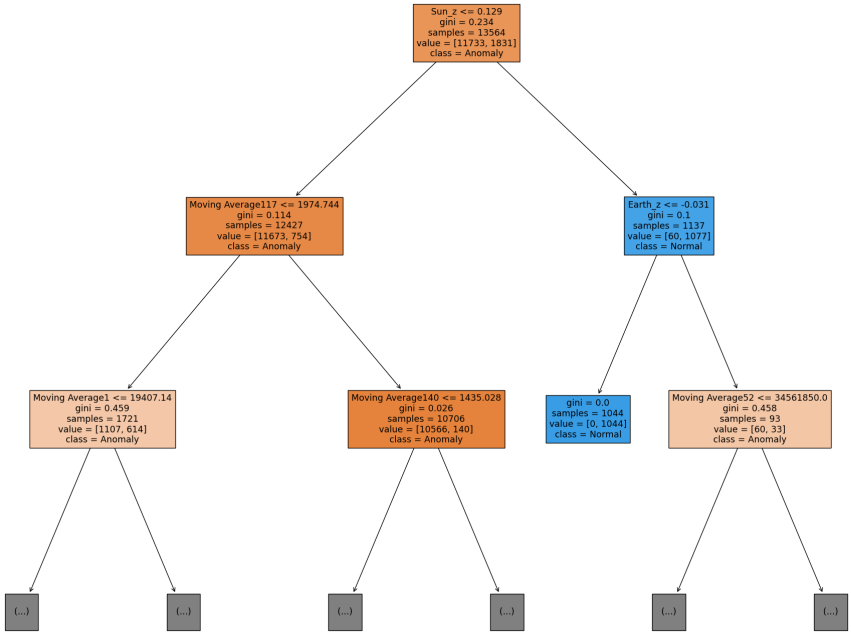

Fig. 5. Decision Tree

the results thereof shown in section V.

C. Recovery

Four 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 an detection method with low accuracy will create cause instability to the EKF, since many anomalous measurements will be included in the measurement update of the EKF.

The replacement methodology changes $v_{meas,k}$ to $v_{est,k}$ at the timestep when the failure is detected. This method depends on the stability and accuracy of the EKF when the failure is detected and highly depends on the accuracy of the detection method. Although this seems to bypass the entire purpose of a measurement update, and might change the EKF's dependency to be more on the sensor than the model, even though the sensor measurement might not be accurate. The EKF will remain stable due to the other measurements being accurate and will save computation time. The EKF will not require any reset and the same number of measurements updates will still occur during a sensor's anomalous behaviour.

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.

IV. Testing Setup

To ensure repeatability of the tests conducted in this article, the github repo is provided ... 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. 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

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

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

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

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.

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^o

D. Sensors

The measurement update of the Kalman filter firstly 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^o. 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 show the theoretical possibilities of the recovery method.

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 EKF.

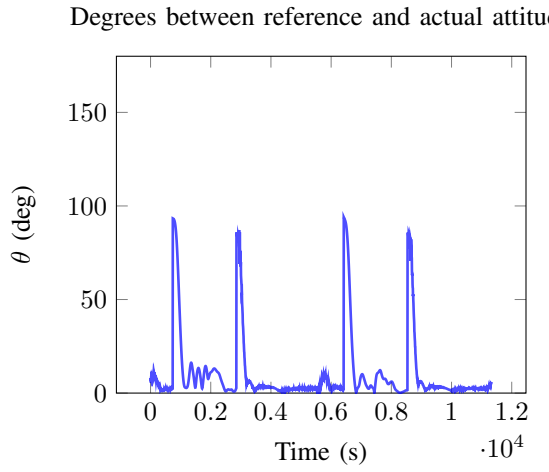

Fig. 6. Pointing Accuracy.

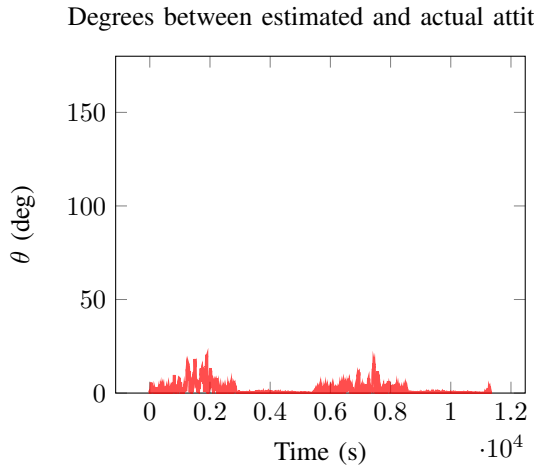

Fig. 7. Estimation Accuracy.

B. *Satellite With Reflection*

If no recovery strategy is implemented for the reflection anomaly, the EKF produces a singular matrix. Therefore there are no results for the satellite with reflection, without any recovery strategy.

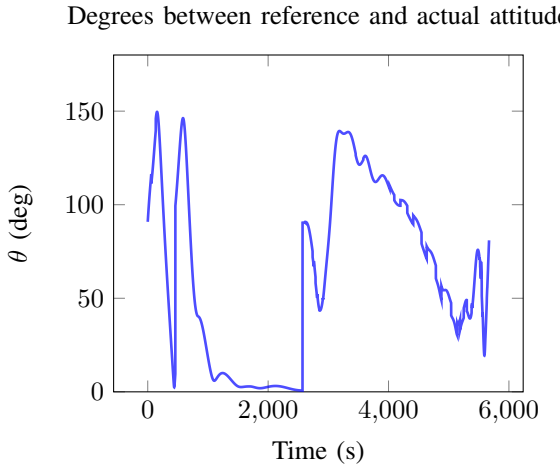

Fig. 8. Pointing Accuracy.

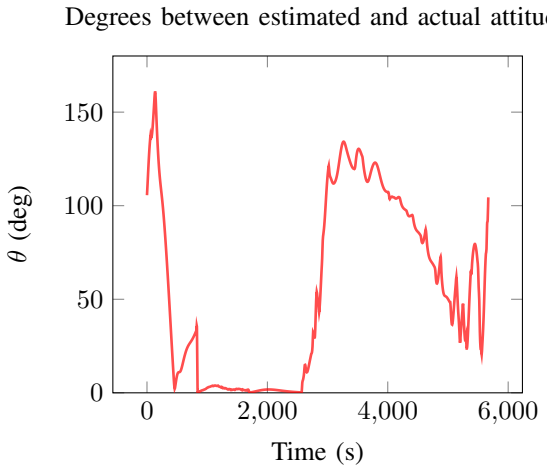

Fig. 9. Estimation Accuracy.

C. Perfect Prediction

This section indicates the results for each recovery method, with the assumption that the detection method is perfect. This means that the prediction methodology will produce an accuracy of 100%.

1) Satellite With Recovery Ignore: For the recovery method where the sun sensor is ignored when a failure is detected, the results are very promising. This indicates that if the prediction methodology is very accurate then the EKF will be stable and the influence of the sun reflection will be minor, and will only change the vector which the control system will follow for the sun following part of the orbit.

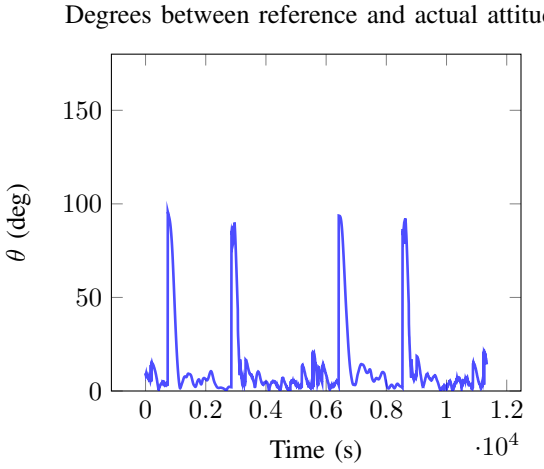

Fig. 10. Pointing Accuracy.

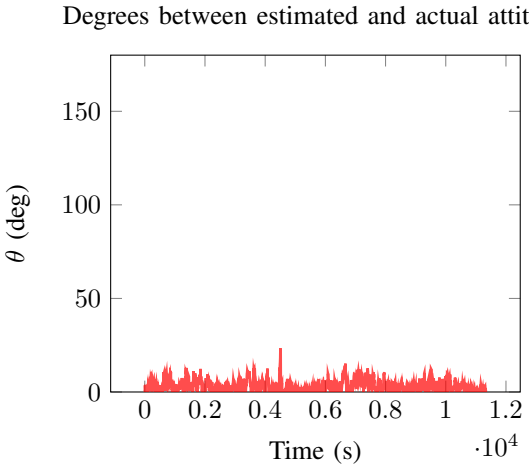

Fig. 11. Estimation Accuracy.

Insert a table to compare random orbit parameters. The mean, standard deviation of each orbit 0-20 for each of the different strategies of reflection.

VI. CONCLUSIONS

Results from kalman filter and attitude determination as well as control compared for EKF with and without FDIR.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

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] 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.

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

[5] 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.

[6] 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.

TABLE III

POINT METRIC FOR VARIOUS METHODS

Orbits		1		2		3		4		5		30	
Detection Strategy	Recovery Strategy	Metric (θ)											
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Random Choice	EKF-Ignore	15,38	25,92	13,92	23,50	13,27	22,39	13,18	21,86	13,01	21,42	12,23	20,47
	EKF-combination	25,01	28,08	28,37	29,99	29,31	30,40	28,66	31,03	30,25	32,99	31,97	34,55
	EKF-reset	92,61	34,38	87,20	38,33	81,89	37,87	79,54	36,59	79,42	38,40	88,96	36,74
	EKF-replacement	61,97	48,24	54,56	45,60	59,89	47,24	58,61	47,75	59,22	47,47	59,02	46,89
Decision Trees	EKF-Ignore	**21,56**	**32,98**	**17,14**	**26,77**	**15,42**	**24,66**	**14,50**	**23,58**	**14,03**	**22,88**	**12,44**	**20,79**
	EKF-combination	21,56	32,98	17,14	26,77	15,42	24,66	14,50	23,58	14,12	22,89	13,20	21,61
	EKF-reset	42,37	35,01	56,60	38,50	54,78	38,56	56,97	41,06	58,16	40,17	60,96	41,25
	EKF-replacement	57,02	46,86	57,21	47,58	55,36	47,36	58,10	47,91	57,20	47,35	56,43	46,36
Random Forest	EKF-Ignore	**16,05**	**25,96**	**13,93**	**23,15**	**13,34**	**22,23**	**13,02**	**21,90**	**12,91**	**21,62**	**12,19**	**20,52**
	EKF-combination	16,05	25,96	13,93	23,15	13,34	22,23	13,02	21,90	12,91	21,62	12,95	21,37
	EKF-reset	61,26	44,09	50,87	42,30	52,37	41,70	45,62	38,93	50,04	39,92	41,04	36,16
	EKF-replacement	57,49	47,73	57,18	45,66	58,24	46,50	57,42	47,09	57,42	48,02	56,95	47,28
Perfect	EKF-Ignore	**14,71**	**25,85**	**13,42**	**23,17**	**12,89**	**22,25**	**12,57**	**21,59**	**12,45**	**21,26**	**12,19**	**20,38**
	EKF-combination	14,71	25,85	13,42	23,17	12,89	22,25	12,57	21,59	12,66	21,47	12,90	21,11
	EKF-reset	48,61	39,05	67,01	38,87	65,83	37,91	70,21	37,00	68,01	36,89	68,89	40,58
	EKF-replacement	26,76	28,44	24,25	26,90	24,23	27,60	24,04	27,66	22,81	26,40	22,86	27,47
None	Perfect Design	**15,54**	**26,75**	**13,08**	**23,18**	**12,39**	**22,13**	**11,87**	**21,61**	**11,64**	**21,4**	**10,59**	**20,38**
	Failure Design	37,63	36,57	42,27	32,76	43,55	32,14	42,97	32,00	42,33	31,93	N/A	N/A

TABLE IV

ESTIMATION METRIC FOR VARIOUS METHODS

Orbits		1		2		3		4		5		30	
Detection Strategy	Recovery Strategy	Metric (θ)											
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Random Choice	EKF-Ignore	**3,15**	**2,92**	**3,09**	**3,06**	**3,10**	**3,02**	**3,23**	**3,13**	**3,26**	**3,14**	**3,22**	**2,99**
	EKF-combination	7,93	11,43	10,23	16,61	10,43	19,07	10,59	19,42	11,56	20,38	13,29	24,23
	EKF-reset	60,50	44,21	59,13	43,68	58,21	43,39	58,14	43,24	57,91	43,08	60,22	42,99
	EKF-replacement	48,60	52,08	41,17	45,62	45,27	47,61	46,04	48,88	46,37	48,92	45,07	47,11
Decision Trees	EKF-Ignore	**13,07**	**32,63**	**8,29**	**17,91**	**6,56**	**12,85**	**5,58**	**10,21**	**5,09**	**8,76**	**3,48**	**3,80**
	EKF-combination	13,07	32,63	8,29	17,91	6,56	12,85	5,58	10,21	5,15	8,87	3,82	4,93
	EKF-reset	18,48	25,62	28,41	32,39	29,68	33,56	31,77	36,11	33,02	37,55	33,56	38,05
	EKF-replacement	40,77	44,19	41,47	43,24	40,21	42,96	42,77	44,46	42,33	45,04	42,29	45,18
Random Forest	EKF-Ignore	**5,63**	**12,52**	**4,19**	**7,51**	**3,83**	**5,97**	**3,70**	**5,22**	**3,55**	**4,76**	**3,25**	**3,20**
	EKF-combination	5,63	12,52	4,19	7,51	3,83	5,97	3,70	5,22	3,55	4,76	3,44	3,81
	EKF-reset	31,44	39,63	24,32	32,66	27,96	34,90	23,43	30,93	25,80	31,38	19,44	26,35
	EKF-replacement	46,93	49,43	43,24	47,37	44,57	48,48	43,80	47,24	43,08	47,21	42,38	46,49
Perfect	EKF-Ignore	**3,44**	**4,95**	**3,21**	**3,77**	**3,20**	**3,48**	**3,17**	**3,37**	**3,14**	**3,31**	**3,21**	**2,97**
	EKF-combination	3,44	4,95	3,21	3,77	3,20	3,48	3,17	3,37	3,18	3,32	3,37	3,49
	EKF-reset	34,33	41,81	43,44	42,07	39,20	38,85	43,24	39,39	41,44	39,28	39,17	38,82
	EKF-replacement	12,60	15,44	11,45	16,04	11,26	16,80	11,36	17,88	10,83	16,69	11,47	18,34
None	Perfect Design	**1,87**	**2,77**	**1,85**	**2,72**	**1,89**	**2,83**	**1,85**	**2,71**	**1,82**	**2,72**	**1,70**	**2,50**
	Failure Design	**43,65**	**49,94**	**47,28**	**46,57**	**52,50**	**46,81**	**53,68**	**46,63**	**52,33**	**46,20**	**N/A**	**N/A**