Fault Isolation and Identification of a Four-Single-Gimbal Control Moment Gyro On-board a 3-axis Stabilized Satellite

Afshin Rahimi[1]

[1]*Department of Mechanical, Automotive and Materials Engineering, University of Windsor, Windsor, Canada, N9B 3P4*

arahimi@uwindsor.ca

ABSTRACT

Control moment gyros are known for their applications in attitude stabilization. These actuators are susceptible to malfunction, which results in faults and failures. Therefore, diagnosing the faults can improve the reliability of completing a mission while reducing maintenance costs. Thus, a model-based fault diagnosis method is proposed here. The intended algorithm is an enhanced version of previous work by the author. The enhancement employs a condensed approach to alleviate the delay caused by the filter's confidence in its estimations. A case-study on a closed-loop controlled satellite is provided along with an extensive Monte Carlo simulation to evaluate the proposed method's performance. The results show that the enhanced method can achieve superior performance while requiring less computational resources by eliminating extra grid search loops.

1. INTRODUCTION

A control moment gyros (CMG) consists of a flywheel gimballed about its axes. The torques generated by the rotation of the flywheels' rotation axes provide a means to control the satellite's orientation in space.

Many researchers have examined fault isolation and identification (FII) in the past few years. Parameter estimation methods (Jiang et al., 2008; Li et al., 2011; Rahimi et al., 2015, 2020; Rahimi, Dev Kumar, et al., 2019; Rahimi, Kumar, et al., 2019; Ye et al., 2015), can achieve fault detection and isolation at the same time as these methods estimate the system parameters directly which offers time, source, and size of the fault altogether.

To the author's knowledge, the FII framework for CMGs as satellite actuators has not been explored extensively in the published literature. Rahimi et al. (2020) have proposed a hierarchical model-based fault detection, isolation, and identification (FDII) approach that employs a binary search to update the parameter estimation filter's posterior covariance matrix elements. The proposed method in (Rahimi et al., 2020) provides an enhancement on a previously developed detection and isolation method in (Rahimi et al., 2017) by reducing the loops in the search for posterior covariance matrix elements from all elements to only the diagonal elements and further reducing it by moving from a multiple-value grid search to a binary grid search. The enhanced method in (Rahimi et al., 2020) provides speed and accuracy improvements in the estimator compared to (Rahimi et al., 2017). However, the author believes that further improvement can be made in the estimation process.

Therefore, this paper proposes an enhancement on the method provided in (Rahimi et al., 2020) by further simplifying the grid search process and using fewer loops to reduce the estimator's computational requirements without sacrificing the performance. Furthermore, the performance of the proposed enhancement is evaluated on a satellite stabilized by a four-single-gimbal-CMG (4SGCMG) cluster in comparison to the previous methods (Rahimi et al., 2020) and (Rahimi et al., 2017) to provide evidence on the superior performance of the novel approach.

The remaining of this manuscript contains the problem definition in Section 2. The proposed enhancement in Section 3. The case-study in Section 4. The simulation setup, along with the results and discussions in Section 5, and finally, the conclusions in Section 6.

2. PROBLEM DEFINITION

Considering the nonlinear system:

$$\Omega: \begin{cases} \xi_{k+1} = f\left(\xi_k, u_k, \theta_k, w_k^\xi\right) \\ \theta_{k+1} = \theta_k + w_k^\theta \\ y_k = g(\xi_k, \theta_k) + v_k \end{cases} \quad (1)$$

where $\xi_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $\theta_k \in \mathbb{R}^l$, $y_k \in \mathbb{R}^m$ denote state, control input, system parameters, and measurements, respectively. $w_k^{\xi} \in \mathbb{R}^n$, $w_k^{\theta} \in \mathbb{R}^l$, $v_k \in \mathbb{R}^m$ denote the additive process noise for states, additive process noise for parameters, and additive measurement noise, respectively. $f(\cdot)$ and $g(\cdot)$ denote a nonlinear process model and a nonlinear measurement model, respectively. The objective is to develop an estimation method to isolate the location and identify the severity of the faults in the system with less computation time and more accuracy than (Rahimi et al., 2020) and (Rahimi et al., 2017).

This requires that the system component faults can affect the physical system parameters (Sobhani-Tehrani et al., 2014), under which the Eq. (1) will have:

$$\theta_k = \theta_0 + \alpha_k \qquad (2)$$

where $\theta_0 \in \mathbb{R}^l$ denotes the nominal parameter values and $\alpha_k \in \mathbb{R}^L$ denotes the fault parameters containing L fault elements. Employing Eq. (2), one can treat the nonlinear FII as a parameter estimation problem.

3. METHODOLOGY

The proposed FII in this study focuses on the second stage of a hierarchical approach by Rahimi et al. (2020) to propose a novel adaptation mechanism for the FII. Hence, fault detection is briefly explained, while the adaptation mechanism is further detailed.

3.1. Fault Detection

To detect the faults in the system, a residual r_k is generated using (Rahimi et al., 2017)

$$r_k = \hat{y}_k^s - \hat{y}_k^m \qquad (3)$$

where $\hat{y}_k^s$ and $\hat{y}_k^m$ are the estimated outputs for the system and model, respectively, with scalar β defined as

$$\beta = r_k^T r_k \qquad (4)$$

The threshold β_0 (Rahimi et al., 2017) triggers a fault when $\beta > \beta_0$. The next step in the process is updating the system model (Ω_s) to match the current state of the system referred to as FII, detailed in the next section.

3.2. Filter Adaptation Algorithm

The posterior estimates covariance matrix $P_{\xi\xi}$ elements represent the confidence in the estimated values (Bisht & Singh, 2014). The issue arises as the filter converges and subsequently becomes slow to adapt to sudden changes. To address this, $P_{\xi\xi}$ elements must be adapted to avoid latency following a fault detection. To achieve this, Rahimi et al. (2017) proposed the covariance adaptation algorithm (CAUKF) using the unscented Kalman filter (UKF) algorithm (2017). Rahimi et al. improved the method using a binary grid covariance adaptation algorithm (GAUKF)

(2020), where the idea is to only consider the binary choice of "increase" or "no change" for the diagonal elements of the $P_{\xi\xi}$. In this study an improved version of the adaptation is proposed as simplified binary grid covariance adaptation algorithm (SGAUKF), as outlined in Table 1. For the proposed approach in this study, in step 2 of Table 1, the choices (combinations) are only formed for the even or odd PP_k diagonal elements. This would reduce the order of options by half resulting in a loop with $2^{D/2}$ instead of 2^D compared to the CAUKF. In Table 1, PP_k is used as a provisional variable for finding diagonal elements of the P_k. PP_i is the binary combination to add to the previous step's posterior covariance P_{k-1}, PP_{k_i} is the candid superimposed posteriori covariance, and r_{P_i} is the residual for estimates of the next time step using PP_{K_i} as posterior covariance. The rationale for only adjusting the odd or even elements in the diagonal for the posterior matrix is the fact that since the parameters that are being estimated are interrelated through the system dynamics, once the filter is adjusted to less confidence in half of its elements, it will try to compensate for the other elements as well. Hence, not having to check for all binary options reduces the computation time without sacrificing the performance. In some cases, it provides better performance, as will be seen in the results section.

3.3. Fault Isolation and Identification

The FII can be accomplished using Eq. (2). The number of possibilities corresponds to the sum of all k-combinations of the faulty units as

$$L = \sum_{p=1}^{n} \binom{n}{p} = \sum_{p=1}^{n} \frac{n!}{p!\,(n-p)!} \qquad (5)$$

where, at time step k, p denotes the number of faulty units, and n is the total number of available units.

A dual state/parameter estimation (Rahimi et al., 2020) is employed to estimate the system states and fault parameters in the system and achieves the FII in one step. At this stage, $\hat{\theta}_k^s$ and $\hat{\theta}_k^m$ are introduced as parameter estimates for the system and model, respectively, using Ω_s. A normalized residual is calculated using Eq. (4) as $\beta = r^T \mathfrak{M} r$, where $\mathfrak{M}$ is the weight matrix for y_k normalization.

1:	$r_{old} = \infty$
2:	Generate binary combinations (total of $2^{D/2}$)
3:	For $PP_i = 1\ to$ combinations count ($2^{D/2}$)
	$PP_{k_i} = P_{k-1} + PP_i$
3b:	Calculate ξ_k using PP_k
3c:	Calculate $r_{P_i} = (z_k - \xi_k)^T (z_k - \xi_k)$
4:	Go to 3
5:	$P_k = PP_{k_i}(index\ of(minimum(r_{P_i}))$

Table 1. SGAUKF algorithm

This process is repeated until the residual is smaller than $\varepsilon_r = 1 \times 10^{-3}$ after which the Ω_s is updated with the new $\hat{\theta}_k^s$. Once the system parameters are estimated, one can evaluate α_k to achieve fault isolation using the fact that during a fault $\alpha_k \neq 0$. Furthermore, employing a 1-to-k encoding, the corresponding fault scenario can be obtained.

4. CASE STUDY: FAULT DIAGNOSIS FOR CMGS

The proposed FII is evaluated using an ACS of a three-axis stabilized satellite with a 4SGCMG cluster. The simulation setup is shown in Figure 2, with details as follows:

4.1. Satellite Dynamics and Kinematics

For a satellite employing CMGs, the dynamics and kinematics equations can be written as (Wie et al., 2001)

$$J\dot{\omega}_{BI}^B = -\omega_{BI}^B \times (J_s\omega_{BI}^B + h_{CMG}) - \dot{h}_{CMG} + \tau_e \qquad (6)$$

$$\begin{bmatrix} \dot{q}_v \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 I + q_v^\times \\ -q_v^T \end{bmatrix}\omega_{BL}^B \qquad (7)$$

where ω_{BI}^B is the satellite's angular velocity. $\tau_e \in \mathbb{R}^{3\times1}$ denotes the external torque. $J = J_s - AJ_wA^T$ with $J_s \in \mathbb{R}^{3\times3}$ denoting satellite's inertia with actuators included. $J_w \in \mathbb{R}^{4\times4} = diag([J_{w1}, J_{w2}, J_{w3}, J_{w4}])$ denotes CMGs moments of inertia. h_{CMG} and $\dot{h}_{CMG}$ can be obtained from Eq. (9) and Eq. (10), respectively. $q_4 \in \mathbb{R}$, $q_v \in \mathbb{R}^{3\times1} = [q_1, q_2, q_3]^T$ denote quaternions. $I \in \mathbb{R}^{3\times3}$ is the identity matrix and superscript $(\cdot)^\times$ denotes skew-symmetric form of matrix $(\cdot)$.

4.2. Actuators

The CMG angular momentum depends on its gimbal angles $\delta = (\delta_1, \dots, \delta_n)$ and flywheels angular speed $\Omega = (\Omega_1, \dots, \Omega_n)$ with

$$\dot{h}_{CMG} = A_{CMG}\dot{\delta} \qquad (8)$$

where $A_{CMG} = A_{CMG}(\delta) \in \mathbb{R}^{3\times n}$ is the Jacobin matrix.

Calculating the inverse of $A_{CMG}\dot{\delta} = \dot{h}_{CMG}$, one can obtain the CMG steering logic. For a 4SGCMG (Figure 1) with an out-of-plane angle of $\beta = 54.73°$,
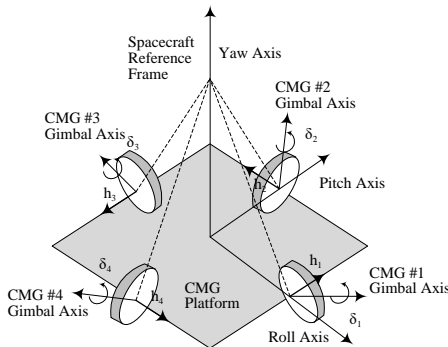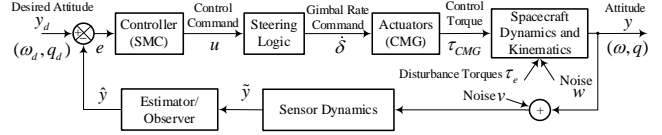

Figure 1. 4SGCMG schematics

Figure 2. FII simulation setup

The angular momentum becomes

$$h_{CMG} = \begin{bmatrix} -c\beta s\delta_1 & -c\delta_2 & c\beta s\delta_3 & c\delta_4 \\ c\delta_1 & -c\beta s\delta_2 & -c\delta_3 & c\beta s\delta_4 \\ s\beta s\delta_1 & s\beta s\delta_2 & s\beta s\delta_3 & s\beta s\delta_4 \end{bmatrix} \times$$
$$\begin{bmatrix} h_{0_1}(\Omega_1) & h_{0_2}(\Omega_2) & h_{0_3}(\Omega_3) & h_{0_4}(\Omega_4) \end{bmatrix}^T \qquad (9)$$

where h_i, δ_i, Ω_i, and h_{0_i}, denote the angular momentum, the gimbal angle, flywheel angular speed, and the momentum magnitude for the ith CMG, respectively. Taking the time derivative, one would arrive at

$$\dot{h}_{CMG} = \begin{bmatrix} h_{0_1}(\Omega_1) & h_{0_2}(\Omega_2) & h_{0_3}(\Omega_3) & h_{0_4}(\Omega_4) \end{bmatrix}A_{CMG}\dot{\delta} \qquad (10)$$

where δ denotes the gimbal angle vector with

$$A_{CMG} = \begin{bmatrix} -c\beta c\delta_1 & -s\delta_2 & c\beta c\delta_3 & -s\delta_4 \\ -s\delta_1 & -c\beta c\delta_2 & -s\delta_3 & c\beta c\delta_4 \\ s\beta c\delta_1 & s\beta c\delta_2 & s\beta c\delta_3 & s\beta c\delta_4 \end{bmatrix} \qquad (11)$$

Consequently, the torque command $\dot{h}$ yields

$$\dot{h}_{CMG} = u = -\tau_c - \omega_{BI}^B \times h_{CMG} \qquad (12)$$

Given control torque τ_c and having $h_0 = h_{0_i}; i = 1 \dots 4$, $\dot{\delta}$ can be obtained as (Wie et al., 2001)

$$\dot{\delta} = (\frac{1}{h_0})A_{CMG}^+\dot{h}_{CMG} \qquad (13)$$

where $A_{CMG}^+ = A_{CMG}^T(A_{CMG}A_{CMG}^T)^{-1}$.

4.3. Controller

A sliding mode controller is used to attain the desired attitude of $q_d \in \mathbb{R}^{4\times1}$ and $\omega_d \in \mathbb{R}^{3\times1}$ (Kumar et al., 2018). The quaternion and angular velocity errors are obtained as

$$q_e = q_{d4}q_v - q_4q_{dv} + q_v^\times q_{dv} \qquad (14)$$
$$q_{e4} = q_{d4}q_4 + q_{dv}^Tq_v$$

$$\omega_e = \omega_{BL}^B - C_e\omega_d \qquad (15)$$

where $C_e = C(q_e, q_{4e})$ is given by

$$C_e = (q_{4e}^2 - q_e^Tq_e)I + 2q_eq_e^T - 2q_{4e}q_e^\times \qquad (16)$$

The sliding manifold can be formulated using these terms as

$$\sigma = \omega_e + \lambda_{ctr}sgn(q_{4e})q_e \qquad (17)$$

where $\lambda_{ctr} > 0$ is the sliding gain, and $sgn(\cdot)$ is the sign function. The required control command is defined as

$$u_r = -p_0\sigma \qquad (18)$$

where $p_0 = 0.1$ (Wie et al., 2001), $\lambda_{ctr} = 1$ (Kumar et al., 2018).

4.4. Fault Formulation

Faults are injected into A_{CMG} in Eq. (11) to represent intermittent time-varying faults using

$$A_{CMG} = A_{CMG} F_p \qquad (19)$$

where

$$F_p = diag\left([f_{p_1}, f_{p_2}, f_{p_3}, f_{p_4}]\right) \qquad (20)$$

where $f_{p_i} \in [0,1]$ denotes the i th CMG's effectiveness, *bearing in mind* $f_p = 1$ for nominal performance and $f_p = 0$ for failure (Kim et al., 2015).

4.5. Fault Scenario

The total number of possibilities for faulty scenarios using Eq. (5) adds up to 16. To easily refer to each scenario, Table 2 assigns each possibility a number where FW abbreviates faulty wheels. For example, in scenario 0, no wheel is faulty, and in scenario 15, all wheels are faulty.

Having the building components of the FII detailed so far, the proposed fault diagnosis for this case study can be formulated as follows:

4.6. Fault Detection

The satellite dynamics and kinematics are defined in Eq. (6) and Eq. (7) with A_{CMG} in Eq. (11). The controller that stabilizes the attitude employs Eq. (18) to obtain u_r and uses that to acquire the $\dot{\delta}$ from Eq. (13). At this stage, the system states, outputs, and controls are defined as

$$\xi = [q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4]$$
$$y = [q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4] \qquad (21)$$
$$u = [\tau_x, \tau_y, \tau_z]$$

where u_i are used to calculate $\dot{h}_{cmg}$ in Eq. (12) and plugged into Eq. (13) to obtain $\dot{\delta}_i$. The simulation is initiated with a system model Ω_s where $\alpha_i = 0$. The estimated output vectors $\hat{y}^s$ and $\hat{y}^m$ employ Ω_s for the UKF in (Rahimi et al., 2020) to obtain the states that leads to residual r_k generation in Eq. (3). The residuals are used to form scalar β in Eq. (4) (Rahimi et al., 2017) for fault detection followed by FII, as discussed in the next section.

4.7. Fault Isolation and Identification

Here, f_{p_i}s are estimated for updating Ω_s where the system states and parameters, control inputs, and measurements are defined respectively as

$$\xi = [q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4]$$
$$\theta = [f_{p_1}, f_{p_2}, f_{p_3}, f_{p_4}]$$
$$y = [q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \delta_1, \delta_2, \delta_3, \delta_4] \qquad (22)$$
$$u = [\tau_x, \tau_y, \tau_z]$$

No.	FW	No.	FW	No.	FW	No.	FW
0	None	4	4	8	2,3	12	1,2,4
1	1	5	1,2	9	2,4	13	1,3,4
2	2	6	1,3	10	3,4	14	2,3,4
3	3	7	1,4	11	1,2,3	15	1,2,3,4

Table 2. Fault scenarios for 4SGCMG

The dual UKF process in (Rahimi et al., 2020) is employed to estimate the states ξ and parameters θ. Having θ estimates, one can obtain α using

$$\hat{\alpha} = \hat{\theta} - \theta_0 \qquad (23)$$

where $\hat{\alpha}$ and $\hat{\theta}$ denote the fault parameter and system parameter estimates, respectively. Once α_is are estimated, the updated Ω_s replaces the reference model, and monitoring continues.

5. RESULTS AND DISCUSSIONS

For performance evaluations, numerical simulations were conducted in MATLAB with an integration step size of $T_s = 0.01\,sec$ for a total simulation time of 200 sec. The simulation was run on an Intel® computer with i7-4790 3.60 GHz CPU and 16.00 GB RAM. A list of other simulation parameters is provided in Table 3 where the Euler angles to quaternion conversions follow the 3-2-1 sequence. The initial and desired conditions for roll, pitch, and yaw are $(-90°, 10°, 10°)$ and $(0°, 0°, 0°)$, respectively. $\sigma_Q = 10^{-5}$ and $\sigma_R = 10^{-4}$ are used for the UKF. Additive noise is considered for the measurements in Eq. (21) and (22) when discussing the signal to noise ratio (SNR) in the following sections. The fault scenario considered is an out-of-phase abrupt, as detailed in Table 4. The system response and fault detection results are not provided as this study focuses on isolation. Hence, only the results for FII in Table 4 scenario with Monte Carlo (MC) simulations are provided. In the following sections, simulation results were obtained for three SNRs: 45dB, 55dB, and 75dB; however, only the 45dB (worst-case scenario) results are reported unless explicitly mentioned otherwise.

Parameter	Value
UKF $[\alpha, \kappa, \lambda]$	$[1,2,2]$
Process noise	$Q_0 = \sigma_Q^2 I_{n \times n}$
Measurement noise	$R_0 = \sigma_R^2 I_{m \times m}$
J_s	$diag[0.015, 0.017, 0.020][kg.\,m^2]$
J_w	$10^{-5} \times diag[1,1,1,1][kg.\,m^2]$
Initial $[q_1 q_2 q_3]$	$[0.696, -0.707, 0.0]$
Initial $[\omega_1 \omega_2 \omega_3]$	$[0,0,0]$ rad/s
Desired $[q_1 q_2 q_3]$	$[0,0,0]$
Desired $[\omega_1 \omega_2 \omega_3]$	$[0,0,0]$ rad/s

Table 3. Simulation parameters

Time (sec)	f_{p_1}	f_{p_2}	f_{p_3}	f_{p_4}
$t < 20$	1	1	1	1
$20 \leq t < 40$	0.5	1	1	1
$40 \leq t < 70$	1	0.1	1	1
$70 \leq t < 100$	1	1	0.1	1
$100 \leq t < 150$	1	1	1	0.1
$t \geq 150$	1	1	1	1

Table 4. Case-study out-of-phase abrupt fault

5.1.1. Monte Carlo Analysis

To conduct Monte Carlo simulations, the confusion matrix (CM) in Table 6 is filled with the simulation results of 100 cases for the fault scenario described in Table 4. Each case consists of random values for the satellite attitude (q, ω) initial conditions in addition to, the start time (TF_s), duration (TF_d), and severity of the fault (α_i) with ranges in Table 5. It should be noted that the duration of the fault is dependent on the fault inception as not to exceed the total simulation time.

5.1.2. Fault Isolation and Identification

Table 6 lists the confusion matrix for the MC simulation where "A→" denotes for actual scenario and "I ↓" the identified scenario from all possible scenarios in Table 2.

	q_i	ω_i (rad/s)	TF_s (s)	TF_d (s)	α_i
Min	-1	-0.5	0	0	0
Max	1	0.5	199	$199-TF_s$	1

Table 5. Parameters range for MC simulations

A→ I↓	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	93	0	0	1	0	0	1	0	1	0	0	1	0	0	1
2	0	93	0	1	1	1	1	1	1	1	0	1	1	1	1
3	0	0	91	1	0	0	1	0	1	0	1	1	1	1	1
4	1	1	1	88	1	1	1	0	0	1	1	1	0	1	1
5	1	0	1	1	90	0	1	1	0	0	0	1	0	1	0
6	1	1	1	1	1	92	0	1	0	0	1	1	0	1	1
7	0	0	1	1	1	1	89	1	1	1	1	1	1	1	1
8	0	1	1	1	1	1	1	93	1	0	1	1	1	0	1
9	1	1	1	1	1	1	0	1	91	1	1	1	0	1	1
10	1	1	0	1	1	1	1	0	1	94	0	1	1	1	0
11	1	1	1	1	1	1	1	1	0	1	1	93	0	0	1	0
12	0	0	0	0	1	1	1	0	1	1	1	87	1	1	1
13	1	0	1	1	0	1	1	1	1	0	0	1	93	1	1
14	0	0	0	1	0	0	0	1	0	0	0	1	0	88	1
15	0	1	1	0	1	0	0	0	0	0	0	1	1	1	89

Table 6. Confusion matrix results using SGAUKF

The average accuracy for the confusion matrix is defined as:

$$accuracy \% = \frac{trace(CM)}{N_{CM}} \quad (24)$$

where the $trace(\cdot)$ outputs summation of the matrix diagonal elements and N_{CM} is the total CM diagonal elements. The results shown in Table 7 compare the accuracy of different methods by Rahimi et al. as discussed earlier, using the definition provided in Eq. (24).

Table 7 shows the inferior CAUKF accuracy in a closed-loop controlled system and the superior performance of the SGAUKF compared to both the CAUKF and GAUKF. The inferior accuracy of the CAUKF in such a system setting is owed to its inability to adapt as quickly to unknown system changes as compared to the other two methods GAUKF and SGAUKF.

Table 8 provides the MC simulations' average execution time for each method. The results show that the CAUKF has the highest execution time since it must loop through extensive grids. GAUKF, however, demands less time than CAUKF since it is only doing a binary grid search; nevertheless, SGAUKF has further decreased the computation time by cutting the loops required for the posterior covariance from 2^D to $2^{D/2}$ while also improving the accuracy. Therefore, SGAUKF improves accuracy and reduces computation time compared to others. The reduction in execution time is not extreme due to the processing power allocation required by other parts of the algorithm that still take time and are more dominant.

Figure 3 illustrates the fault identification results where the SGAUKF estimates follow the true values for the system parameters most closely, among other methods that visually prove the superiority of the SGAUKF. To further examine the numerical performance, the root mean squared error (RMSE) for these estimates are calculated using

$$RMSE = \sqrt{\frac{\sum_{t=t_{start}}^{t_{end}} \left(\hat{f}_{p_{i,t}} - f_{p_{i,t}} \right)^2}{t_{end} - t_{start}}} \quad (25)$$

Noise level (SNR)	CAUKF %	GAUKF %	SGAUKF %
75dB	77.13	97.13	98.25
55dB	72.33	92.33	93.41
45dB	79.60	89.60	90.93
Average	75.27	93.02	94.20

Table 7. Fault isolation accuracy comparison

Method	CAUKF	GAUKF	SGAUKF
Execution Time (sec)	9.69	8.74	6.36

Table 8. Execution time comparison

Method	Noise level (SNR)	Pre-fault ($\times 10^{-2}$)	Faulty ($\times 10^{-2}$)	Post-fault ($\times 10^{-2}$)
CAUKF	75dB	3.12	4.45	3.24
	55dB	3.54	4.87	3.76
	45dB	3.87	5.45	3.41
GAUKF	75dB	1.74	2.15	1.83
	55dB	1.98	2.44	2.06
	45dB	2.10	3.14	2.16
SGAUKF	75dB	1.44	2.03	1.71
	55dB	1.36	2.27	1.95
	45dB	1.98	2.89	2.76

Table 9. f_{p_i} RSME comparison

where $\hat{f}_{p_i}$ and f_{p_i} are the estimated and true parameter values, respectively. n is the number of parameters, and t_{start} and t_{end} denote the start and end time for the RMSE calculation in the simulation, respectively.

The t_{start} and t_{end} are introduced to form the period before a fault occurs (pre-fault), during fault (faulty), and after the fault recovery (post-fault) and calculate RSMEs for these periods and list them in Table 9. Table 9 values are calculated for each parameter f_{p_i} and averaged over the range of pre-fault, faulty, and post-fault periods. It can be seen from the results in Table 9 that the SGAUKF has a superior performance in comparison to other methods. It is evident that during the fault, RSMEs are larger, and the difference is directly proportional to the noise level.

Table 10 summarizes the computation time for the FII, where most allocation is in the isolation module given the iterative composition. The computation time increases with the noise level as more iterations are required to complete the adaptation in the posterior covariance matrix.

6. CONCLUSIONS

A modified adaption method (SGAUKF) was proposed in this paper to adjust the elements of the UKF posterior matrix so that abrupt changes in system parameters can be isolated promptly.

Noise level (SNR)	Limit	Isolation and Identification (sec)		
		CAUKF	GAUKF	SGAUKF
75dB	Min	0.02	0.02	0.01
	Max	0.36	0.02	0.02
55dB	Min	0.03	0.03	0.02
	Max	0.49	0.03	0.03
45dB	Min	0.05	0.05	0.03
	Max	0.53	0.05	0.05

Table 10. FII computation time comparison

The proposed method was tested on a 4SGCMG to control the satellite attitude. Simulations included out-of-phase abrupt faults with extensive Monte Carlo investigations. Confusion matrices were employed along with RMSE and computation times to compare isolation results. Results showed 94% accuracy in isolation results compared to 93% for GAUKF and 75% for CAUKF. Despite insignificant accuracy improvement, computation time was decreased from 8.74 to 6.36 sec, i.e., ~27%, compared to GAUKF, proving the SGAUKF as a promising option for FII of nonlinear systems with comparable accuracy to GAUKF and less computation time simpler structure requirements.
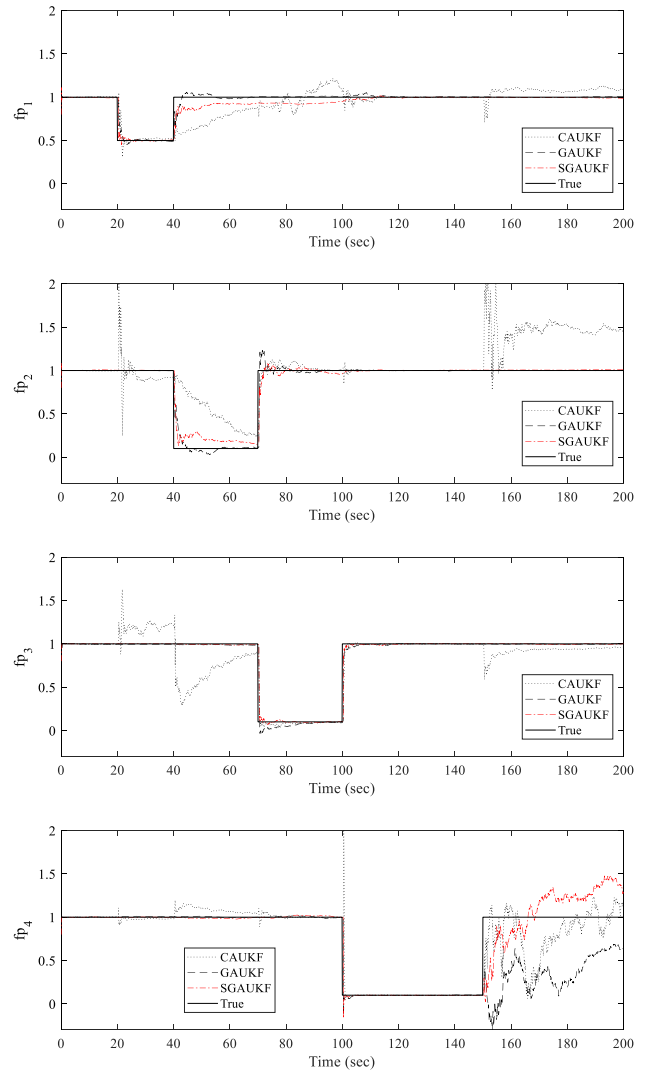
Figure 3. Fault identification performance comparison

REFERENCES

Arkinson, K. E. (1989). *An Introduction to Numerical Analysis* (2nd ed.). John Wiley & Sons.

Bisht, S. S., & Singh, M. P. (2014). An Adaptive Unscented Kalman Filter for Tracking Sudden Stiffness Changes. *Mechanical Systems and Signal Processing*, *49*(1–2), 181–195. https://doi.org/10.1016/j.ymssp.2014.04.009

Jiang, T., Khorasani, K., & Tafazoli, S. (2008). Parameter Estimation-Based Fault Detection, Isolation and Recovery for Nonlinear Satellite Models. *IEEE Transactions on Control Systems Technology*, *16*(4), 799–808. https://doi.org/10.1109/TCST.2007.906317

Kim, D., Turner, J. D., & Junkins, J. L. (2015). Optimal Actuator Failure Control Using a Homotopy Method. *Journal of Guidance, Control, and Dynamics*, *38*(4), 623–630. https://doi.org/10.2514/1.G000012

Kumar, K. D., Godard, Abreu, N., & Sinha, M. (2018). Fault-tolerant attitude control of miniature satellites using reaction wheels. *Acta Astronautica*, *151*(May), 206–216. https://doi.org/10.1016/j.actaastro.2018.05.004

Li, X., Zhang, Q., & Su, H. (2011). An adaptive observer for joint estimation of states and parameters in both state and output equations. *International Journal of Adaptive Control and Signal Processing*, *25*(9), 831–842. https://doi.org/10.1002/acs.1244

Rahimi, A., Dev Kumar, K., & Alighanbari, H. (2019). Fault Isolation of Reaction Wheels for Satellite Attitude Control. *IEEE Transactions on Aerospace and Electronic Systems*, 1–1. https://doi.org/10.1109/TAES.2019.2946665

Rahimi, A., Dev Kumar, K., & Alighanbari, H. (2020). Fault detection and isolation of control moment gyros for satellite attitude control subsystem. *Mechanical Systems and Signal Processing*, *135*, 106419. https://doi.org/10.1016/j.ymssp.2019.106419

Rahimi, A., Kumar, K. D., & Alighanbari, H. (2015). Enhanced Adaptive Unscented Kalman Filter for Reaction Wheels. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(2), 1568–1575. https://doi.org/10.1109/TAES.2014.130766

Rahimi, A., Kumar, K. D., & Alighanbari, H. (2017). Fault estimation of satellite reaction wheels using covariance based adaptive unscented Kalman filter. *Acta Astronautica*, *134*, 159–169. https://doi.org/10.1016/j.actaastro.2017.02.003

Rahimi, A., Kumar, K. D., & Alighanbari, H. (2019). Failure Prognosis for Satellite Reaction Wheels Using Kalman Filter and Particle Filter. *Journal of Guidance, Control, and Dynamics*, 1–4. https://doi.org/10.2514/1.G004616

Sobhani-Tehrani, E., Talebi, H. A., & Khorasani, K. (2014). Hybrid fault diagnosis of nonlinear systems using neural parameter estimators. *Neural Networks*, *50*, 12–32. https://doi.org/10.1016/j.neunet.2013.10.005

Wie, B., Bailey, D., & Heiberg, C. (2001). Singularity robust steering logic for redundant single-gimbal control moment gyros. *Journal of Guidance, Control, and Dynamics*, *24*(5), 865–872. https://doi.org/10.2514/2.4799

Ye, H., Wang, W., & Zhai, S. (2015). Fault Diagnosis Based on Parameter Estimation in Closed-Loop Systems. *IET Control Theory & Applications*, *9*(7), 1146–1153. https://doi.org/10.1049/iet-cta.2014.0717

BIOGRAPHIES

Afshin Rahimi was born in Tehran, Iran in 1988. He received his B.Sc., M.Sc., and Ph.D. degrees in Aerospace Engineering from KNTU in Iran and Ryerson University in Canada in 2010, 2012, and 2017, respectively. He worked at Pratt & Whitney Canada from 2017 to 2018 and started an Assistant Professor position at the University of Windsor in 2018 in the Department of Mechanical, Automotive and Materials Engineering, where he resides now. For the last ten years, he has been involved in various industrial research, technology development, and systems engineering projects/contracts related to the control & diagnostics of satellites, UAVs, and commercial aircraft subsystems. His research work is primarily focused on model-based and data-driven fault detection, diagnostics, and prognosis; machine learning and intelligent systems; linear & nonlinear controller/observer design; and avionics, sensors, and measurement technologies. More recently, he has started working on smart factories and intelligent industrial automation in industry 4.0 in collaboration with industrial partners using computer vision, machine learning and artificial intelligence. He is a member of IEEE and AIAA institutions in addition to the PHM society.