A Framework for Data–Driven Fault Diagnosis of Numerical Spacecraft Propulsion Systems

Kazushi Adachi[1], Samir Khan[2], Shinichi Nakasuka[3], Kohji Tominaga[4], Seiji Tsutsumi[5], Yu Daimon[6] and Taiichi Nagata[7]

[1,2,3] Department of Aeronautics and Astronautics, The University of Tokyo, Tokyo, 113–0033, Japan

[1]adachi–kazushi@g.ecc.u–tokyo.ac.jp
[2]kviz3671@g.ecc.u–tokyo.ac.jp

Japan Aerospace Exploration Agency, Tsukuba, Ibaraki, 305–8505, Japan

[4]tominaga.kohji@jaxa.jp
[5]tsutsumi.seiji@jaxa.jp
[6]daimon.yu@jaxa.jp
[7]Nagata.taiichi@jaxa.jp

ABSTRACT

The increasing complexity of space exploration missions introduces significant challenges to maintaining spacecraft health, particularly in the propulsion systems, due to the inherent communication delays with Earth. This research introduces a novel framework for constructing a data-driven diagnostic system using data generated from both simulation models and actual spacecraft telemetry. A detailed trade-off analysis between diagnostic accuracy and computational efficiency is conducted, and an extensive literature review positions the framework within the current research landscape. Future work will focus on enhancing the framework's capability to address unknown anomalies through advanced machine learning techniques. The study addresses the limitations imposed by computational resources and sensor installation constraints through Sequential Forward Selection (SFS) for optimized sensor placement and feature selection. The framework's effectiveness is demonstrated through implementation on a microcomputer, showing promising results in terms of diagnostic accuracy and processing speed, thus highlighting its potential for onboard spacecraft application. This study not only advances the autonomous capabilities of spacecraft in deep space, but also contributes to the broader field of Prognostics and Health Management (PHM) by providing a scalable, efficient approach to fault diagnosis in critical spacecraft systems. The suggested methodology illustrates a promising approach to optimizing diagnostic scenarios for spacecraft systems. However, the trade-offs observed necessitate a careful consideration of task-specific requirements and the potential need for adjustments to maintain a high level of accuracy alongside computational efficiency.

https://doi.org/10.36001/ijphm.2024.v15i3.3866

1. INTRODUCTION

In recent years, the ambitions for deep space exploration have expanded dramatically, with initiatives like the Lunar Gateway and Mars exploration pushing the boundaries of human achievements beyond Earth's orbit [1,2]. These ambitious missions, however, come with their own set of challenges, particularly the significant communication delays between spacecraft and Earth. For instance, messages to and from Mars can take anywhere from 3 to 20 minutes each way, complicating immediate responses to any issues that may arise on spacecraft in deep space [3]. One subsystem of spacecraft that stands crucial in this context is the propulsion system, responsible for orbital transfer and attitude control. The inherent communication delays underscore the necessity for autonomous onboard diagnostic systems capable of detecting and identifying faults independently. This autonomy is vital for mission success, as delayed responses due to communication lag could have dire consequences.

However, achieving autonomy in spacecraft operations is no small feat. It involves overcoming significant hurdles, such as the limitations in computational resources and the restrictions on the number of sensors that can be installed. Spacecraft are equipped with computers that must withstand extreme conditions like harsh radiation and thermal environments, often resulting in the use of lower-performance computing systems. Moreover, considerations related to weight and communication channels limit the number of sensors that can be integrated within the propulsion system, challenging the design of effective diagnostic systems that must work within these constraints, using minimal data and computational resources.

The field of Prognostics and Health Management (PHM) in spacecraft primarily focuses on anomaly and fault detection. Early detection of these anomalies is crucial as it allows for timely corrective actions, thereby enhancing the reliability and longevity of spacecraft systems and potentially preventing mission-critical failures. Model-based methods offer clear insights into system behavior and are adept at addressing unknown anomalies using established rules. However, their complexity and the high cost of model construction pose significant challenges, especially for systems with restricted computational capabilities like spacecraft [4]. On the other hand, data-driven methods, characterized by their lower costs and ease of modification, do not require intricate knowledge for model development and operate at lower computational costs during inference. Yet, these methods heavily rely on the availability and quality of data, which can be a limiting factor for spacecraft due to the expensive nature of physical experiments and the scarcity of data.

To improve the practical relevance and reliability of the framework, future work will integrate actual spacecraft data into the validation process. This integration aims to corroborate the simulated results and ensure the framework's applicability to real-world spacecraft systems. Preliminary steps include collaborations with space agencies to access telemetry data from existing missions. By employing Sequential Forward Selection (SFS) for optimized sensor placement and feature selection, the framework aims to address the challenges posed by computational and sensor installation constraints. This approach not only caters to anticipated faults but also deals with unknown anomalies, demonstrating its feasibility through implementation on a microcomputer with a proven track record in spacecraft applications. This method demonstrates a significant development in enhancing the self-governance of spacecraft. This research underscores the importance of advanced simulation models in enhancing the reliability and safety of spacecraft propulsion systems by providing a:

- Realistic model to test and improve lower fault diagnosis methods.
- The ability to accurately simulate and diagnose potential faults and anomalies throughout the propulsion system ensures the success and safety of future space missions, particularly in the face of the unpredictable and challenging conditions encountered in deep space exploration.

2. RELATED WORK

To position the proposed framework within the current research landscape more effectively, a comprehensive literature review has been conducted. This review encompasses various fault diagnosis methods, including optimization techniques like Genetic Algorithms and Particle Swarm Optimization, and machine learning approaches such as Random Forests and Neural Networks. The expanded literature review highlights the unique contributions and advantages of the proposed framework compared to existing methods. While model-based approaches excel in interpretability and handling data absent from training datasets, data-driven methods stand out for their low construction costs, lack of requirement for expert knowledge during model building, and inference computational costs. Despite their distinct advantages, both methodologies face unique challenges when applied to the stringent and limited environment of spacecraft systems, driving the need for innovative solutions like the proposed framework. Kolcio et al.'s work [5]–[8] introduced the "MONSID System," a model-based health management system tailored for Mars rovers. This system processes sensor and command data through physical models, comparing these results against actual sensor data to identify discrepancies indicative of system anomalies. Its ability to pinpoint the exact component failure makes it a robust solution against unforeseen anomalies, albeit at the cost of requiring intricate, high-fidelity motion models and real-time computation. Similarly, Aaseng et al.'s study [9] on the Orion Exploration Flight Test 1 (EFT-1) developed a model-based fault diagnosis system for the spacecraft's Electrical Power System (EPS). By inputting both test data and actual telemetry, the system monitored the EPS's state, adjusting based on output discrepancies. This method proved effective, operating at 1Hz on a standard Linux notebook PC, though it necessitated an in-depth analysis of technical documentation and expert consultation.

Extending model-based methodologies to Guidance, Navigation, and Control (GNC) systems, Henry et al.'s research [10–13] showcased high-precision fault detection and control tailored for ESA spacecraft rendezvous missions. These missions, where system faults could directly compromise mission objectives, benefited greatly from such precise diagnostic capabilities. However, the dependency on specific spacecraft components and sensors underscores a limitation: each mission demands a bespoke system.

In contrast, data-driven approaches, exemplified by Gao et al.'s study [14], leverage existing data without the need for physical system models. Utilizing Principal Component Analysis (PCA) [15] and Support Vector Machines (SVM) [16], this methodology demonstrated high accuracy in fault detection using spacecraft telemetry data. Despite its effectiveness, the reliance on linear feature extraction through PCA might limit its applicability to more complex, nonlinear system behaviors. Hundman et al.'s approach [17] introduced anomaly detection using Long Short-Term Memory (LSTM) [18] networks combined with dynamic thresholding, demonstrating high accuracy in detecting telemetry anomalies from satellites and Mars rovers. This method's reliance on deep learning and dynamic data thresholding presents a novel way to handle the inherent data limitations and computational constraints faced by spacecraft systems. However, its primary focus on univariate data may overlook anomalies stemming from system interactions, highlighting a gap in its applicability for onboard spacecraft execution.

The connection of model-based and data-driven methods in spacecraft health management reveals a shared goal: to ensure mission success through effective fault diagnosis and management. While model-based approaches offer deep system insights and robust anomaly detection, they require extensive expertise and computational resources. Conversely, data-driven methods provide flexibility, lower costs, and adaptability, albeit dependent on the availability and quality of data. This synthesis of methodologies underlines the evolving landscape of spacecraft health management, where the integration of both approaches could pave the way for more autonomous, resilient, and successful deep space missions.

2.1. Research on Health Management of Spacecraft Propulsion Systems

The management of spacecraft health, particularly propulsion systems, is critical for the success of space missions. As spacecraft venture further into deep space, the ability to detect and diagnose system abnormalities autonomously becomes increasingly vital. Traditional diagnostic methods rely on data from spacecraft components, such as solar sensors and attitude controllers. Research by Gueddi et al. [19], Mansell et al. [20], and Xiao et al. [21] has have explored data-driven diagnostic systems that monitor such components. Yet, these methods often detect issues only after they have impacted the spacecraft's attitude, necessitating corrective action that consumes additional propellant.

An emerging area of interest in spacecraft health management is the monitoring of fluid flow within propulsion systems, specifically focusing on the phenomenon known as the water hammer [22]. This phenomenon, a sudden pressure surge or wave resulting when a fluid in motion is forced to stop or change direction suddenly, can indicate the presence of anomalies within the propulsion system. The water hammer effect, characterized by equations that describe the fluid's behavior under such conditions [24][25], offers a novel approach to detecting faults. Notably, the work of Kumar et al. [26] and Lecourt et al. [27] has advanced the understanding and simulation of water hammer phenomena in environments that mimic spacecraft propulsion systems.

Building on this foundation, Kawazu et al [28–33] have utilized simulation models to assess and manage the health of spacecraft propulsion systems. These models, developed using the Modelica language [34] and SimulationX software [35], enable detailed analysis of water hammer effects and other fluid dynamics within the system. Such simulations provide valuable insights into potential anomalies and the overall health of the propulsion system. Furthering this line of research, Tominaga et al. [36] created a 1D-CAE simulation model based on a quad-engine configuration of a spacecraft propulsion system. This model generates data for various scenarios to evaluate the system's health and identify potential faults. While subsequent studies by Minami et al. [37], Lee et al. [38], and Kato et al. [39] have applied data-driven methods to diagnose faults using this data, challenges remain in ensuring the robustness of these diagnostic systems under real-world conditions and computational constraints.

Addressing these challenges, this study proposes a fault diagnosis method designed to be resilient against mechanical noise and other real-world variables. By implementing a pre-trained model on a microcomputer with a proven track record in spacecraft applications, this method's accuracy and processing speed are assessed. The dataset for this study, developed by JAXA, offers a more realistic simulation of a spacecraft propulsion system, enhancing the potential for effective onboard fault diagnosis. This approach aims not only to advance the capabilities of spacecraft in deep space, but also to contribute to the broader field of spacecraft health management by providing a scalable and efficient solution for fault diagnosis.

The research delves into the development and application of a sophisticated spacecraft propulsion system simulation model by JAXA, designed to simulate real-world conditions and enhance the reliability and safety of spacecraft operations. This model is instrumental in understanding the intricate behaviors of spacecraft propulsion systems and forms the basis for testing and improving fault diagnosis methods.

3. SPACECRAFT PROPULSION SYSTEM MODEL

At the heart of the simulation is an 18-branch piping system, incorporating sixteen Reaction Control System (RCS) thrusters (SV1-SV16) and two main engines (ME) (SV17, 18). The model uses pure water as the working fluid, selected for its similar density and sound speed to hydrazine, a common spacecraft fuel. This fluid is propelled from the tank to various system parts under a base pressure of 2MPa. The simulation precisely controls the operation of electromagnetic valves SV1-SV4 of the RCS, allowing for open and close actions, while the main engines remain operational throughout the RCS's activity. Notably, the simulation introduces a realistic touch by incorporating random fluctuations of 0-1ms in valve timing, simulating the unpredictability found in actual spacecraft operations.

3.1. The Pressure Fluctuation Data

The model's ability to capture the dynamics of pressure fluctuations within the system is a key feature. For instance, data obtained at pressure gauge P3 under standard conditions reveals significant pressure drops coinciding with the operation of RCS valves. These drops, followed by water hammer effects caused by the valves' closing actions, provide critical insights into the system's behavior under various operational scenarios. Such detailed observations are invaluable for refining diagnostic techniques to swiftly identify and address system anomalies.

3.2. Training and Test Data: Anomalies and Faults

The simulation model generates data reflecting three primary types of system anomalies and faults:

- Solenoid Valve (SV) Faults: Faults originating from the RCS's solenoid valves (SV1-SV4), which may operate partially, affecting the flow rate of the propellant.
- Bubble Anomalies: Introduced bubbles at specific accumulator locations (BP2, 17, 32) mimic potential operational issues, impacting the propagation of pressure waves within the system.
- Simultaneous SV Faults and Bubble Anomalies: This complex scenario combines SV faults with bubble anomalies, presenting a challenging diagnostic scenario to test the system's diagnostic capabilities.

Moreover, the test data encompasses these scenarios alongside hypothetical unknown anomalies, challenging the diagnostic system to identify and classify a wide range of potential issues.

An Exploratory Data Analysis (EDA) was meticulously performed on the dataset generated from the spacecraft propulsion system simulation. This analysis was particularly focused on specific intervals: 100–400ms, 500–800ms, and 900–1200ms. These intervals were chosen for their significance in showcasing the water hammer effects triggered by the operation of the Reaction Control System (RCS) solenoid valves (SV1-SV4). The decision to focus on these segments stems from the understanding that pressure fluctuations, especially those resulting from the closing of solenoid valves, can be profoundly influenced by vibrations emanating from downstream thrusters or the combustion processes of engines. These factors become critically important when considering the operations of actual spacecraft, where such vibrations can significantly impact pressure sensor readings.

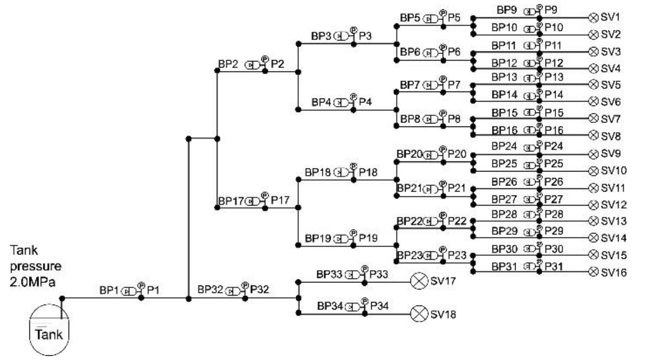


Figure 1: The schematic diagram of the numerical spacecraft propulsion system

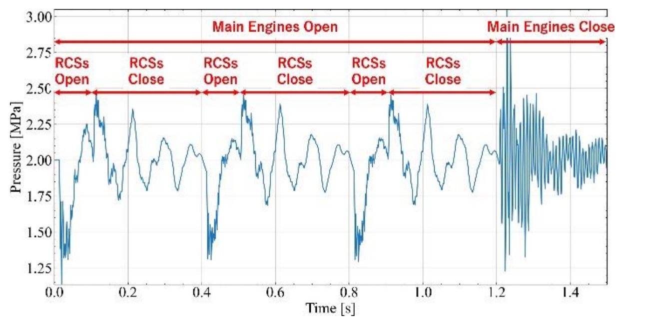


Figure 2: Example of time series pressure fluctuation data at P3

Table 1: Training data generated by Simulation Model

Condition	No of data
Normal	20
SV fault: Location: SV1,SV2,SV3,SV4 Valve opening ratio: 0%, 50%	8
Bubble anomaly: Location: BP2, BP17, BP32	6
SV fault + Bubble Anomaly: Location: SV1,SV2,SV3,SV4 Valve opening ratio: 0% 50%, BP2, BP17, BP32	24

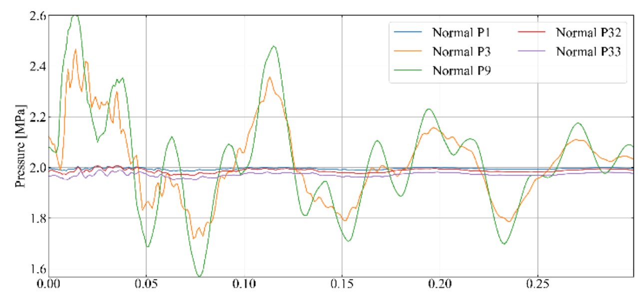
Table 2: Test data generated by Simulation Model

Condition	No of data
Normal	5
SV fault Location: SV1,SV2,SV3,SV4 Valve opening ratio: 0%, 50%	8
Bubble anomaly Location: BP2, BP17, BP32	6
SV fault + Bubble Anomaly SV Location: SV1,SV2,SV3,SV4	24

Valve opening ratio: 0% 50%, Bubble Location: BP2, BP17, BP32	
Unknown SV2,4 multi faults SV6 accidentally open with SV1–4 BP1 Bubble BP9 Bubble SV1,3 multi faults and BP2 BP2,17 multi anomalies	6

The data analysis revealed that the pressure fluctuations were more pronounced at points P9 and P3, which are proximal to the RCS solenoid valves responsible for inducing water hammer effects. This observation was consistent across both time-series data and Fast Fourier Transform (FFT) analysis results. It indicates that these points are directly impacted by the water hammer phenomenon, leading to larger pressure fluctuations.

On the contrary, locations such as P1, P32, and P33, which are situated further from the primary sources of water hammer (SV1, SV2, SV3, SV4) and are influenced by system branching, exhibited lesser fluctuations. The attenuation of the water hammer impact at these distant points suggests that the structural layout of the propulsion system plays a significant role in the distribution and magnitude of pressure fluctuations within the system. This exploratory analysis not only underscores the critical nature of water hammer effects in the context of spacecraft propulsion system health, but also highlights the importance of sensor placement and data interpretation in diagnosing potential issues. Understanding how pressure fluctuations vary across different points in the system provides valuable insights into the operational dynamics of spacecraft propulsion systems and lays the groundwork for developing more accurate and reliable diagnostic tools.



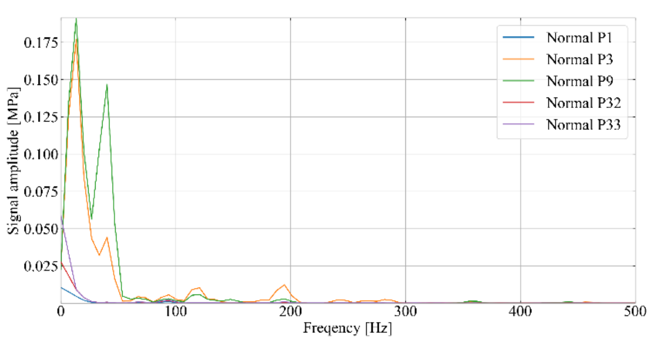


Figure 3: Example of time series data and FFT results at P1, P3, P9, P32, P33, 34

4. METHODOLOGY

A detailed trade-off analysis between diagnostic accuracy and computational efficiency has been conducted. While the optimization process significantly reduces computational load, it is crucial to evaluate the impact on diagnostic accuracy. This analysis, presented in Table 4, highlights specific scenarios where accuracy may be compromised and proposes strategies to mitigate these effects, ensuring a balanced approach that maintains high diagnostic capabilities while optimizing efficiency. The framework outlines a sequence of steps designed to process and interpret the data, facilitating precise diagnostic outcomes. These steps include:

- Scenario 1: Known Anomaly Detection: The objective here is to identify any deviations from normal operational patterns within the dataset. This step involves distinguishing between normal operational data and potential anomalies that could indicate issues within the propulsion system.
- Scenario 2: Unknown Anomaly Detection: This scenario focuses on detecting anomalies that were not previously identified or defined in the dataset. It aims to uncover new or unforeseen issues that could impact the propulsion system's performance.
- Scenario 3: Classification of SV Faults, Bubble Anomalies, and SV Faults + Bubble Anomalies: This stage classifies the detected anomalies into specific categories, including solenoid valve (SV) faults, bubble anomalies within the propulsion system, and instances where both SV faults and bubble anomalies occur simultaneously.
- Scenario 4: Classification of SV Fault Locations: After identifying SV faults, this scenario pinpoints the specific location of each fault among the solenoid valves (SV1-SV4). Accurate location identification is vital for targeted maintenance and repairs.
- Scenario 5: Classification of Bubble Anomaly Locations: Similar to scenario 4, this step aims to determine the exact locations of bubble anomalies within the propulsion system, specifically identifying whether they occur at BP2, BP17, or BP32. This information is critical for understanding the anomaly's impact on system performance and for guiding corrective actions.
- Scenario 6: Classification of SV Faults + Bubble Anomalies: The final scenario involves classifying instances where both SV faults and bubble anomalies are present. This comprehensive classification provides a nuanced understanding of the propulsion system's health, highlighting complex scenarios that may require specialized intervention.

Through this structured framework, each piece of time-series data is meticulously analyzed and categorized, enabling a thorough diagnosis of the spacecraft propulsion system. This approach not only enhances the precision of fault detection and anomaly classification but also supports the development of targeted strategies for maintaining and improving system health and reliability with:

Data Augmentation: This initial phase enhances the dataset by generating additional data points through techniques such as synthetic data generation. Specifically, we used a simulation model that closely mimics the real spacecraft propulsion system to create realistic fault scenarios. This included varying the operating conditions and introducing controlled faults to generate diverse and representative data. This step aims to enrich the dataset, ensuring a robust foundation for subsequent analysis. We explore the method to bolster the model's generalization capabilities by effectively increasing the size of the training data. This is particularly applied to water hammer events observed during the closure of the Reactor Coolant System (RCS), where a unique time-series data segment is dissected into three sub-intervals: 100-400ms, 500-800ms, and 900-1200ms. Each interval is then independently analyzed as a standalone data set.

Extracting key features from the time-series data is crucial for transforming raw data into a format that is amenable to analysis. This process involves identifying significant attributes or characteristics that effectively represent the data, facilitating easier detection of anomalies and faults. Central to our methodology is the process of feature extraction, which is the transformation of time-series data into a more malleable, non-time-series format. This step is crucial for the data to be effectively utilized by machine learning models. The need for this transformation arises from the inherent variability in the time intervals of data capture, especially due to the inconsistent operation of valves in the propulsion system. This inconsistency can result in data that is not always collected at the same, expected time intervals, with any deviation in the time of data capture potentially skewing the model's predictions. To address this, we employ a range of data processing techniques. For instance, the study leverages the Fourier Transform (FFT) to derive n peak FFT features, indicating the application of the FFT for data simplification. Furthermore, to ensure a level playing field for all the derived features, we apply a process of standardization. This normalizes the data with the equation:

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

where x is the data value, μ is the mean, and σ is the standard deviation, to ensure that all the data is on the same scale and thus treated equally by the model.

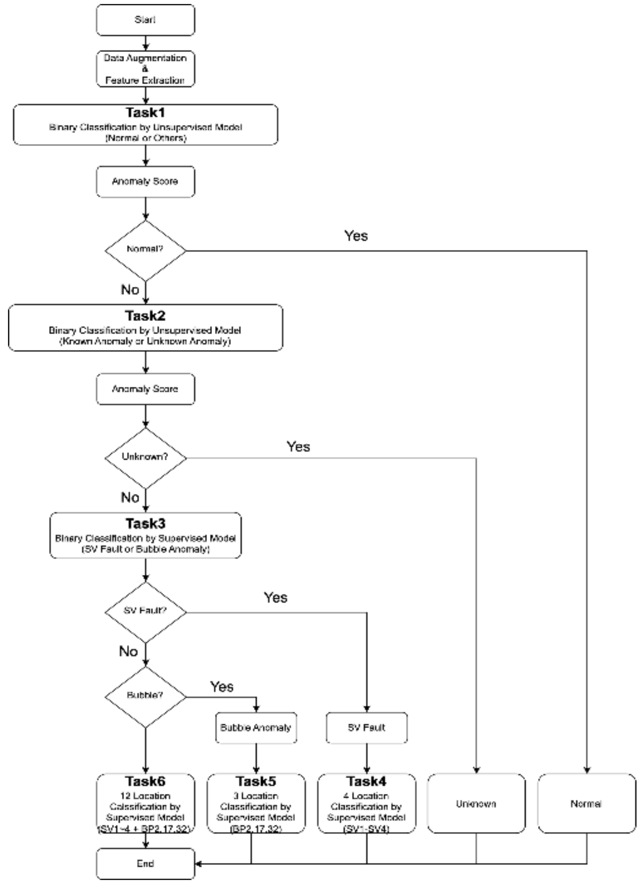


Figure 4: Flowchart of the proposed diagnostic framework

A unique challenge in this process is the high dimensionality of the data. The process of data augmentation yields three datasets for each time interval, and with 34 sensors, we are presented with a 300×34 matrix (data points × number of sensors) for each dataset. From this, we extract 17 different types of data features, both statistical and frequency domain features, for each of the 34 sensors, which results in a large, multi-dimensional dataset. The main task is then to discern the most computationally manageable and significant features, as processing the total number of all possible data points efficiently is too large.

Moreover, in the real-world design of spacecraft, the number of sensors that can be feasibly installed is bound by weight, space, and cost limitations. Therefore, it is not only a task of computational simplification but also of hardware optimization to reduce the number of sensors used without compromising the data's integrity. This optimization of the number of sensors and the data they collect is a subject of in-depth study and is set to be a subject of future work. Lastly, the data, upon being collected and optimized, is standardized. This is an essential pre-processing step that ensures that the data from each sensor is on a common scale, which is a key step in allowing the model to evaluate the data impartially.

This data pre-processing step is a prelude to the data's eventual use in training the model, with the aim of the whole exercise being to make the most of the model's potential to predict and manage the space vehicle's in-flight status and safety.

Table 3: Features extracted from time–series data

	Feature Type	No of Features
Statistical	RMS Mean Variance Kurtosis Skewness Peak to Peak Energy Crest Factor Shape Factor Clearance Factor Impulse Factor	11
FFT	First Peak Frequency First Peak Amplitude Second Peak Frequency Second Peak Amplitude Third Peak Frequency Third Peak Amplitude	6

4.1. Models

This section discusses the machine learning models employed for diagnosing the data, with a focus on Principal Component Analysis (PCA). PCA is a key dimension reduction technique that identifies the most significant axes or principal components to represent the original high-dimensional data's variability. This is achieved with minimal data loss, enhancing the data's interpretability for machine learning models.

The PCA model is initially trained with only the 'normal' data to define a standard subspace. The novelty or 'anomaly' of the test data is then quantified by the model's ability to reconstruct the new data using this pre-learned subspace. The main steps for this are:

- Mapping and reduction: The data, with p normalized features, is mapped to a lower q dimension (q<p) to capture the most relevant data structure.
- Projection and reconstruction: The model projects the data onto the new q dimensions and attempts to reconstruct it. This step uses the model's main components, derived from the most significant eigenvectors of the training data's variance-covariance matrix.
- Error analysis: The method then evaluates the test data's 'anomaly' by the error in reconstructing the new information. A large difference between the original and reconstructed data suggests an anomaly.

This error, calculated as the Euclidean norm between the test and reconstructed data, is a good measure of the system's status. A high error score indicates a high chance of the new test data being an anomaly, which is a critical part of the system's self-diagnosis. Through this method, PCA can be a powerful tool in the real-time health management of complex systems, by effectively reducing the data's complexity and making it more manageable for in-depth analysis.

One class SVM The One-Class Support Vector Machine (OC-SVM) is a specialized model designed to distinguish between 'normal' and 'anomaly' data points. It operates by learning the data space's boundary where the normal, or more common, data points are found. The model is particularly skillful at handling data that isn't linearly separable, thanks to its use of a kernel function—often a Gaussian kernel. This capability allows the OC-SVM to map the data effectively, placing normal points in high-density areas far from the data space's origin, and mapping anomalies, or data in low-density regions, closer to the origin. The OC-SVM's problem-solving process is about minimizing the value for all data points, to find the most optimal model's decision boundary. This minimization also ensures that a pre-defined fraction of the data, controlled by a hyperparameter, is found near the model's decision boundary. This method of operation makes OC-SVM a robust solution for real-time data analysis, enabling the detection of data points that deviate from the 'normal' range, which is a crucial part of the self-diagnosis in health management systems.

K-NN The k-Nearest Neighbors (k-NN) method, initially proposed by Evelyn et al., serves as a versatile tool in the realm of supervised learning, primarily for classifying data points into distinct classes. Beyond its conventional application, k-NN has been adapted for anomaly detection, leveraging the concept of 'neighborhood' to identify deviations from normal patterns. In essence, k-NN operates by locating the nearest k neighbors to a given data point and computing an anomaly score based on the distance to these neighbors. This approach is particularly effective in datasets where normal instances predominate, with the anomaly score reflecting the average distance to the k nearest neighbors. Such a mechanism is adept at quantifying the extent of a data point deviation from its closest counterparts, thereby offering a robust metric for anomaly detection. Despite its intuitive nature and the ease of parameter adjustment, k-NN may encounter limitations when dealing with high-dimensional data or exceedingly large datasets due to computational constraints.

Mahalanobis distance This emerges as a sophisticated metric capable of gauging the similarity between an unknown sample and a known group, factoring in the correlation among variables. This metric is instrumental in identifying anomalies, with a larger Mahalanobis distance indicating a significant divergence from the known sample group. The process entails calculating a mean vector and covariance matrix from the training data, followed by computing the anomaly score for test data based on these parameters.

Gaussian Mixture Model (GMM) GMM introduces a probabilistic approach, assuming the data originates from a blend of several Gaussian distributions. It identifies the specific distribution each sample likely belongs to, facilitating anomaly detection through the estimation of normal data distribution. This estimation is achieved by maximizing the likelihood's expected value via the EM algorithm, with anomalies pinpointed through their low likelihood of belonging to this distribution.

Logistic regression It is a supervised learning method used for binary classification. It extends its utility to multiclass scenarios through the softmax function. It calculates the probability of each class based on training data, employing a cross-entropy loss function for optimization. This model is adept at class prediction, assigning the class with the highest probability as the predicted outcome.
SVM stands out for its ability to delineate the optimal decision boundary, maximizing the margin between classes in the feature space. Applicable to both linear and nonlinear classification challenges, SVM's adaptability is further enhanced through the kernel trick for nonlinear data. Employing the One-vs-All method for multiclass classification, SVM constructs individual models for each class, treating the classification of each as a binary problem. Despite its non-probabilistic nature, Platt Scaling can be applied to SVM outputs, converting decision function outcomes into probabilities, thereby facilitating evaluation through AUC metrics.

4.2. Diagnostics Scenarios

After the preliminary steps of data augmentation and feature extraction, the diagnostic process unfolds through six distinct scenarios, aimed at determining whether the data indicates normal conditions, an unknown anomaly, a Solenoid Valve (SV) fault, a bubble anomaly, or a combination of SV fault and bubble anomaly. Moreover, for SV faults, bubble anomalies, and their combinations, the process seeks to pinpoint the specific locations of these occurrences.

4.2.1. Scenario 1: Known Anomaly Detection

The first scenario is centered on differentiating normal from abnormal data through the application of machine learning models, thereby filtering out the normal data. Unsupervised learning models, such as Principal Component Analysis (PCA), One-Class Support Vector Machine (OC-SVM), Mahalanobis distance, and Gaussian Mixture Model (GMM) are utilized and evaluated against each other. The choice for unsupervised models stems from the need to account for unknown anomalies in the test data, which could be inaccurately classified into known anomaly categories by a straightforward classification model. These models are trained exclusively on normal data to define the "normal space." When new data is introduced, the model assesses how significantly it deviates from the normal data, assigning an anomaly score. A threshold for anomaly detection is established based on the anomaly scores from the training phase, with data surpassing this threshold labeled as abnormal. This threshold is set using the 3-sigma rule, which involves adding three standard deviations to the mean of the anomaly scores. Given the presence of unknown anomalies, hyperparameter optimization is bypassed in favor of arbitrary parameter selection.

4.2.2. Scenario 2: Unknown Anomaly Detection

Scenario 2 aims to identify and segregating unknown anomaly data that does not fall within the scope of known anomalies covered in the training data. This scenario proceeds under the assumption that all normal data have been eliminated in scenario 1, employing the same models. While the threshold and feature optimization techniques mirror those of scenario 1, the distinction lies in using known anomaly data for training. This data is considered "normal" for the purpose of learning its characteristics, with the model evaluating the deviation of new data from this learned space to compute an anomaly score.

4.2.3. Scenario 3: Categorization of SV Faults, Bubble Anomalies, and Their Combinations

The third scenario categorizes the data into three groups: SV fault, bubble anomaly, or a combination of both. This is achieved using supervised learning models, such as logistic regression and SVM, with the assumption that both normal and unknown anomaly data have been previously removed. To mitigate overfitting and evaluate model performance on new data, the training data is split into smaller training and evaluation datasets. Hyperparameters, including the learning rate and regularization coefficient, are fine-tuned using the evaluation dataset, with Optuna employed for hyperparameter optimization.

4.2.4. Scenario 4: Classification of SV Fault Locations

Scenario 4 focuses on classifying the specific locations of SV faults among SV1-SV4. It is predicated on the removal of all data except for SV fault data through the initial three scenarios. Employing the same models as in scenario 3, this scenario treats different degrees of openness (0%, 50%) as indicative of the same fault location, framing it as a 4-class multiclass classification challenge.

4.2.5. Scenario 5: Classification of Bubble Anomaly Locations

This scenario is dedicated to pinpointing the locations of bubble anomalies among BP2, BP17, and BP32. Following the exclusion of all data except for bubble anomaly data, as in previous scenarios, this scenario is structured as a 3-class multiclass classification problem, utilizing the methodologies and models from Scenarios 3 and 4.

4.2.6. Scenario 6: Classification of SV Faults + Bubble Anomalies Locations

The final scenario undertakes the classification of combinations of bubble anomalies and SV faults, encompassing 12 possible scenarios across BP2, BP17, BP32, and SV1-SV4. This scenario operates under the assumption that all data, except for the combined SV fault and bubble anomaly data, has been filtered out. It addresses this 12-class multiclass classification problem using the same models and approaches as outlined in scenarios 3, 4, and 5.

4.3. Evaluation Metrics

To gauge the effectiveness of each machine learning model deployed in this study, two widely recognized metrics, the F1 Score and the Area Under the Curve (AUC) value, are employed. These metrics are pivotal for a holistic evaluation of a model's predictive accuracy. The F1 Score serves as a balanced measure of a model's precision and recall, calculated as the harmonic mean of these two metrics. Precision quantifies the accuracy of positive predictions made by the model, whereas recall measures the model's ability to identify actual positives. The F1 Score, therefore, offers a comprehensive metric that encapsulates both precision and recall, making it particularly valuable in scenarios with class imbalances. The formulae for computing the F1 Score, Precision, and Recall are as follows:

$$F1\ Score = \frac{2.precision.recall}{precision + recall} \tag{2}$$

$$Precision = \frac{true\ positive}{true\ postive + false\ positive} \tag{3}$$

$$Recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{4}$$

A maximum F1 Score of 1 indicates exemplary model performance. In multi-class classification scenarios, where simple F1 Scores or AUC values are insufficient, Macro averaging is utilized. This technique averages the metrics computed for each class, ensuring equal contribution from all classes and mitigating the effects of class imbalance. The Macro F1 Score, derived from averaging the F1 Scores of individual classes, effectively addresses the challenges posed by classes with less data samples and provides insights into the model's performance variance across different classes.

It should be noted that while the F1 Score provides a balanced measure of precision and recall, in the context of propulsion system autonomy, recall is of greater importance due to the critical need for fault accommodation. In practice, the system's many redundancies make it more crucial to minimize missed detections rather than false positives. However, to show a detailed example we will use F1 score, where precision is equally important as recall.

4.4. Sequential Feature Selection (SFS) for Sensor and Feature Optimization

Sequential Feature Selection (SFS), a component of the wrapper method in machine learning, employs a greedy search algorithm to identify the optimal feature combination, reducing the feature space from a higher dimension d to a lower dimension k (< k<d). This process aims to eliminate redundant features and noise, thereby potentially enhancing computational efficiency and reducing errors in non-regularized algorithms. The study employs Sequential Forward Selection due to the computational challenges associated with backward selection in large feature sets. This method has proven effective in various applications, including cancer image classification and UAV image diagnostics.

The optimization process unfolds in two stages:

- Sensor Location Optimization: Utilizing the average F1 score across all scenarios as the objective function, SFS selects the best sensor location from among those equipped with 11 statistical and 6 frequency domain features.
- Feature Combination Optimization: With the optimal sensor location determined, SFS is then applied to each diagnostic scenarios to identify the most effective feature combinations, aiming for the highest F1 scores.

This two-step approach ensures the selection of the most impactful sensor locations and feature combinations for each diagnostic scenario.

4.5. Computational Environment

The feasibility of executing the proposed method onboard spacecraft is assessed using a Raspberry Pi Zero W, a microcomputer with a proven track record in space missions, including CubeSat deployments. This device, chosen for its compatibility with Python and its lightweight computational capabilities, features a 1GHz single-core ARMv6 CPU and 512MB RAM, operating on a 32-bit OS. The Raspberry Pi Zero W's specifications underscore its suitability for conducting lightweight computational tasks in space applications, offering a practical platform for implementing data-driven fault diagnosis in numerical spacecraft propulsion systems.

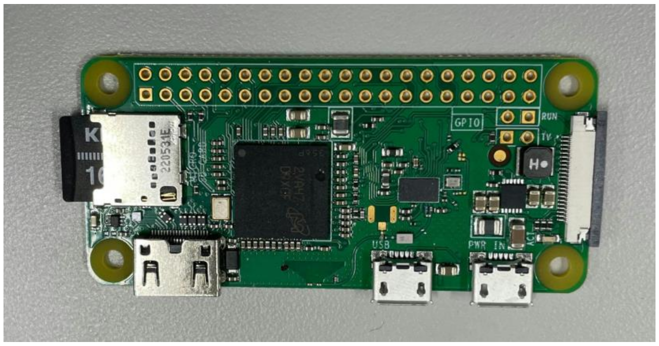


Figure 5: Appearance of the Raspberry Pi

5. RESULTS AND DISCUSSION

5.1. Sensor Optimization

This section explores the application of the proposed framework to a dataset, focusing on optimizing sensor and feature selection, and comparing the performance in terms of accuracy and processing time on the Raspberry Pi Zero W. The framework's effectiveness is assessed through a structured approach that starts with sensor optimization and concludes with a detailed analysis of computational efficiency.

Sensor optimization was conducted using Sequential Forward Selection (SFS), with the average F1 score serving as the primary metric. This optimization excluded scenario 2 from validation due to its specific focus on unknown anomalies, which would require a different approach for accurate assessment. Figure 6 illustrates the optimization process, where different sensor positions were evaluated for their contribution to the F1 score. Initially, P9 emerged as the optimal sensor choice, providing the highest F1 score when considered alone. This selection was based on its ability to capture critical information with minimal redundancy. As the optimization progressed, the process evaluated additional sensors that, when combined with P9, further improved the F1 score. The second sensor selected was P15, which, although initially not as prominent as P3 in detecting water hammer effects, offered complementary information to P9, making it the next best choice. The sequence of sensor selection continued, with each step aiming to maximize the F1 score by adding the most beneficial sensor to the existing set. This method led to a combination that peaked with an F1 score near 0.8. The process continued to add sensors, but after reaching a total of eight sensors, the F1 score began to stabilize.

Interestingly, the addition of extra sensors beyond five sensors marked a turning point in the optimization process. Beyond this point, the accuracy starts to decline, suggesting that adding more sensors might introduce duplicated information or data that does not contribute meaningfully to the model. This pattern indicates that the effectiveness of additional sensors diminishes after a certain point, likely due to the redundancy of data or the limitations of the sensor array. Consequently, the study concluded that a minimalistic approach, favoring the use of P9 exclusively, strikes the best balance between accuracy and computational efficiency. By carefully selecting only the most informative sensors, the framework ensures that high accuracy is maintained without unnecessary computational overhead, particularly important when deploying the model on resource-constrained devices like the Raspberry Pi Zero W.

It is important to clarify that the order of sensors on the X-axis in Figure 6 reflects the sequence in which sensors were selected during the optimization process to maximize the F1 score, rather than the individual peak F1 score of each sensor. For instance, while P6 might achieve the highest F1 score when considered alone, the optimization process prioritized combinations of sensors that collectively provided the best performance. Thus, P9 was selected first because it contributed most effectively to the combined score when starting the selection process, even though P6 may have shown a higher individual score later in the process. This approach underscores the importance of considering sensor combinations rather than just individual sensors, as the goal is to build a model that performs optimally with the selected set of sensors, rather than focusing solely on the highest-scoring individual sensor.

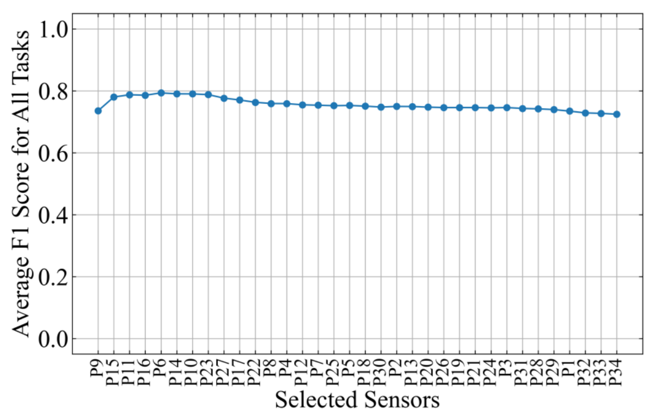


Figure 6: The result of sensor optimization by SFS

To conclude, the order of the sensors on the X-axis in Figure 6 represents the sequence in which sensor combinations were selected during the SFS process to maximize the F1 score. While P6 achieves the highest F1 score individually, P9 was chosen first because it provided the best initial contribution to the combined score

5.2. Feature Optimization Results

Following sensor optimization, feature selection through SFS focuses on identifying the most impactful features for each diagnostic scenario, excluding scenario 2 for consistency. Figure 7 illustrates the optimization journey across scenario, with scenario 1 peaking at five features before experiencing a decline in accuracy. In contrast, scenarios 4, 5, and 6 demonstrate a positive correlation between the number of features and accuracy, with scenario 5 achieving a perfect macro F1 score of 14 features. These trends underscore the nuanced impact of feature selection on model performance, highlighting the potential for overfitting or underfitting based on the feature set's complexity.

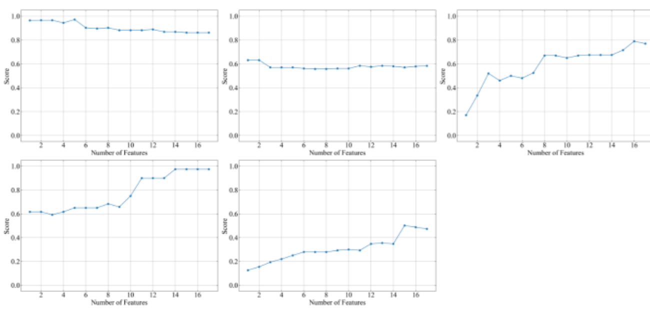


Figure 7: The result of feature optimization by SFS (from top to bottom: scenarios 1,3-6)

5.3. Comparative Analysis on Raspberry Pi Zero W

The evaluation on the Raspberry Pi Zero W, detailed in Figure 8, contrasts the performance of all sensors and features against the streamlined approach of the proposed framework. Notably, scenarios 1 and 5 exhibit improved accuracy with the optimized framework, whereas scenarios 3, 4, and 6 see a reduction. ... the model's ability to capture complex patterns in specific assignments.

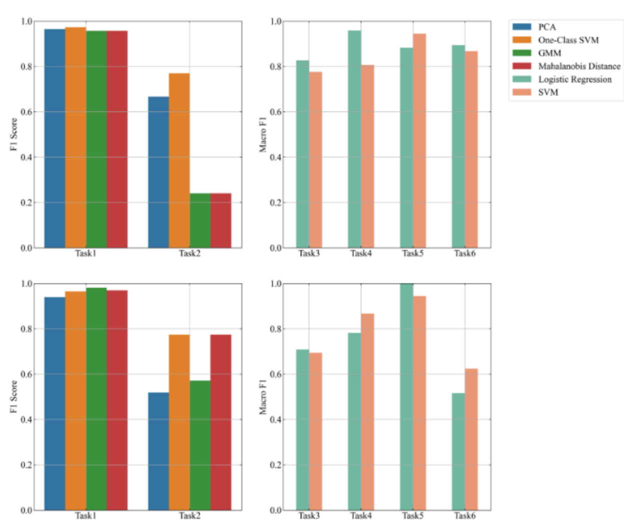


Figure 8: Comparison of F1 scores between using all sensors and features, and the proposed method (above: all sensors and features, below: proposed method)

5.4. Computational Efficiency

The processing times outlined in Table 4 emphasize the significant reduction in computational load achieved through the proposed framework. By narrowing the focus to a single sensor and a curated set of features, the framework not only lowers spacecraft costs but also accelerates processing times by approximately 15-fold. This efficiency gain primarily affects feature extraction and standardization, crucial for onboard spacecraft execution given the typical 1Hz down sampling rate. However, the critical analysis reveals a nuanced landscape.

While the proposed framework promises substantial benefits in terms of efficiency and cost, the observed decrease in accuracy for specific assignments raises important considerations. The balance between computational speed and diagnostic precision becomes a pivotal factor, especially in the context of space missions, where both are paramount. Moreover, the reliance on simulated data for optimization underscores the need for further validation with actual spacecraft data to ensure the framework's robustness and reliability in operational settings.

Table 4: Comparison of the total execution time and each scenario's execution time between using all sensors and features, and the proposed method

Scenario	All Sensors, All Features [ms]	Proposed Framework [ms]
Feature Extraction	9983	525
Standardization (Total of each Scenario)	850	182
Scenario 1	58.0	6.26
Scenario 2	57.7	6.45
Scenario 3	8.63	3.09
Scenario 4	4.32	2.83
Scenario 5	3.89	2.88
Scenario 6	7.14	3.20
Total	10972	731

6. CONCLUSION

This study presents a forward-looking approach aimed at enhancing the autonomy of spacecraft by introducing an efficient diagnostic framework specifically designed for spacecraft propulsion systems. Incorporating real-world data, conducting a detailed trade-off analysis, broadening the literature review, and focusing on handling unknown anomalies will significantly enhance the framework's practical utility and robustness. Leveraging data derived from simulations, this method seeks to streamline the diagnostic process by minimizing the number of sensors required on the spacecraft. It achieves this by optimizing sensor placement and feature selection, using Sequential Forward Selection (SFS) to greatly reduce the system's execution time. When implemented on a microcomputer with a proven track record in spacecraft operations, the framework demonstrated a substantial reduction in execution time—approximately 15-fold—compared to traditional methods utilizing all available sensors and features. This efficiency gain notably surpasses the standard downsampling rate of spacecraft, albeit at the expense of a slight decrease in overall accuracy.

6.1. Future Work

Looking ahead, this study has identified several key areas for further development to enhance the framework's effectiveness and applicability in real-world spacecraft operations.

Improving accuracy: By enhancing the accuracy of the proposed method, we can incorporate a greater number of sensors and features. This will provide more detailed data, improving diagnostic precision.

Addressing data discrepancies: The issue is with the differences between simulated data and actual spacecraft data. A possible solution is to quantify and address any discrepancies to ensure the framework accurately reflects real-world conditions.

Expanding diagnostic scope: The framework needs to support autonomous reconfiguration, control, and decision-making based on diagnostic outcomes. This will help go beyond just fault identification. As a result, future work should investigate other advanced machine learning techniques, such as Deep Learning and Reinforcement Learning to adapt and accurately identify novel fault types.

Handling unknown anomalies: Another challenge is the with detecting and managing unforeseen anomalies. This warrants investigation into ensemble learning methods to combine multiple models, leveraging their collective strengths to create a more robust and resilient diagnostic system.

Developing a comprehensive solution: We not only want to diagnose faults accurately, but also enable automatic system adjustments and decision-making processes. It is important to take a more holistic approach for real-world spacecraft operations, where the ability to autonomously adapt and respond to diagnostic findings is essential for mission success.

Hydrazine valve operation: Operating hydrazine valves more slowly could extend their lifespan by reducing mechanical stress and wear. However, slower operations might obscure prognostic signals, making early fault detection more challenging. As a result, further investigation is needed to balance these trade-offs. Integrating additional sensing technologies or advanced data analytics techniques may help capture subtle anomalies even at slower operation speeds.

Cost-Benefit analysis of sensor and feature optimization: Additional sensors and features come with costs such as increased weight, space, and computational load. It is useful to conduct a cost-benefit analysis to balance improved diagnostic capability against the added resource requirements. This ensures a practical and feasible diagnostic framework.

By addressing these challenges and areas for improvement, future research will significantly advance the field of spacecraft autonomy. This will pave the way for more resilient and self-sufficient space exploration missions, ensuring mission success through enhanced diagnostic and adaptive capabilities.

REFERENCES

[1] M. Smith et al., “The Artemis Program: An Overview of NASA's Activities to Return Humans to the Moon”, in 2020 IEEE Aerospace Conference, 2020, pp. 1–10.

[2] M. K. Biswal M and R. N. Annavarapu, “Human Mars Exploration and Expedition Challenges”, in AIAA Scitech 2021 Forum, 2021, p. 628.

[3] C. Caini, R. Firrincieli, T. de Cola, I. Bisio, M. Cello, and G. Acar, “Mars to Earth communications through orbiters: Delay Tolerant/Disruption–Tolerant Networking performance analysis”, International Journal of Satellite Communications and Networking, vol. 32, no. 2, pp. 127–140, 2014.

[4] S. Khan and T. Yairi, “A review on the application of deep learning in system health management”, Mechanical Systems and Signal Processing, vol. 107, pp. 241–265, 2018.

[5] K. O. Kolcio, “Model–based fault detection and isolation system for increased autonomy”, AIAA SPACE 2016. p. 5225, 2016.

[6] K. Kolcio and L. Fesq, “Model–based off–nominal state isolation and detection system for autonomous fault management”, in 2016 IEEE Aerospace Conference, 2016, pp. 1–13. doi:10.1109/AERO.2016.7500793.

[7] K. Kolcio, L. Fesq, and R. Mackey, “Model–based approach to rover health assessment for increased productivity”, in 2017 IEEE Aerospace Conference, 2017, pp. 1–13. doi: 10.1109/AERO.2017.7943835.

[8] A. Nikora, P. Srivastava, L. Fesq, S. Chung, and K. Kolcio, “Assurance of model–based fault diagnosis”, in 2018 IEEE Aerospace Conference, 2018, pp. 1–14. doi: 10.1109/AERO. 2018.8396550.

[9] G. B. Aaseng, “Scaling up model–based diagnostic and fault effects reasoning for spacecraft”, in AIAA SPACE 2015 Conference and Exposition, 2015, p. 4465.

[10] D. Henry, A. A. Barahona, N. Ellero, V. Dubanchet, P. Colmenarejo, and F. Ankersen, "Model–based fault diagnosis and fault–tolerant data fusion algorithms for the ESA's In–Orbit–Assembly project", in 2021 5th International Conference on Control and Fault–Tolerant Systems (SysTol), 2021, pp. 211–218. doi: 10.1109/SysTol52990.2021.9595822.

[11] D. Henry, C. Le Peuvédic, L. Strippoli, and F. Ankersen, "Model–based FDIR and Fault Accommodation for a Ren– dezvous Mission around the Mars Planet: the Mars Sample Return Case", IFAC–PapersOnLine, vol. 49, no. 5, pp. 266– 271, 2016, doi: https://doi.org/10.1016/j.ifacol.2016.07.124.

[12] D. Henry et al., "Model–based fault diagnosis and tolerant control: the ESA's e.Deorbit mission", in 2019 18th Euro– pean Control Conference (ECC), 2019, pp. 4356–4361. doi: 10.23919/ECC.2019.8796282.

[13] R. Fonod, D. Henry, C. Charbonnel, and E. Bomschlegl, "A class of nonlinear unknown input observer for fault diagnosis: Application to fault tolerant control of an autonomous space– craft", in 2014 UKACC International Conference on Con– trol (CONTROL), 2014, pp. 13–18. doi: 10.1109/CONTROL. 2014.6915108.

[14] Y. Gao, T. Yang, N. Xing, and M. Xu, "Fault detection and diagnosis for spacecraft using principal component analysis and support vector machines", in 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2012, pp. 1984–1988. doi: 10.1109/ICIEA.2012.6361054.

[15] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis", Chemometrics and Intelligent Laboratory Systems, vol. 2, no. 1–3, pp. 37–52, 1987.

[16] C. Cortes and V. Vapnik, "Support–vector networks", Machine learning, vol. 20, pp. 273–297, 1995.

[17] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding", in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, in KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, p. 387. doi: 10.1145/3219819.3219845.

[18] S. Hochreiter and J. Schmidhuber, "Long Short–term Memory", Neural computation, vol. 9, pp. 1735–80, 1997, doi: 10.1162/neco.1997.9.8.1735

[19] I. Gueddi, O. Nasri, K. Benothman, and P. Dague, "Fault detection and isolation of spacecraft thrusters using an extended principal component analysis to interval data", International Journal of Control, Automation and Systems, vol. 15, pp. 776– 789, 2017.

[20] J. R. Mansell and D. A. Spencer, "Deep learning fault diagno– sis for spacecraft attitude determination and control", Journal of Aerospace Information Systems, vol. 18, no. 3, pp. 102– 115, 2021.

[21] B. Xiao and S. Yin, "A Deep Learning Based Data–Driven Thruster Fault Diagnosis Approach for Satellite Attitude Con– trol System", IEEE Transactions on Industrial Electronics, vol. 68, no. 10, pp. 10162–10170, 2021, doi: 10.1109/TIE. 2020.3026272.

[22] M. Yamamoto et al., "Surge Pressure Management in HTV Propulsion System", in 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2007.

[23] B. Brindle, J. Gilbert, J. Moore, and G. Risha, "Effects of Water Hammer on Propulsion Systems", 2016, p. [24] E. B. Wylie, V. L. Streeter, and L. Suo, Fluid transients in systerms, vol. 1. Prentice Hall Englewood Cliffs, NJ, 1993.

[25] A. Kodura, "An analysis of the impact of valve closure time on the course of water hammer", Archives of Hydro–Engineering and Environmental Mechanics, vol. 63, no. 1, 2016.

[26] A. Kumar, P. S. Prasad, and M. R. Rao, "Experimental studies of water hammer in propellant feed system of reaction control system", Propulsion and Power Research, vol. 7, no. 1, pp. 52–59, 2018.

[27] R. Lecourt and J. Steelant, "Experimental investigation of wa– terhammer in simplified feed lines of satellite propulsion systems", Journal of Propulsion and Power, vol. 23, no. 6, pp. 1214–1224, 2007.

[28] K. Kawatsu et al., "Resilient redundant spacecraft GN&C sys– tem fault detection and diagnostics", in Aerospace Europe Conference, AEC2, 2020.

[29] K. Kawatsu, C. Inoue, Y. Terauchi, Y. Daimon, G. Fujii, and A. Noumi, "System–level Integrated Modeling and Simulation Targeting Space Propulsion System for Design Evaluation and Risk Assessment", in 32nd International Symposium on Space Technology and Science, 2019.

[30] G. Fujii et al., "Verification of the Transient Behavior of the SLIM Propulsion System (Smart Lander for Investigating Moon)", in AIAA Propulsion and Energy 2021 Forum, 2021, p. 3582.

[31] H. YAMAMOTO, K. KAWATSU, Y. DAIMON, and G. FUJII, "Development of a Dynamic Response Model Considering Trapped Gas Effect for Spacecraft Liquid Propulsion Systems", Journal of Evolving Space Activities, vol. 1, no., p. 34, 2023, doi: 10.57350/jesa.34.

[32] K. KAWATSU, Y.–i. IZATO, and A. MIYAKE, "Framework for Model–Based Approach to Life Cycle Risk Management for Spacecraft Liquid Propulsion System", Journal of Evolving Space Activities, vol. 1, p. 78, 2023.

[33] K. Kawatsu, G. Fujii, T. Nagata, H. Yamamoto, and Y. Dai– mon, "Preliminary Demonstration of Concept for Fault Diag– nosis in Resilient Redundant Spacecraft Propulsion", in AIAA SCITECH 2023 Forum, 2023, p. 1259.

[34] "Modelica Language Specification".

[35] "SimulationX".

[36] K. Tominaga et al., “Dataset Generation Based on 1D–CAE Modeling for Fault Diagnostics in a Spacecraft Propulsion Sys– tem”, in PHM Society Asia–Pacific Conference, 2023.
[37] T. Minami and J. Lee, “PHM for Spacecraft Propulsion Sys– tems: Similarity–Based Model and Physics–Inspired Features”, in PHM Society Asia–Pacific Conference, 2023.
[38] S. K. Lee et al., “Hybrid Approach of XGBoost and Rule–based Model for Fault Detection and Severity Estimation in Space– craft Propulsion System”, in PHM Society Asia–Pacific Confer– ence, 2023.
[39] Y. Kato, T. Kato, and T. Tanaka, “Anomaly Detection in Space– craft Propulsion System using Time Series Classification based on K–NN”, in PHM Society Asia–Pacific Conference, 2023.
[40] B. Schölkopf, J. C. Platt, J. Shawe–Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high–dimensional distribution”, Neural computation, vol. 13, no. 7, pp. 1443– 1471, 2001.
[41] E. Fix and J. L. Hodges, “Discriminatory analysis. Nonparametric discrimination: Consistency properties”, International Statistical Review/Revue Internationale de Statistique, vol. 57, no. 3, pp. 238–247, 1989.
[42] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets”, in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 427–438.
[43] P. C. Mahalanobis, “On the generalized distance in statistics”, Sankhyā: The Indian Journal of Statistics, Series A (2008–), vol. 80, p. S1—S7, 2018.
[44] C. M. Bishop and N. M. Nasrabadi, Pattern recognition and machine learning, vol. 4, no. 4. Springer, 2006.
[45] B. Awwad Shiekh Hasan and J. Q. Gan, “Sequential EM for unsupervised adaptive Gaussian mixture model based classifier”, in International Workshop on Machine Learning and Data Mining in Pattern Recognition, 2009, pp. 96–106.
[46] P. Karsmakers, K. Pelckmans, and J. A. K. Suykens, “Multi– class kernel logistic regression: a fixed–size implementation”, in 2007 International Joint Conference on Neural Networks, 2007, pp. 1756–1761. doi: 10.1109/IJCNN.2007.4371223.
[47] A. Widodo and B.–S. Yang, “Support vector machine in machine condition monitoring and fault diagnosis”, Mechanical systems and signal processing, vol. 21, no. 6, pp. 2560–2574, 2007.
[48] J. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”, Adv. Large Margin Classif., vol. 10, p., 2000.
[49] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection”, Journal of machine learning research, vol. 3, no. Mar, pp. 1157–1182, 2003.
[50] S. Bouatmane, M. A. Roula, A. Bouridane, and S. Al–Maadeed, “Round–Robin sequential forward selection algorithm for prostate cancer classification and diagnosis using multispec– tral imagery”, Machine Vision and Applications, vol. 22, pp. 865–878, 2011.
[51] S. Shafiee, L. M. Lied, I. Burud, J. A. Dieseth, M. Alsheikh, and M. Lillemo, “Sequential forward selection and support vec– tor regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery”, Computers and Electronics in Agriculture, vol. 183, p. 106036, 2021, doi: https://doi.org/10.1016/j.compag.2021.106036.
[52] Raspberry Pi Foundation, “Raspberry Pi Zero W”. 2023.
[53] J. Danos, C. Page, S. Jones, and B. Lewis, “USU’s GASPACS CubeSat”, 2022.