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

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. An Extended Kalman Filter (*EKF*) is a common estimator used in satellite missions. 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.

A. Related Work

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.

II. SIMULATION ENVIRONMENT

sgp4 simulation environment. Disturbances.

A. Dimensions of Satellite

The dimensions of the satellite are shown in Table I.

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

The Sputnix dimensions for the sun sensor are used.

B. Orbit Parameters

Low earth orbiting (500km)

C. Validation

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

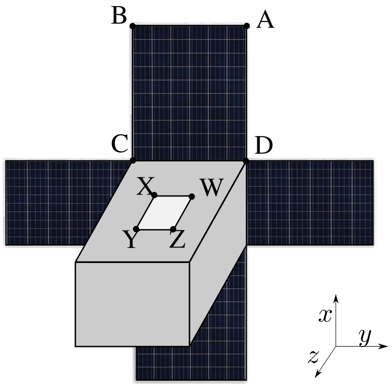

Fig. 1. Cube Sat

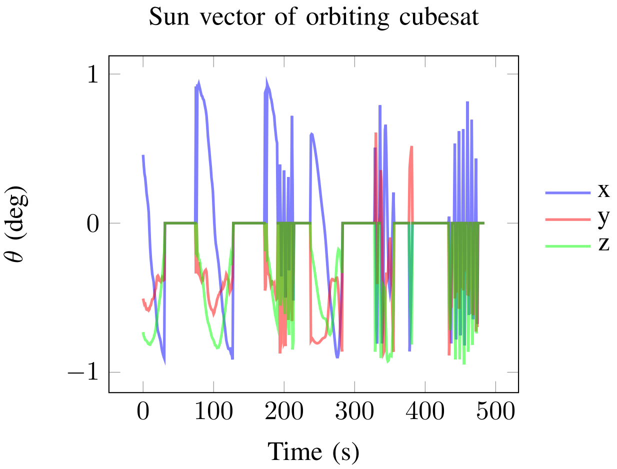

Fig. 2. Sun vector.

D. 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,

*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

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. The integration method used in the simulation is the 4th order Rungka-Kutta method to solve the differential equations. The integration method is shown in 1 where $(\hat{\omega}_B^I)_k^-$ is calculated for the first step of the model update.

Algorithm 1 Rungka-Kutta 4th order Algorithm at time k

```
Satellite Body Inertia $\mathbf{J} = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.45 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$
Timestep $(T_s) = 1s$
Number of iterations $(I) = 10$
Step size $h = \frac{T_s}{h}$
Disturbance torques $N_d = N_{gg} + N_{aero} + N_{rw} - N_{gyro}$
Control torques $N_c = N_m - N_w$
$\mathbf{N} = N_c + N_d$
for $n \in I$ do
    $k_1 = h(\mathbf{J^{-1}N})$
    $k_2 = h(\mathbf{J^{-1}N} + \frac{k1}{2})$
    $k_3 = h(\mathbf{J^{-1}N} + \frac{k_2}{2})$
    $k_4 = h(\mathbf{J^{-1}N} + k_3)$
    $\omega_{n+1} = \omega_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$
end for
$(\hat{\omega}_B^I)_k^- = \omega_{n+1}$
return $(\hat{\omega}_B^I)_k^-$
```

where N_{gg}, gravity gradient disturbance torque, N_{aero}, aerodynamic disturbance torque, N_{rw}, reaction wheel disturbance torque, N_{gyro}, gyroscopic torque, N_m, magnetic control torque and N_w, reaction wheel control torque, are modelled according to Janse van Vuuren [2].

The calculation for the $\hat{q}_k^-$ is done with Eq 3 [2]

$$\hat{q}_k^- = \left[cos(k_q)\mathbf{I}_{4x4} + \frac{1}{\|(\hat{\omega}_B^O)_k^-\|} sin(k_q)\mathbf{\Omega}_k^- \right] \hat{q}_{k-1}^+$$

$$\text{where } k_q = \frac{Ts}{2}\|(\hat{\omega}_B^O)_k^-\|$$

$$\begin{aligned}(\hat{\omega}_B^O)_k^- &= (\hat{\omega}_B^I)_k^- - \hat{\mathbf{A}} \begin{bmatrix} 0 & -(\omega_k) & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} \hat{\omega}_{ox} & \hat{\omega}_{oy} & \hat{\omega}_{oz} \end{bmatrix}^T\end{aligned}$$

$$\|(\hat{\omega}_B^O)_k^-\| = \sqrt{\hat{\omega}_{ox}^2 + \hat{\omega}_{oy}^2 + \hat{\omega}_{oz}^2}$$

$$\text{and } \mathbf{\Omega}_k^- = \begin{bmatrix} 0 & \hat{\omega}_{oz} & -\hat{\omega}_{oy} & \hat{\omega}_{ox} \\ -\hat{\omega}_{oz} & 0 & \hat{\omega}_{ox} & \hat{\omega}_{oy} \\ \hat{\omega}_{oy} & -\hat{\omega}_{ox} & 0 & \hat{\omega}_{oz} \\ -\hat{\omega}_{ox} & -\hat{\omega}_{oy} & -\hat{\omega}_{oz} & 0 \end{bmatrix} \tag{3}$$

The estimated state vector, $\hat{x}_k^-$ can now be expressed as

$$\hat{x}_k^- = \begin{bmatrix} (\hat{\omega}_B^I)_k^- & \hat{q}_k^- \end{bmatrix} \tag{4}$$

Q_k is the covariance matrix representing the discrete system noise and is assumed to be zero-mean and Guassian. Φ_k is the discrete system perturbation model. H_k is the discrete measurement perturbation Jacobian Matrix. $\mathbf{R}_k$ is the measurement noise covariance matrix. The state covariance matrix P_k can be propagated with Eq 5.

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1}^+ \Phi_k^T \tag{5}$$

The $\mathbf{e}_k$ calculated with Eq 6

$$\mathbf{e}_k = v_{meas,k} - \hat{\mathbf{A}}_k^- v_{model,k} \tag{6}$$

where $v_{meas,k}$ is the measured vector in **SBC** and $v_{model,k}$ is the modelled **ORC** vector. The gain matrix $\mathbf{K}_k$ is used to determine the the influence of $\mathbf{e}_k$ on updated state vector, $\hat{x}_k^+$. $\mathbf{K}_k$ can be calculated as

$$\mathbf{K}_k = \mathbf{P}_k^- (\mathbf{H}_k^-)^T \left[\mathbf{H}_k^- \mathbf{P}_k^- (\mathbf{H}_k^-)^T + \mathbf{R}_k \right]^{-1} \tag{7}$$

after which the updated state vector can be calculated with Eq 8

$$\hat{x}_k^+ = \hat{x}_k^- + \mathbf{K}_k \mathbf{e}_k \tag{8}$$

The state covariance matric can then be updated as

$$\mathbf{P}_k^+ = \left[\mathbf{I}_{7\times7} - \mathbf{K}_k \mathbf{H}_k^+ \right] \mathbf{P}_k \left[\mathbf{I}_{7\times7} - \mathbf{K}_k \mathbf{H}_k^+ \right] + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{9}$$

During the measurement update of $\hat{x}_k^-$ with the error of $v_{meas,k}$ and $v_{model,k}$, the $\hat{x}_k^+$ is largely affected by anomalous behaviour in the sensor measurements. The sensitivity of the Kalman filter to this behaviour as well as related work will be discussed in section II-E.

E. *Sensitivity of Kalman Filter*

The effect of sensor anomalies on kalman filter.

III. Sensors

The sensors used for the measurement update of the Kalman filter is a magnetometer, nadir sensor, sun sensor and star tracker and the measurement update is also done in the order as listed. This is due to the noise models of the sensors, as all the sensor noise models are based on zero-mean Guassian random noise. There are two sun sensors, a coarse and fine sun sensor and both of them can experience sun reflection.

IV. Sensor anomalies

To test the various methods proposed, the common sun sensor anomaly of reflection is implemented. The sun sensor's interpreted vector in the SBC frame changes due to reflection from the deployed solar panels that reflects the sun light. The reflection is modelled according to Figure 3 and discussed in section IV-A.

A. *Reflection*

The reflection anomaly is modelled for a any dimensions but with the specific shape of and design of the cubesat as shown in Figure 1.

The 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

$$R = V - 2N^T(V \cdot N) \tag{10}$$

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

$$P = ax + by + cz = d \tag{11}$$

The reflected vector can also be translated to equations of

$$\begin{aligned} x &= \alpha t \\ y &= \beta t \\ z &= \zeta t \end{aligned} \tag{12}$$

Since we can calculate the coefficients for Eq 12 from the reflected vector, we can calculate t, by substituting $x, y\&z$ into Eq 11. Consequently the intersecting points x, y and z can be calculated as

$$P(x,y,z) = (o_1 + \alpha t, o_2 + \beta t, o_3 + \zeta t) \tag{13}$$

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

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 **??** 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 13. 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.

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.

V. 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.

$$x_{k+1} = \mathbf{A}x_k \tag{14}$$

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

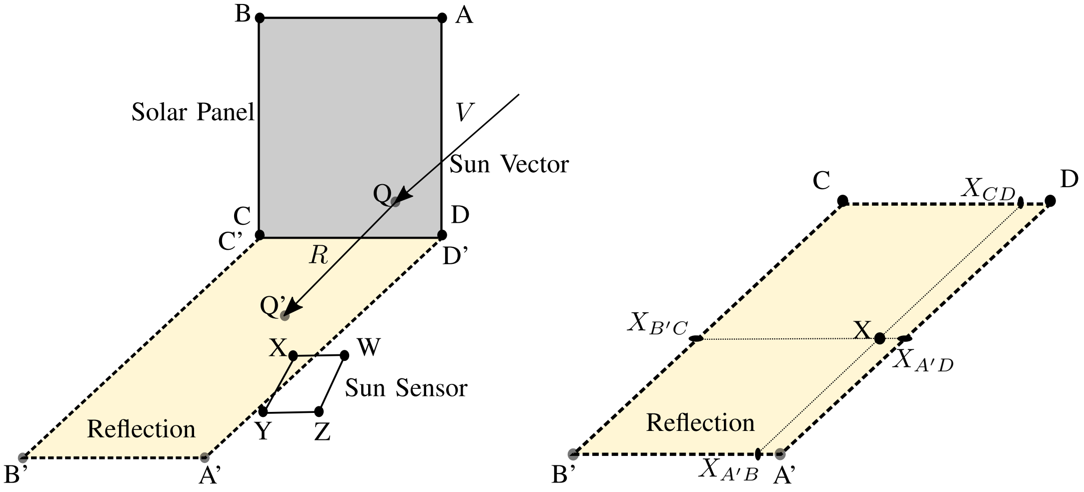

Fig. 3. Reflection

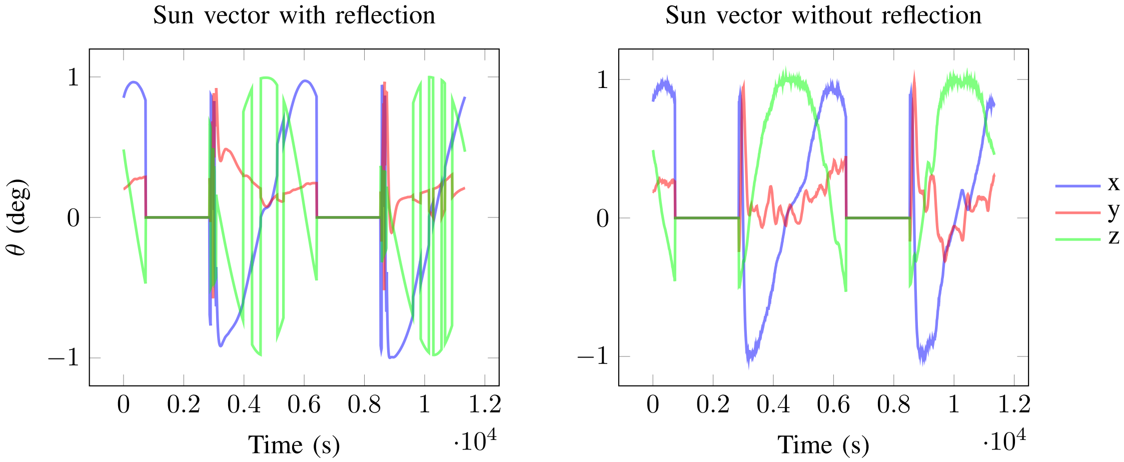

Fig. 4. Sun vector without reflection.

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

$$\mathbf{A} = \mathbf{X}\mathbf{X}^{\dagger} \tag{15}$$

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.

$$x_{k+1} = \boldsymbol{A}x_k + \boldsymbol{B}y_k \tag{16}$$

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 16 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. [5] 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}

$$\hat{x}_{k+1} = A\hat{x}_k + By_k + K(x_k - \hat{x}_k) \tag{17}$$

After the calculation of $\hat{x}_{k+1}$ Silva et al. [5] proposes a moving average of the innovation covariance

$$\boldsymbol{V}_k = \frac{1}{N}\sum_{i=k-N}^{k}(x_i - \hat{x}_i)(x_i - \hat{x}_i)^T \tag{18}$$

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 19.

$$GI = 1 - \sum_{i=1}^{n} (P_i)^2 \tag{19}$$

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

It is also possible to discuss random forests if it ends up being used in the results, because it is already implemented.

C. Recovery

Three different methods of recovery are compared. The methods are name Ignore, Backtrack and Replacement.

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 method can also have variations, such as the change of the measurement noise covariance matrix R_k. This however will dramatically change K_k and will destabilize the EKF rather than stabilizing it. Both methods results however will be discussed in section VII.

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.

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 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.

VI. Testing methodology

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.

VII. 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.

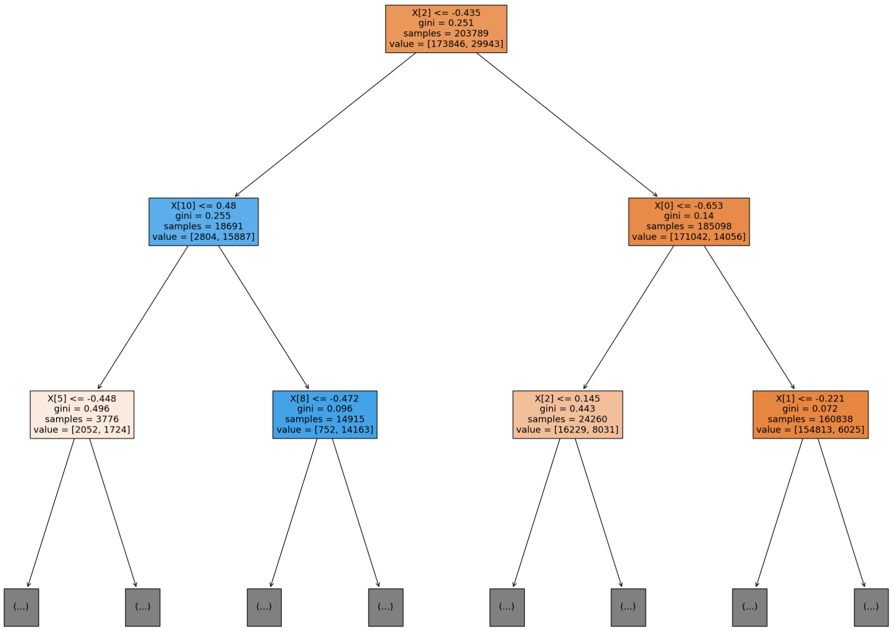

Fig. 5. fig:DecisionTree

Orbits	1	1
Metric	Mean	Std
DecisionTrees	18.2925147454 77845	28.632580144211847
Perfect	16.472394692952275	25.782235910658237
None	16.472394692952275	25.782235910658237

A. *Perfect Designed Satellite Without Reflection*

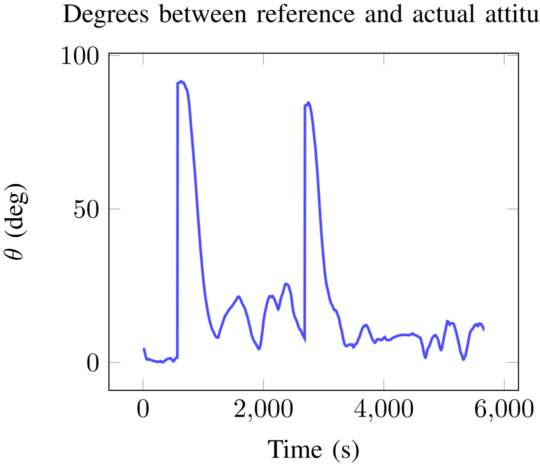

Fig. 6. Pointing Accuracy.

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.

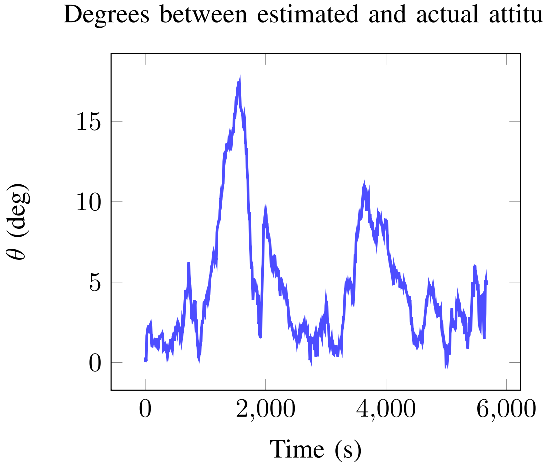

Fig. 7. Estimation Accuracy.

VIII. 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.

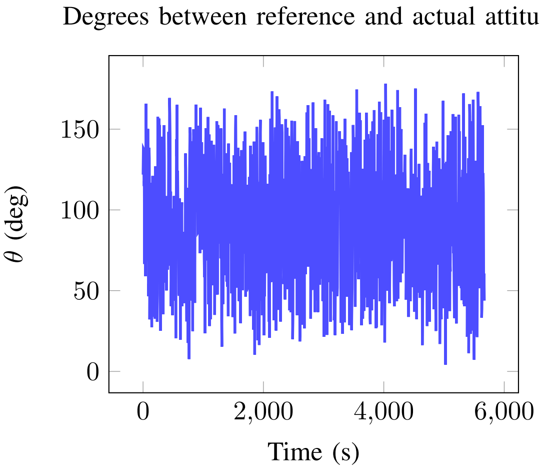

Fig. 8. Pointing Accuracy.

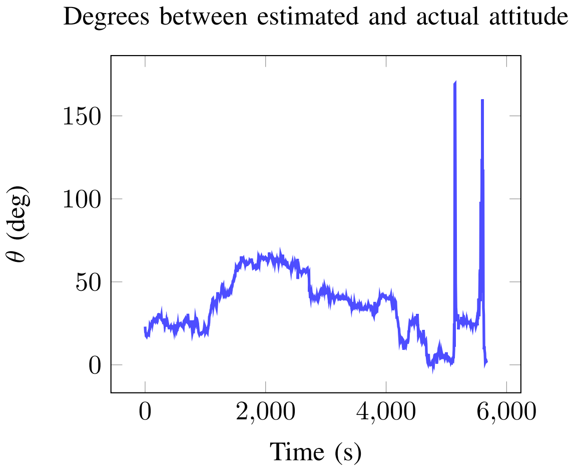

Fig. 9. Estimation Accuracy.

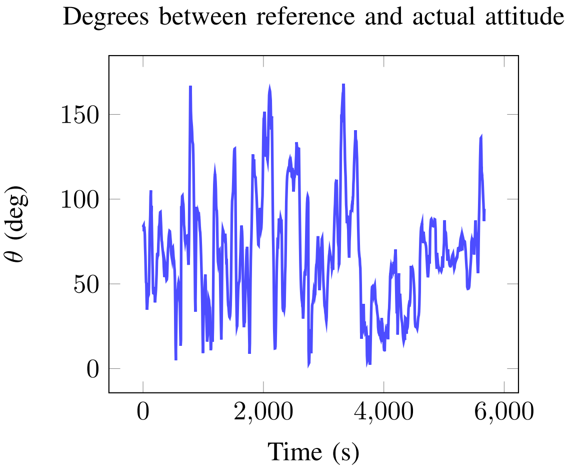

Fig. 10. Pointing Accuracy.

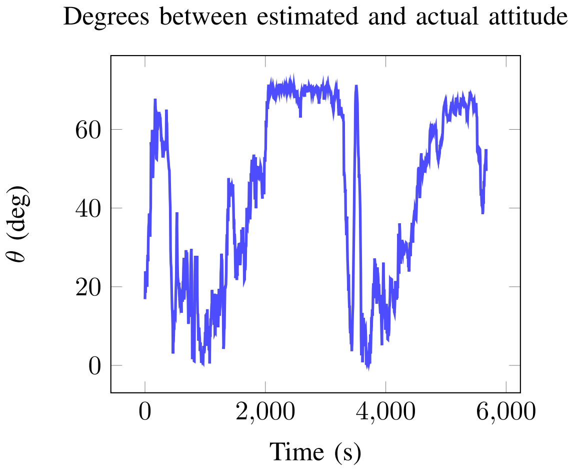

Fig. 11. Pointing Accuracy.

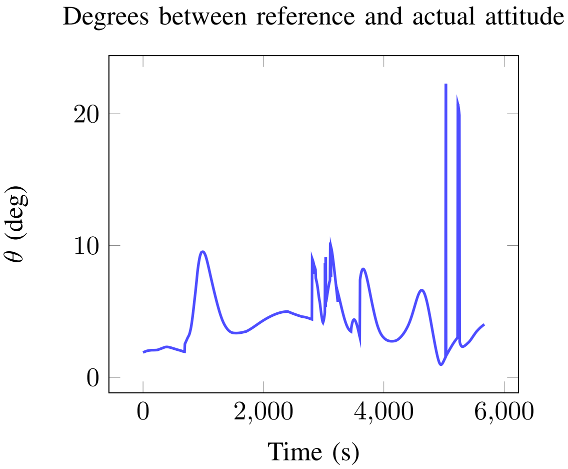

Fig. 12. Pointing Accuracy.

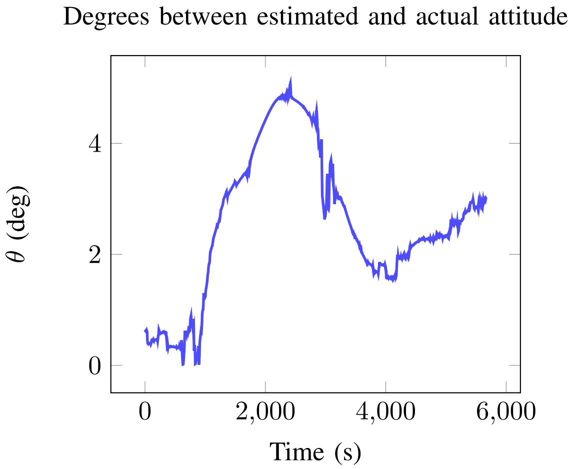

Fig. 13. Pointing Accuracy.