Fault isolation of reaction wheels onboard 3-axis controlled in-orbit satellite using ensemble machine learning techniques

Afshin Rahimi, Atilla Saadat

Department of Mechanical, Automotive and Materials Engineering
University of Windsor, Ontario, Canada
arahimi@uwindsor.ca, saadata@uwindsor.ca

Abstract - The primary objective of this study is to explore novel applications of data-driven machine learning methods for isolation of nonlinear systems with a case-study for an in-orbit closed-loop controlled satellite with reaction wheels as actuators. High-fidelity models of the 3-axis controlled satellite are developed to provide an abundance of data for both healthy and various faulty conditions of the satellite. This data is then used as input for the proposed data-driven fault isolation method. Once a fault is detected, the fault isolation module is activated where it employs a machine learning technique which incorporates ensemble methods involving random forests, decision trees, and nearest neighbors. Results of the classified faulty condition are then cross-validated using k-fold and leave-one-out methods. A comprehensive comparison of the performance of different combinations for the ensemble architecture. Results show promising outcomes for fault isolation of the non-linear systems using ensemble methods.

***Keywords**: Attitude determination and control system (ADCS), fault detection and isolation (FDI), spacecraft, satellite, ensemble machine learning, reaction wheel*

1 Introduction

With the ever-growing number of complex systems, the cost of maintenance is rapidly increasing. In addition, the transition from larger systems to smaller interconnected units dictates less available space on each for hardware redundancy and calls for alternative solutions. Projections indicate as many as 2,600 nano/microsatellites will require launch over the next five years. Furthermore, annual nano/microsatellite launches have grown by over 200% in the last five years [1]. This growth requires advanced monitoring systems that can compensate for the lack of redundancy in hardware due to smaller designs. With the projections above for the next 5 to 10 years, it is evident that advances in analytical redundancy along with diagnosis, prognosis, and health monitoring frameworks can help adopt the emerging technology and safeguard its progression. With the recent increase in spacecraft development projects worldwide, and accessibility for launch vehicle procurement, the need for reliable satellite subsystem Assembly, Integration and Test (AIT) methods become crucial for mission success. One such satellite subsystem, the Attitude Determination & Control System (ADCS), is critical to satellite operation across various mission types. ADCS stabilizes and aligns the satellite to the desired vector during the mission cycle using sensors to determine attitude and actuators to control orientation, despite external torque disturbances [2]. Conventional satellite actuators include reaction wheels (RW), magnetorquers (MT), and control moment gyroscopes (CMG).

Due to the importance of ADCS on satellites, any sub-component malfunction in orbit would jeopardize the success of the mission. As a result, there is a great incentive to develop and improve fault detection and isolation

(FDI) methods for ADCS. For general complex systems, there has been increasing attention utilizing FDI methods in industries such as nuclear engineering [3], chemical engineering [4], automotive engineering [5] and wind turbine engineering [6]. Unlike the various earth-based engineering applications, satellites must operate in an environment where human intervention is minimized. Hence, it is beneficial that satellites are retrofitted for autonomous and intelligent data processing, failure mode classification, and recovery methods.

Many techniques have been proposed to solve the issue of proper FDI implementation in complex systems and ADCS on satellites. FDI methods are commonly categorized into two concepts, model-based [7],[8], and data-driven. As spacecraft design increase in complexity, highly accurate models of components may not be feasible, thus a model-based FDI method would not outperform alternative approaches. There are efforts shown by researchers using data-driven FDI methods for satellite ADCS and sophisticated systems. In [9], a learning-based, decision tree (DT) approach is demonstrated. In [10], five fault classifiers were utilized for gas turbine engines: Support Vector Machine (SVM), the nearest neighbor classifier (K-NN), probabilistic neural network (PNN), Gaussian mixture models (GMM), and principal component analysis (PCA).

It has been shown in previous research that ensemble machine learning techniques complemented with specific classifier methods have proven effective for their applications. The novelty this work presents is the use of ensemble learning techniques for ADCS FDI on satellites. The case study explored in this paper uses a controls model of a RW to generate training datasets for the proposed ensemble machine learning algorithm.

The remainder of this paper is organized as follows: in Section 2, the problem at hand is formally presented. In Section 3, the methodology employed to solve the problem is explained. In Section 4, the case study used to evaluate the performance of the proposed methodology is detailed. Section 5 provides a preliminary finding of the study, and Section 6 concludes the paper with closing remarks and future work recommendations.

2 Problem Definition

In this section, the problem of fault diagnosis in components of a general nonlinear system is formally stated. Consider the following nonlinear system described in discrete-time state space representation:

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

where $\xi_k \in \mathbb{R}^n$ is the state vector at time step k, $u_k \in \mathbb{R}^m$ is the control input vector, $\theta_k \in \mathbb{R}^l$ is the system parameter vector, $y_k \in \mathbb{R}^m$ is the measurement vector, $w_k^{\xi} \in \mathbb{R}^n$ is the additive process noise for states, $w_k^{\theta} \in \mathbb{R}^l$ is the additive process noise for parameters, $v_k \in \mathbb{R}^m$ is the additive measurement noise. $f(\cdot)$ is a nonlinear process model, and $g(\cdot)$ is a nonlinear measurement model where in the case of full state measurement $y_k = \xi_k + v_k$ is considered.

The objective is to design and develop a data-driven FDI scheme that is capable of autonomously isolating the location of faults in the system under the following assumptions: *Assumption (i).* The control signal and the state vector remain bounded prior to and after a fault. *Assumption (ii).* There are no fault occurrences from the inception of a fault to its isolation. *Assumption (iii).* Fault severity changes are ‘‘slow’’ compared to the system output dynamics. *Assumption (iv).* All additive noises in the system are bounded and of white Gaussian form.

In this paper, it is assumed that the system component faults are reflected as changes in the physical system parameters [11]. Therefore, detecting changes in system parameters can be representative of faults in the system. The faulty system can be described as in Eq. (1) also known as multi-parameter fault model with:

$$\theta_k = \theta_0 + \alpha_k \tag{2}$$

where $\theta_0 \in \mathbb{R}^l$ is the nominal parameter values vector and $\alpha_k \in \mathbb{R}^L$ is the fault parameter vector containing L fault elements. The fault model given by Eq. (2) enables one to state the problem of nonlinear fault diagnosis in the form of an on-line nonlinear parameter tracking problem. Furthermore, a single-parameter fault model, as described below, can be defined for fault isolation. Consider the multi-parameter fault model given by Eq. (2) with L fault parameters. One can extract L single-parameter models, $\Omega_i, i = 1, \dots, L$, from the model in Eq. (2) as follows [11]:

$$\Omega_i: \{\theta_k^i = \theta_0^i + \alpha_k^i \qquad i = 1, \dots, L. \tag{3}$$

The goal of the data-driven algorithm is then set to classify the current state of the system as one of the possible L faulty cases, once a fault is detected by employing each single-parameter fault model in Eq. (3), where the ith parameter model will essentially represent the ith system parameter θ_k^i and consequently its fault parameter, namely α_k^i.

3 Methodology

This section provides details of the proposed methodology to address the abovementioned problem. The software was developed in Python with the use of Scikit-learn, a python package for machine learning, which built the ensemble machine learning classifier.

3.1 Data Preprocessing

For the machine learning algorithm to accurately differentiate the various fault scenarios, proper time domain features are generated as the input matrix for the training set, as opposed to raw time series data. These features all derive from the residual time series, $r \in \mathbb{R}^{N\times 6}$, generated from the difference between the nominal and faulty as

$$r = \left[\left(q_{k,healthy} - q_{k,faulty}\right) \left(\omega_{i,healthy} - \omega_{i,faulty}\right) \right] \tag{4}$$

where N represents the number of data points in the time domain of a given scenario dataset. The residual matrix only contains the first three quaternion values in the output dataset, as the fourth quaternion is dependent on the three and thus invaluable as a feature. Therefore, the residual matrix defines $k \in \mathbb{N}, k = 1,2,3$.

The residual matrix is then used as the input for various time domain feature functions, described in Table I [12]. The time domain features used are as follows:

Table I – List of features used for data-preprocessing in X_{set} [12]

Feature	Equation	Feature	Equation
Mean ($\bar{r}$)	$\bar{r} = \frac{1}{n}\sum_{n=1}^{i} r_i$	Peak (r_p)	$r_p = \max(r_i)$
Root Mean Square (r_{rms})	$r_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} r_i^2}$	Standard Deviation (r_{std})	$r_{std} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(r_i - \bar{r})^2}$
Skewness (r_{skew})	$r_{skew} = \frac{\sum_{i=1}^{n}(r_i - \bar{r})^3}{(n-1)r_{std}^3}$	Kurtosis (r_{kur})	$r_{kur} = \frac{\sum_{i=1}^{n}(r_i - \bar{r})^4}{(n-1)r_{std}^4}$
Mead Absolute Deviation (r_{mad})	$r_{mad} = median(\lvert r_i \rvert)$		

These generated residual properties will act as the feature matrix, $X_{set} \in \mathbb{R}^{S\times 36}$, for a given dataset corresponding to a fault scenario

$$X_{set} = \left[\bar{r}\ r_p\ r_{rms}\ r_{std}\ r_{skew}\ r_{kur}\ r_{mad}\right] \tag{5}$$

where S is the size of the samples, in this case, 4800 (16 scenarios × 300 datasets per scenario). The target value matrix, $Y_{set} \in \mathbb{N}$, corresponds to the given dataset's fault scenario.

The X_{set} and Y_{set} are split into an 80:20 test and train set, where the entire set 80% (3840 datasets) is used for training and 20% (960 datasets) are used for testing. This splitting of the dataset yields four new sets: X_{train}, X_{test}, Y_{train}, Y_{test}.

3.2 Ensemble Machine Learning Classifiers

Various machine learning classifiers were chosen for comparison with ensemble machine learning methods, both to verify proper implementation and analyze result accuracies. Specifically, the methods utilized for FDI include Adaboost Random Forest (RF), Adaboost Decision Tree (DT), k-Nearest Neighbors (k-NN), and Multilayer Perceptron (MLP) neural network.

The ***RF classifier*** is given n training subsets and trains several decision tree classifiers with a random subset of features. Each tree then tries to predict the outcome, and averaging is used to compensate for the bias. ***DT classifiers*** create a tree with decision nodes, iteratively developed with data subsets with the goal of a prediction output at the node with the least cross-validated error. DT hyperparameters were kept to default values in the scikit-learn library. ***AdaBoost*** is a meta-classifier that enhances the RF and DT classifiers, done by calculating the weighted error rate of incorrectly estimated outcomes and iteratively adjusts the weight of corresponding trees in the ensemble. The RF hyperparameters were selected as follows: The number of trees in the forest is set to 200, the maximum depth is set to 5, and the number of features to consider when looking for the best split is set to $\sqrt{N_{features}}$.

The ***k-nearest neighbors*** is an instance-based classification algorithm, where classification is based on majority voting of the nearest neighboring points relative to a given point, where k is the number of neighbors to find for voting. The number of neighbors to use was set as $k = 3$. ***Multilayer perceptron neural network*** is a subset of feedforward artificial neural networks, consisting of an input layer, N hidden layers, and an output layer. The MLP optimizes the log loss function, a measure of the predicted probability divergence from the actual input, using the stochastic gradient-based optimizer. MLP hyperparameters were kept to default values in the scikit-learn library. ***Principal Component Analysis*** (PCA) was also utilized to reduce the dimensionality of the data into 3 dimensions, in order to visually represent the data for human interpretation and aiding in algorithm debugging and diagnostics (see Figure 3).

4 Case study

To evaluate the performance of the proposed FDI, an attitude control subsystem (ACS) of a three-axis stabilized satellite is considered where actuator components experience faults.

4.1 Satellite attitude

The attitude dynamics of a rigid body satellite controlled by reaction wheels can, in general, be described by the following nonlinear differential equation:

$$J\dot{\omega}_{BI}^{B} = -\omega_{BI}^{B} \times (J_s\omega_{BI}^{B} + AJ_w\omega_{RW}) - A\tau_{RW} + \tau_e \tag{6}$$

where $\omega_{BI}^{B} \in \mathbb{R}^{3\times1}$ is the angular velocity of the spacecraft relative to the inertial frame expressed in the body frame. $\tau_e \in \mathbb{R}^{3\times1}$ is the external torque. $A \in \mathbb{R}^{3\times4}$ maps the influence of the actuators to the principal axes of the spacecraft. J is defined as $J = J_s - AJ_wA^T$ where $J_s \in \mathbb{R}^{3\times3}$ is the moment of inertia of the spacecraft including the actuators. $J_w \in \mathbb{R}^{4\times4} = diag([J_{w1}, J_{w2}, J_{w3}, J_{w4}])$ denotes the axial moment of inertia of each reaction wheel. $\omega_{RW} \in \mathbb{R}^{4\times1}$ denotes the axial angular velocity of the reaction wheels. It should be noted that for the rest of this work, ω_{RW} is represented without its subscript simply as ω, not to be confused with the satellite angular velocities that have sub and superscripts. τ_{RW} is the torque generated by RWs.

The kinematic equations for the spacecraft using quaternions can be formulated as

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

where $\bar{q} = \left[\bar{e}\sin\left(\frac{\Phi}{2}\right)/\cos\left(\frac{\Phi}{2}\right)\right] = \begin{bmatrix}q_v\\ q_4\end{bmatrix}$ is unit quaternion, Φ denotes the principal angle, $\bar{e} = [e_1, e_2, e_3]^T$is the principal axis from Euler's theorem ($e_1^2 + e_2^2 + e_3^2 = 1$). $q_4 \in \mathbb{R}$ and $q_v \in \mathbb{R}^{3\times1} = [q_1, q_2, q_3]^T$ denote the Euler parameters representing the spacecraft body frame orientation with respect to the orbital frame where $q_v^T q_v + q_4 = 1$. $I \in \mathbb{R}^{3\times3}$ is the identity matrix and $q_v^{\times}$ is the skew-symmetric matrix of the quaternion vector.

To assess the performance of the proposed FDI scheme, based on the above model, a highly accurate simulation of a three-axis stabilized satellite has been developed. The simulation model consists of the above nonlinear satellite attitude dynamics, a high-fidelity nonlinear model of the reaction wheel [13] and a sliding mode controller (SMC) that is designed to stabilize the satellite. Figure 1 depicts the block diagram of the closed-loop representation of the ACS subsystem that has been simulated in this section.

4.2 Actuators

The selection of reaction wheels (RWs) for attitude control is well justified due to their popularity in active-satellite-attitude-control. The reaction wheels considered in this work are ITHACO 'type A' reaction wheels that are currently being manufactured by Goodrich Corporation. A high fidelity nonlinear model of the reaction wheel has been obtained from Bialke [13] and has been integrated into the ACS dynamics. This high-fidelity model is also required for enhancing the robustness of the FDI scheme proposed in this work with respect to modeling errors. The nonlinear model of the RW, including discontinuous functions approximated with sigmoidal functions, can be expressed as follows [14]:

$$\begin{aligned}\dot{I}_{RW} &= G_d\omega_d[f_3(\omega, I_{RW}) - f_5(\omega)] - \omega_d I_{RW} + G_d\omega_d V_{Comm}\\ \dot{\omega}_{RW} &= \frac{1}{J_w}\{f_1(\omega) + k_t I_{RW}[f_2(\omega) + 1] - \tau_v\omega - \tau_c f_4(\omega) + \tau_{noise}\}\end{aligned} \tag{8}$$

where f_1 and f_2 account for motor disturbances, f_3 accounts for the EMF torque limiting block, f_4 accounts for the analytical approximation of the sign function in the Coulomb friction block, f_5 represents the speed limiter block, and V_{Comm} is the torque command voltage, which is input from the controller [8].

4.3 Controller

To obtain the desired attitude of $q_d \in \mathbb{R}^{4\times1}$ and $\omega_d \in \mathbb{R}^{3\times1}$, a simplified version of a nonlinear sliding mode controller is adapted from [15] with the quaternion tracking error is defined as

$$\begin{aligned}q_e &= q_{d4}q_v - q_4 q_{dv} + q_v^{\times} q_{dv}\\ \mathrm{q_{e4}} &= q_{d4}q_4 + q_{dv}^T q_v\end{aligned} \tag{9}$$

where $q_e^T q_e + q_{4e}^2 = 1$. The corresponding rotation matrix $C_e = C(q_e, q_{4e})$ is given by

$$C_e = (q_{4e}^2 - q_e^T q_e)I + 2q_e q_e^T - 2q_{4e}q_e^{\times} \tag{10}$$

where $C_e^T C_e = 1$, $\|C_e\| = 1$, $\det(C_e) = 1$, $\dot{C}_e = -\omega_e^{\times} C_e$, and I is the identity matrix. Next, the relative angular velocity $\omega_e \in \mathbb{R}^{3\times 1}$ is defined as follows

$$\omega_e = \omega_{BL}^B - C_e \omega_d \tag{11}$$

Given these error terms, the sliding manifold is defined as

$$\sigma = \omega_e + \lambda sgn(q_{4e}) q_e \tag{12}$$

where $\lambda > 0$ is the sliding gain and $sgn(q_{4e})$ is the sign function for q_{4e}. The required control command is then obtained from

$$u_r = -\eta A^T \frac{\sigma}{\|\sigma\|} \tag{13}$$

with η defined as

$$\eta = p_0 + p_1 \|X\| \tag{14}$$

where p_0 and p_1 are known as positive constants, and $X \in \mathbb{R}^{6\times 1} = [q_v, \omega_{BL}^B]^T$. With the required control command (u_r) and actuator dynamics available, it is possible to calculate the simplified required input to the actuators adapted from [15] as

$$V_{comm} = R_a K_t^{-1} u_r \tag{15}$$

In this work, all control parameters λ, p_0, and p_1 are equal to 1 based on values given in [15] and validated simulation results. It is important to note that the controller outputs are bounded to $\pm 5\ V$ due to the saturation limits on the RWs [13].

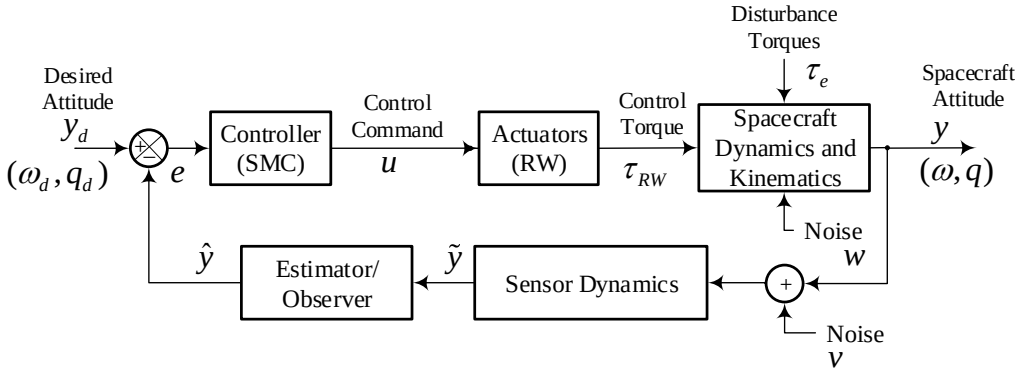


Figure 1 Proposed FDII simulation setup

4.4 Fault formulation

Four identical reaction wheels are used in a three-axis stabilized satellite. A standard four-wheel reaction wheel assembly is considered for this study. A central hierarchical FDI unit is considered for health monitoring of the whole system. The simulation data are obtained from closed-loop ACS simulation of a three-axis stabilized low earth orbit (LEO) satellite. Transient time-varying faults are injected into two of the reaction wheel components, namely motor current, and bus voltage. Faults in the motor current are modeled and injected as variations in motor torque gain k_t. Faults in the bus voltage are modeled and injected as drops in the voltage of the power bus V_{bus}. Consequently, the two fault parameters are defined. In other words, the multi-parametrized fault model is obtained by replacing $V_{bus,j}$ with $V_{bus,j_0} + \alpha_j^1$, and replacing $k_{t,j}$ by $k_{t,j_0} + \alpha_j^2$ where j is the index for the reaction wheel unit among the four units considered in this case study and α_j^i are unknown fault parameters that indicate the possible presence of faults in the bus voltage and motor current of each wheel. Due to the additive form of the fault parameters introduced above, the value for α_j^i in the healthy condition would be zero and at any given time, the deviation from zero for any of these fault parameters could potentially be an indicator of the

severity of the fault and its size. It is important to note that the fault is introduced in the actuator non-measurable parameters; however, in the fault isolation, only system satellite outputs namely q and ω are monitored for faulty behavior.

4.5 Fault scenario

To calculate the total number of combinations for a four-wheel RW assembly, we use combination theory, and it sums up to 16 unique combinations. To be able to refer to each combination (fault scenario) more conveniently, all possible cases are assigned with a number in Table II where the faulty-wheel number is defined in **Figure 2** for each RW assembly configuration.

Table II - Fault scenarios for RW assemblies

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

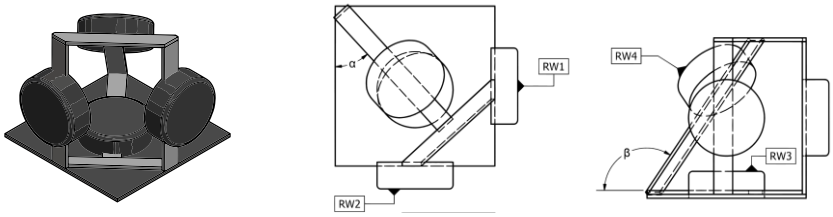


Figure 2 RW assembly of standard four-wheel configuration

4.6 Dataset Structure

The data is structured in input format for dataset generation as listed in Table III.

Table III - Fault scenarios for RW assemblies

Index	Description	Possible values	Unit
1	Fault Scenario	0-15	integer
2	Kt in fault	0/1	binary
3	Vbus in fault	0/1	binary
4	Kt inception time	0-60	sec
5	Vbus inception time	0-60	sec
6	Kt fault duration	0-20	sec
7	Vbus fault duration	0-20	sec
8	Kt severity	0.029	Nm/A
9	Vbus severity	4-8	Volt

For every single simulation, a set of items in Table III are inputted in the MATLAB simulation for the closed-loop system, and the output of the simulation is stored in an output format as presented in Table IV.

Table IV - Fault scenarios for RW assemblies

Item	Unit	Description
time	[sec]	the time in simulation starting from 0 with increments of 0.1
$q_{i_{healthy}}$	[]	the nominal quaternion parameter i for satellite
$\omega_{i_{healthy}}$	[rad/sec]	the nominal angular speed for the satellite around i (x,y,z) axis
$I_{RW_{i_{healthy}}}$	[A]	the nominal current of RW_i onboard satellite
$\omega_{RW_{i_{healthy}}}$	[rad/sec]	the nominal angular speed of flywheel of RW_i onboard satellite
$q_{i_{faulty}}$	[]	the faulty quaternion parameter i for satellite
$\omega_{i_{faulty}}$	[rad/sec]	the faulty angular speed for the satellite around i (x,y,z) axis
$I_{RW_{i_{faulty}}}$	[A]	the faulty current of RW_i onboard satellite
$\omega_{RW_{i_{faulty}}}$	[rad/sec]	the faulty angular speed of flywheel of RW_i onboard satellite

This data is then used as input for training and test of the proposed fault isolation machine learning algorithm.

5 Results and discussion

In order to achieve a baseline acceptable accuracy score, many alterations were made to the algorithm to simplify the case study. The following changes were made to achieve a reasonable accuracy score:

1. half of the total generated dataset was used to increase algorithm runtime,
2. only scenarios 0-4 were analyzed to determine functional baseline fault identification,
3. residual data timeframe was determined from known fault inception time and duration
4. the fault severities were set to a constant value if a fault occurred to reduce effects of fault severity deviation

Table V presents the output of the various machine learning classifier, comparing the accuracy scores of the predicted test dataset scenarios, y_{pred} to the actual scenario, y_{test}.

Table V - Classifier Method and corresponding $\boldsymbol{y_{pred}}$ to $\boldsymbol{y_{test}}$ accuracy scores

Classifier Method	Accuracy Score
k-Nearest Neighbor	37.16%
Multilayer Perceptron Neural Network	38.51%
Adaboost Decision Tree	52.03%
Adaboost Random Forest	58.78%

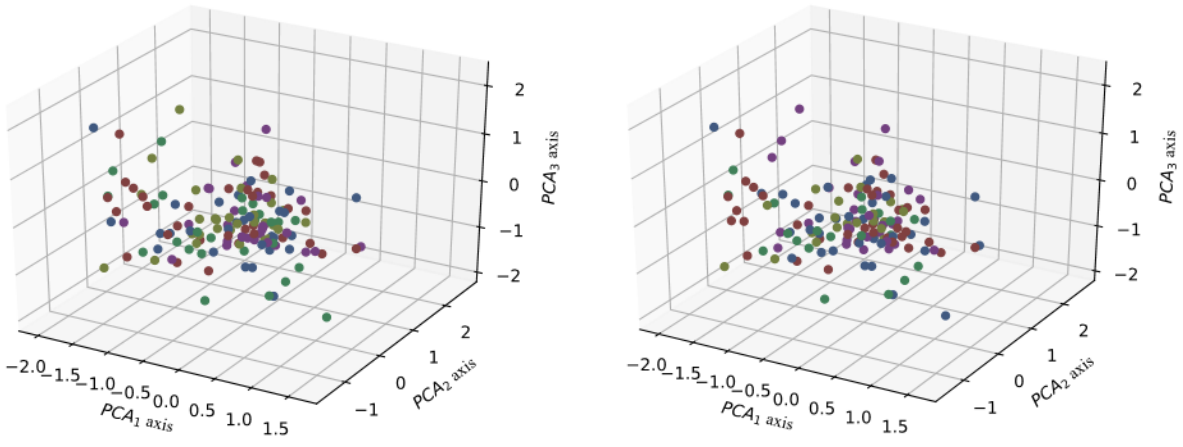
The confusion matrix for the Adaboost RF classifier is

Table VI - Confusion Matrix for Adaboost RF $\boldsymbol{y_{pred}}$ vs $\boldsymbol{y_{test}}$

$\boldsymbol{y_{pred}/y_{test}}$	0	1	2	3	4
0	29	2	1	1	1
1	2	17	1	5	5
2	2	0	15	4	5
3	3	2	1	14	9
4	4	4	3	6	12

Based on the results of Adaboost RF, it is evident that for the given dataset provided, the proposed ensemble method would improve the classification of ADCS fault scenario. However, a classifier accuracy score of ~59% is not suitable for ADCS FDI, as most machine learning applications aim to achieve an accuracy score of 95-99%. There are several hypotheses as to why the methodology failed to produce reliable scenario predictions:

1. There are not enough distinguishable features in the X_{set} for the classifier to determine the scenario accurately
2. The residual time series, r, does not contain enough sets that exhibit relevant deviations caused by faults meaning that the data-driven approach may require more sets other than q_i and ω_i
3. The hyperparameters were not selected based on hyperparameter optimization methods



(a) (b)

Figure 3 PCA 3D X_{train} values colored by (a) actual scenario number (b) predicted scenario number

6 Conclusion

In this study, the application of employing the ensemble methods to classify faults in a nonlinear system was explored. A 3-axis controlled satellite was simulated to generate data for pre-processing of the machine learning methods employed. The residual was generated as the difference between the nominal and faulty output, and the results as a time-series output were transformed into a time-domain feature-space, used to extract unique distinguishers between different scenarios. k-Nearest Neighbor, Multilayer Perceptron Neural Network, Adaboost Decision Tree and Adaboost Random Forest were employed as the methods of choice for the classification of the data set. The classification of the dataset resulted in an accuracy of ~59% for Adaboost Random Forest, the highest of the classification methods explored.

Future work on this ADCS FDI application is to be done to increase the accuracy score to acceptable levels for confident application. Diagnosing the issues outlined in the discussion section of this study would the first line of actions in that direction. The inclusion of all scenarios and fault severities would follow, optimizing the classifier hyperparameters to increase the prediction accuracy of all scenarios and fault types. Once accomplished, the proposed ensemble FDI would lend invaluable analytical redundancy and health-monitoring capabilities to a host of non-linear systems and would particularly be useful in space applications where hardware redundancy is limited and direct access to system components is costly.

7 References

[1] Doncaster, B., Williams, C., and Shulman, J., *2018 Nano/Microsatellite Market Forecast, 8th Edition*, Atlanta, GA: 2018

[2] Larson, J. R. W. W. J., *Space Mission Analysis and Design*, Space Technology Library/Microcosm Press, 2005

[3] Zhang, Q., An, X., Gu, J., Zhao, B., Xu, D., and Xi, S., "Application of FBOLES-a prototype expert system for fault diagnosis in nuclear power plants," *Reliability Engineering and System Safety*, 1994,

doi: 10.1016/0951-8320(94)90015-9

[4] Nimmo, I., "Abnormal situation management," *Proceedings of the Industrial Computing Conference*, 1995

[5] Isermann, R., Schwarz, R., and Stölzl, S., "Fault-tolerant drive-by-wire systems," *IEEE Control Systems Magazine*, 2002, doi: 10.1109/MCS.2002.1035218

[6] Simani, S., Castaldi, P., and Tilli, A., "Data-driven approach for wind turbine actuator and sensor fault detection and isolation," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2011

[7] Rahimi, A., Kumar, K. D., and Alighanbari, H., "Enhanced Adaptive Unscented Kalman Filter for Reaction Wheels," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 2, Apr. 2015, pp. 1568–1575, doi: 10.1109/TAES.2014.130766

[8] Rahimi, A., Kumar, K. D., and Alighanbari, H., "Fault estimation of satellite reaction wheels using covariance based adaptive unscented Kalman filter," *Acta Astronautica*, Vol. 134, May 2017, pp. 159–169, doi: 10.1016/j.actaastro.2017.02.003

[9] Barua, A., Sinha, P., and Khorasani, K., "A diagnostic tree approach for fault cause identification in the attitude control subsystem of satellites," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 45, No. 3, Jul. 2009, pp. 983–1002

[10] Donat, W., Choi, K., An, W., Singh, S., and Pattipati, K., "Data Visualization, Data Reduction and Classifier Fusion for Intelligent Fault Diagnosis in Gas Turbine Engines," *Journal of Engineering for Gas Turbines and Power*, 2008, doi: 10.1115/1.2838993

[11] Sobhani-Tehrani, E., Talebi, H. A., and Khorasani, K., "Hybrid fault diagnosis of nonlinear systems using neural parameter estimators," *Neural Networks*, Vol. 50, Feb. 2014, pp. 12–32, doi: 10.1016/j.neunet.2013.10.005

[12] Park, D., Kim, S., An, Y., and Jung, J.-Y., "LiReD: A Light-Weight Real-Time Fault Detection System for Edge Computing Using LSTM Recurrent Neural Networks," *Sensors*, Vol. 18, No. 7, Jun. 2018, p. 2110, doi: 10.3390/s18072110

[13] Bialke, B., "High Fidelity Mathematical Modeling of Reaction Wheel Performance," *1998 Annual AAS Rocky Mountain Guidance and Control Conference, Advances in the Astronautical Sciences*, Breckenridge, CO: 1998, pp. 483–496

[14] Sobhani-Tehrani, E., and Khorasani, K., "Identification For Nonlinear Systems Using Hybrid Approach," *Master's Thesis*, 2008, pp. 12-13,15,18,37,92

[15] Kumar, K. D., Godard, Abreu, N., and Sinha, M., "Fault-tolerant attitude control of miniature satellites using reaction wheels," *Acta Astronautica*, Vol. 151, No. May, Oct. 2018, pp. 206–216, doi: 10.1016/j.actaastro.2018.05.004