Fault Detection to Increase Reliability of Kalman Filter for Satellite Attitude Determination

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 could increase the stability and reliability of the Kalman filter for satellite attitude determination.**

I. INTRODUCTION

The Extended Kalman filter... Various prediction methodologies... Various sensor anomalies... (specific anomalies such as solar reflection).

II. RELATED WORK

A. Extended Kalman Filter

The implementation of the estimated kalman filter *EKF* is for estimation of the current satellite attitude with sensor fusion of the magnetometer, star tracker, sun sensor and nadir sensor to accurately estimate the attitude and rotation rate of the satellite. The EKF will be used due to the non-linear nature of the system. The EKF consists of two fundamental parts, the model update and the measurement update. The general form for a system model can be expressed as

$$\dot{x_t} = \mathbf{f}(x_t) + s_t \tag{1}$$

where $\mathbf{f}(x_t)$ is a non-linear function of x_t. The state vector, x, for the full 7-state EKF consists of the quaternion vector, q and the inertial-referenced angular velocity, ω_B^I.

$$x = [q, \omega_B^I]^T \tag{2}$$

The estimated state vector x will be denoted as $\hat{x}$ and the estimated vector before and after the measurement update will be indicated with a superscript $'-'$ and $'+'$ respectively. To calculate the model update the dynamics and kinematics of the system model is used to calculate both ω_B^I and q. Where the euler dynamic equation for $\dot{\omega}_B^I$ and the known torques of the system can be expressed as

$$\delta\hat{\omega}_B^I = \mathbf{J^{-1}}\big((N_m)_{k-1} - (N_w)_{k-1} - (N_{gg})_{k-1} - (N_{gyro})_{k-1}\big) \tag{3}$$

The integration method used in the simulation is the 4th order Rungka-Kutta method to solve the differential equations.

Algorithm 1 Multi-variate Guassian Distribution Algorithm

```
Timestep (T_s) = 1s
Number of iterations (h) = 10
N = T_s / h
for n ∈ N do
    k_1 = h f(x_n, y_n)
    k_2 = h f(x_n + h/2, y_n + k_1/2)
    k_3 = h f(x_n + h/2, y_n + k_2/2)
    k_4 = h f(x_n + h, y_n + k_3)
    y_{n+1} = y_n + k_1/6 + k_2/3 + k_3/3 + k_4/6
end for
return y_{n+1}
```

where h is the step height, which is set to $T_s/10$, $T_s = 1$.

B. Sensitivity of Kalman Filter

The effect of sensor anomalies on kalman filter.

III. SIMULATION

Proof of simulation validity, use actual data and simulated data to plot comparative graphs.

A. Dimensions of Satellite

IV. SENSORS

V. SENSOR ANOMALIES

List of sensor anomalies and a model of specific anomalies. (solar panel reflection)

A. Reflection

‘ Previous work done by Cilden-Guler et al. [1] provides models that determine albedo effects from the earth and adjust the CSS measurements to improve accuracy. he assumption is made that the solar panel can be modelled as a simple plane. Therefore light that hits the solar panel will reflect as if it hits a perfectly smooth mirror. It is also assumed that if any reflection from the solar panel hits the sun sensor, the sun sensor will then default to the reflection ray instead of the modelled sun vector. The reflected sun vector, R, can be calculated as

$$R = V - 2N^T(V \cdot N) \tag{4}$$

Where V is the incoming sun vector and N is the normal vector to the plan $ABCD$ of the solar panel as seen in

*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

Figure 1. 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

$$P = ax + by + cz = d \tag{5}$$

The reflected vector can also be translated to equations of

$$\begin{aligned} x &= \alpha t \\ y &= \beta t \\ z &= \zeta t \end{aligned} \tag{6}$$

Since we can calculate the coefficients for Eq 6 from the reflected vector, we can calculate t, by substituting $x, y\&z$ into Eq 5. Consequently the intersecting points x, y and z can be calculated as

$$P(x, y, z) = (o_1 + \alpha t, o_2 + \beta t, o_3 + \zeta t) \tag{7}$$

where o_1, o_2, o_3 is the points of origin, in this case the points on the solar panel.

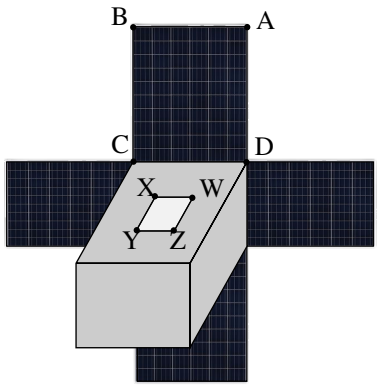


Fig. 1. Cube Sat

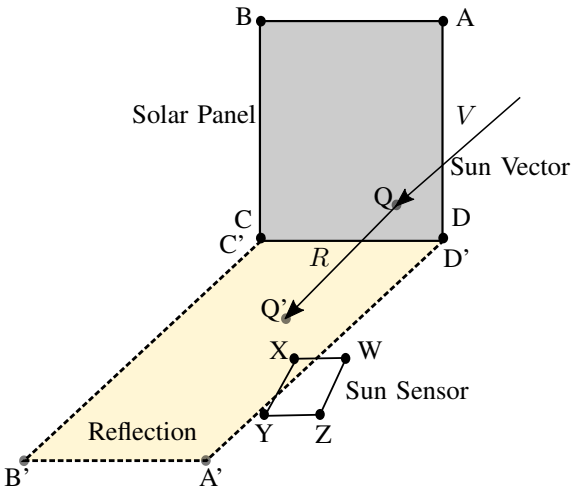


Fig. 2. Reflection

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 can be taken into account. From Figure 2 it is evident that if the solar panel reflects on corner y that corner x will also receive light from the reflection. The same is true for corner z and w. Since C' will be at the exact same position as C, the reflection from the sun does not need to be calculated for C, this is also true for D. Therefore it is only necessary to calculate the reflected positions A' and B'.

The reflected position A' can be calculated as the intersection of the reflected vector RR' with plane $xyzw$ with Eq 7. We also know the position of A, based pon the dimensions of the solar panel, consequently we can calculate A'. The same applies to B and B'. To then determine whether y or x is within the region of reflection, we calculate whether x is between the lines of $A'D'$ and $B'C$ as well as between the lines ACD and $A'B'$. This is done by calculating the coordinates of $X_{B'C}$, $X_{A'D}$, $X_{A'B}$ and X_{CD} to determine if X is in the reflection zone, which can be done with logical if statements.

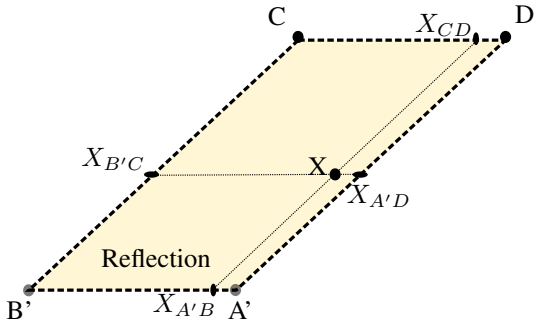


Fig. 3. Reflection

The results for the sun vector with and without reflection is shown in Figure... For modelling purposes, the reflection in this example has no influence on the estimation and control of the satellite.

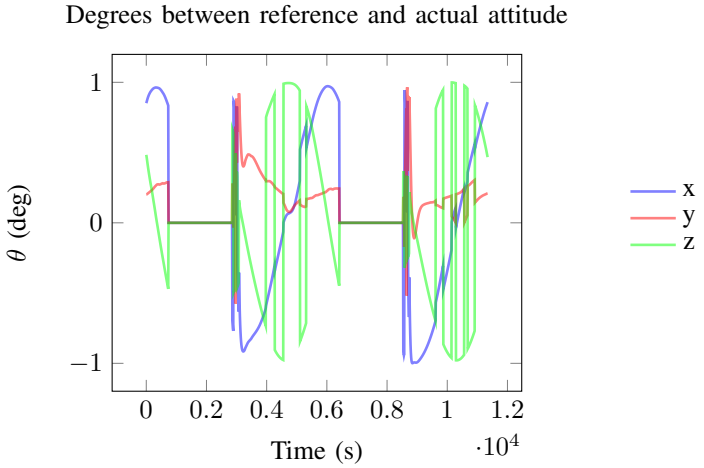


Fig. 4. Sun vector.

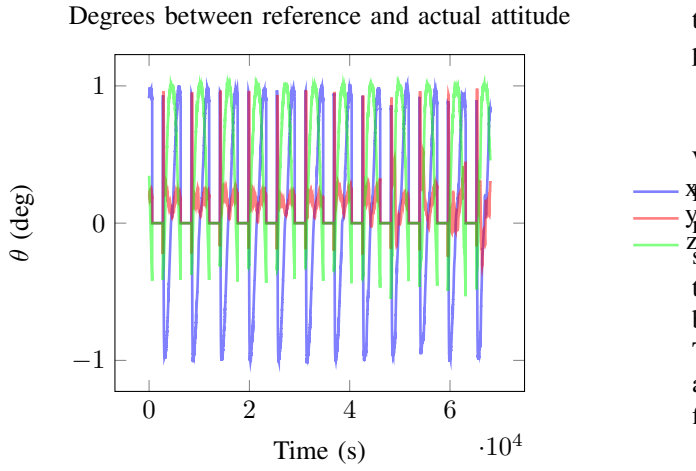


Fig. 5. Sun vector.

VI. 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. [4] to detect failures in sensors.

The proposed method by Silva et al. [4] uses Dynamic Mode Decomposition (DMD), which was originally developed by Schmid et al. [3] and further expanded to include control by Proctor, Brunton, and Kutz [2], 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.

$$x_{k+1} = \mathbf{A}x_k \tag{8}$$

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

$$\mathbf{A} = \mathbf{X}\mathbf{X}^\dagger \tag{9}$$

This necessitates the required data for the state vector. The article by Silva et al. [4] 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.

$$x_{k+1} = \boldsymbol{A}x_k + \boldsymbol{B}y_k \tag{10}$$

Where y_k is the other sensor measurements. This is adjusted for our use case, where y_k is the control inputs for the magnetotorquers and reaction wheels and x_k is all of the sensor measurements. Consequently, the model of 10 denotes the prediction of the sensor measurements in time step $k+1$ based on the current sensor measurements and control inputs. Thereafter, as implemented by Silva et al. [4] the model is adjusted by with a Kalman Filter. From $\boldsymbol{A}$ and $\boldsymbol{B}$ the Kalman filter can be implemented to predict x_{k+1}

$$\hat{x}_{k+1} = A\hat{x}_k + By_k + K(x_k - \hat{x}_k) \tag{11}$$

After the calculation of $\hat{x}_{k+1}$ Silva et al. [4] proposes a moving average of the innovation covariance

$$\boldsymbol{V}_k = \frac{1}{N}\sum_{i=k-N}^{k}(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T \tag{12}$$

The moving average is used as an additional input parameter for the classification of anomalies based on the $\boldsymbol{x_k}$.

B. Classification

The first step of FDIR is to classify whether an anomaly exists in the current sensor data. For the proposed method, decision trees 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 to split a data samples from both a satellite orbit based on being in an eclipse or not, would be simply to measure whether the sun vector is 0. With the assumption that the sun sensor has no anomalous behaviour.

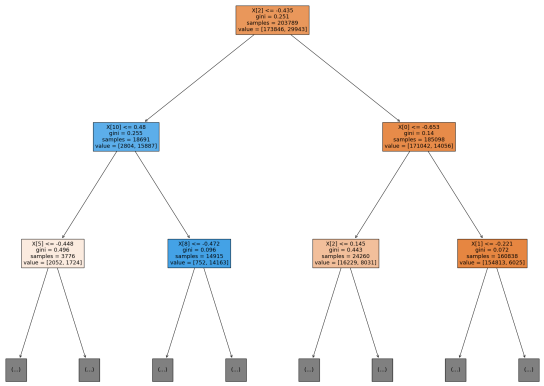
However to split the data for the anomalies we need to decide which input parameter will be used to make the first split, root node. The Gini index provides a measure of the probability of a data sample being being wrongly classified at a given node. This can be calculated with Eq 13.

$$GI = 1 - \sum_{i=1}^{n}(P_i)^2 \tag{13}$$

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

C. Recovery

Backtracking on kalman filter... Removing sensor measurement update... If a fault is predicted, the estimated vector model from the kalman filter is used to to determine the reference position for the control as well as the calculations for the control.



VII. Testing Methodology

Simulation... Induce anomalies based on physical models or at a specified timestamp. J

VIII. Results

If sun sensor is the last sensor to be updated in the measurement then a singular matrix error occurs.

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 possibilities of the recovery method.

The simulation is run for 20 orbits, with each orbit running for 5700s. If a fault is induced, it is induced after the first two orbits and the specific anomaly then occurs for the following 18 orbits.

A. *Perfect Designed Satellite Without Reflection*

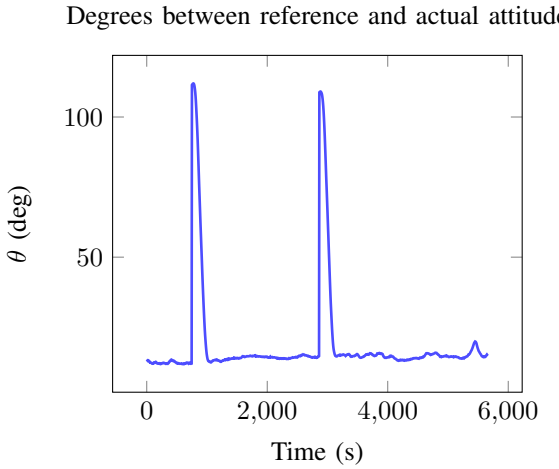


Fig. 6. Pointing Accuracy.

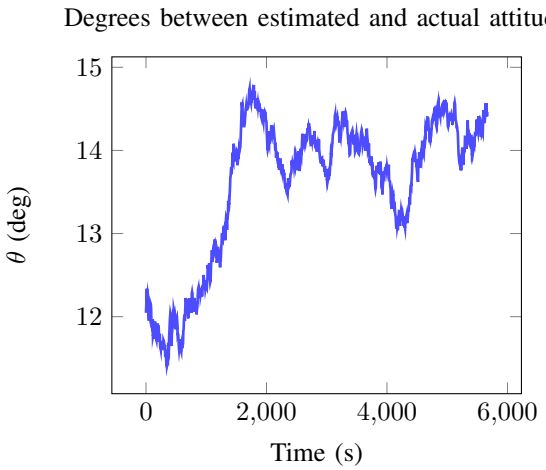


Fig. 7. Estimation Accuracy.

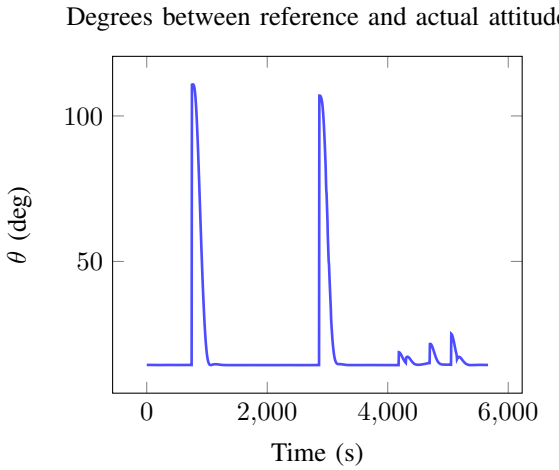


Fig. 8. Pointing Accuracy.

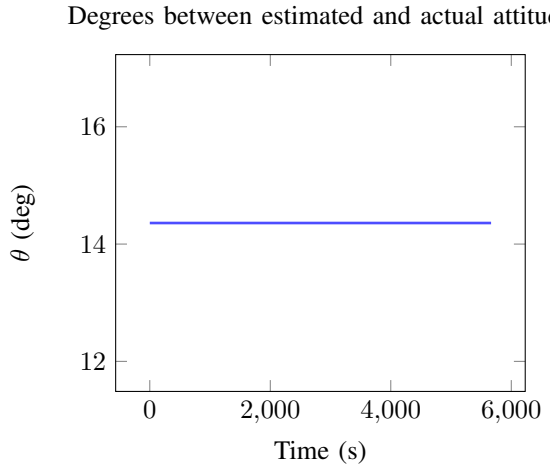


Fig. 9. Estimation Accuracy.

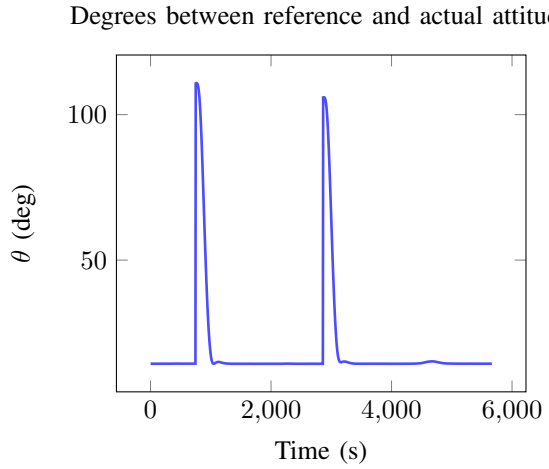


Fig. 10. Pointing Accuracy.

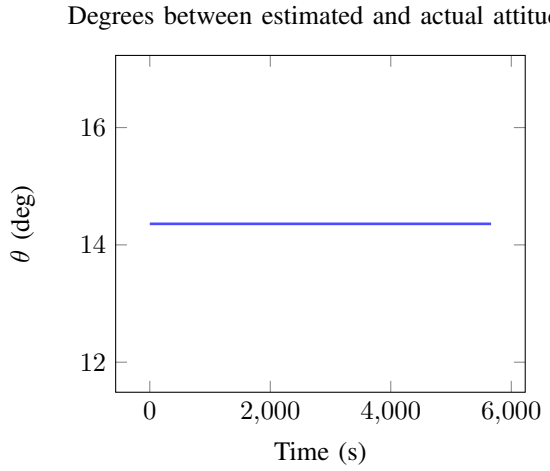


Fig. 11. Pointing Accuracy.

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.

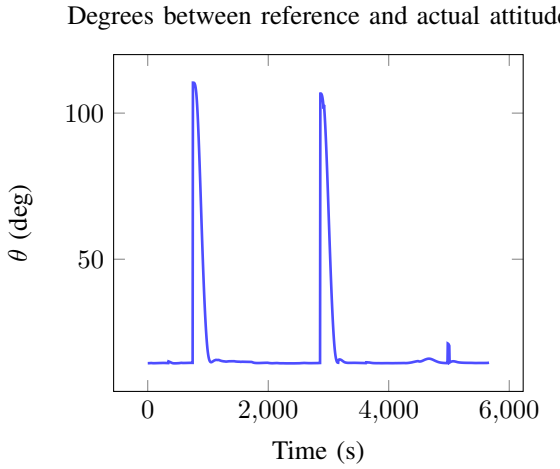


Fig. 12. Pointing Accuracy.

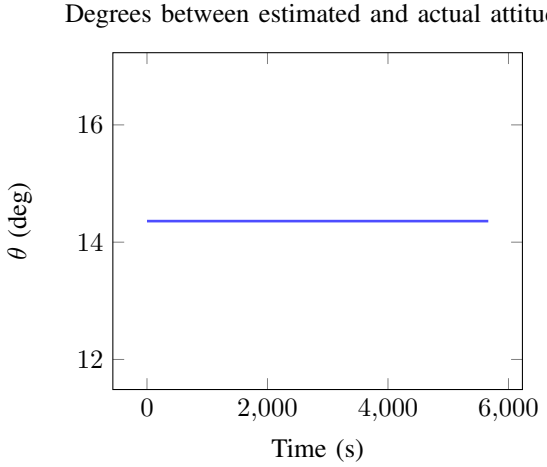


Fig. 13. Pointing Accuracy.

IX. CONCLUSIONS

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

APPENDIX

Table of anomalies

ACKNOWLEDGMENT

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

Orbits	**1**	**1**
Metric	Mean	Std
DecisionTrees	18.292514745477845	28.632580144211847
Perfect	16.472394692952275	25.782235910658237
None	16.472394692952275	25.782235910658237