Modular AI for Faults: Local Watch and Efficient Response

Richard Stottler, Evan Finnigan, Sowmya Ramachandran,
Abhimanyu Singhal, Christoper Healy
Stottler Henke Associates, Inc.
1650 S Amphlett Blvd # 300
San Mateo, CA 94402
stottler, efinnigan, sowmya, asinghal, chealy @stottlerhenke.com

Abstract—**As NASA continues to develop and launch satellites, rovers, landers, and other spacecraft, there is a growing need to increase the level of autonomy of each system. As the number of spacecraft increases, it becomes increasingly difficult to actively monitor each of them for off-nominal behavior, even with the help of a large staff. Furthermore, time-sensitive faults might require a faster response time than what human operators or mission-control can provide. Therefore, one of the key capabilities for an autonomous system is the ability to manage faults and off-nominal behavior. The paper will describe a new framework called MAIFLOWER (Modular AI for Faults: Local Watch and Efficient Response) designed to provide generalizable, automatic anomaly detection, diagnosis and recovery capabilities. MAIFLOWER offers several key benefits. First, MAIFLOWER provides a robust fault detection interface to quickly, accurately, and autonomously identify anomalous behavior from onboard telemetry data. Second, MAIFLOWER integrates several anomaly detection methods from knowledge-based approaches such as model-based reasoning (MBR) to data-driven methods like machine learning (ML) and thermodynamic analysis into a single package, leveraging strengths of each method while mitigating their weaknesses. Third, the fault management modules of MAIFLOWER are designed to cue adaptive execution modules as well as trigger replanning and rescheduling procedures to appropriately respond to faults, creating a closed loop execution environment for continuous anomaly detection and fault management. The paper also describes the application of MAIFLOWER to Astrobotic's Vertical Solar Array Technology Optimized for Lunar Traverse (VOLT), a mobile power generation and transmission platform being developed as part of Astrobotic's lunar power grid architecture.**

TABLE OF CONTENTS

1. INTRODUCTION

Monitoring and diagnosing off-nominal behavior for autonomous space systems is a very critical, but challenging problem. The majority of autonomous systems in space will be unmanned and, as our explorations take us further from the Earth, the latencies involved in human interventions when faults occur will make this an infeasible option. Furthermore, anomalies can render systems non-operational in a matter of seconds, making it critically important to detect and diagnose them very rapidly. Therefore, autonomous systems require AI based off-nominal behavior detection and diagnosis systems and the ability to develop and execute recovery plans autonomously without the need to communicate with ground controllers [1]. This paper describes the Modular AI for Faults: Local Watch and Efficient Response (MAIFLOWER) systems which provides these capabilities.

Model-based reasoning (MBR) and data-driven or machine learning methods are typically used for anomaly detection and diagnosis (REF). Each has associated costs and benefits. Unsupervised learning approaches are effective at detecting previously unknown or unforeseen anomalies. Furthermore, unsupervised approaches like clustering can automatically model patterns of nominal behavior from unlabeled data and therefore can be used to rapidly develop system models using data generated either from the system itself or a simulation or a digital twin. Machine learning approaches, however, cannot effectively trace the root causes for an anomaly for a variety of reasons: 1. Not having the information (i.e., labels) needed to discriminate between faulty and nominal behavior; and 2. lacking insight into un-instrumented system components. The latter is an important point: not every component of a system will be instrumented to provide telemetry, and thus the machine learning system will not have visibility into significant portions of the system. MBR approaches on the other hand are effective at detecting and diagnosing faults

979-8-3503-5597-0/25/$31.00 ©2025 IEEE

that are known but require upfront model development. Furthermore, they may not detect unforeseen anomalies not captured in the model.

Combining MBR with machine learning can offer the benefits of both methods while mitigating some of their shortcomings. MAIFLOWER is a framework for doing this. It combines model-based reasoning (MBR) and a novel data-driven approach called Thermodynamic Reasoning Intelligent Anomaly Detection (TRIAD) [2]. TRIAD combines analysis in time and frequency domains for fast anomaly detection. TRIAD is designed to incorporate additional data-driven anomaly detection methods such as deep neural network-based methods, thus making it a flexible and extensible framework. In addition, TRIAD has several key strengths over other unsupervised learning techniques: first, it can dynamically adjust to what is necessary for the scenario being monitored. Second, TRIAD can be used to intelligently aggregate other anomaly detection algorithms (as opposed to running them in parallel). These upsides dovetail, in that a number of state-of-the-art techniques for anomaly detection can be bundled into a computationally-extensive instantiation of TRIAD, and then whittled down to provide optimal performance for a given compute budget. For a sparse compute budget, TRIAD can provide high performance without incorporating any additional methods.

We applied MAIFLOWER to Astrobotic Technology's **Vertical Solar Array Technology Optimized for Lunar Traverse** (VOLT), a mobile power generation and transmission platform being developed under several NASA contracts as a core component of Astrobotic's lunarpower grid architecture (LunaGrid) for the lunar south pole. The VOLT carries a 20-meter tall photovoltaic solar array which generates 10kW of power to deliver to other lunar systems via wired and wireless connections. In its nominal operations, VOLT will egress from its lander, transit to a desired location near peaks of persistent light at the lunar south pole, settle into the lunar soil by oscillating its wheels, and then deploy the Roll Out Solar Array (ROSA), developed by Redwire of Goleta, CA. Redwire has developed and successfully deployed ROSAs on the International Space Station (ISS), and has modified the technology to support lunar surface deployment. The ROSA is deployed by unrolling two composite booms. As the ROSA is unfurled, a suite of sensors including inertial measurement units (IMUs) and force sensors on the wheels continuously monitor the array's movement, measuring angular deflections on the gimbal and rover chassis. Figure 1 shows the deployed configuration of the VOLT and Figure 2 shows the stowed configuration. The VOLT's primary structural elements, including the booms, gimbal, and chassis, have been designed to keep the deployed array within a tight tolerance of +/- 3 degrees of the local gravity vector to maintain stability. However, the deflections in the system must be continuously monitored during deployment to detect potential stability issues and correct if present.

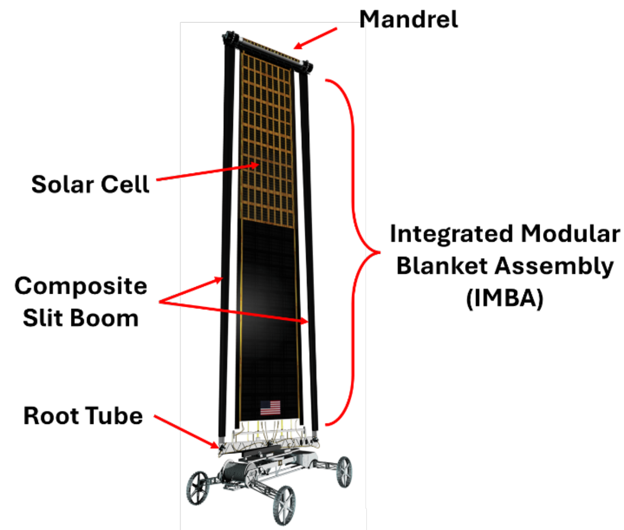

Figure 1. VOLT in deployed configuration[1]

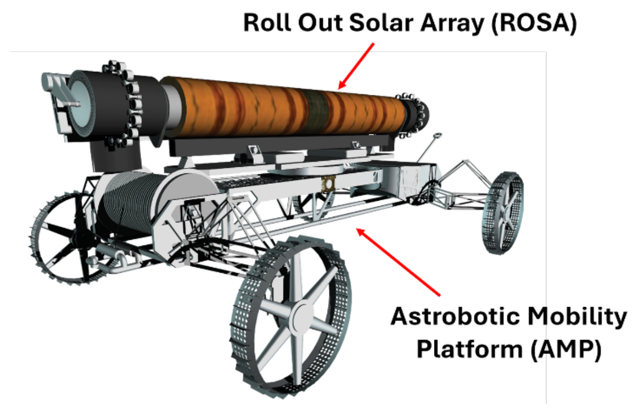

Figure 2. VOLT in stowed configuration[2]

One challenge during ROSA deployment is the local variation and instability of regolith, which are especially significant on terrain with a high slope. This is a relevant challenge, because the ROSA must be deployed on an illuminated peak, which will have a high slope of up to 15 degrees, to have the persistent light required to generate power. It is therefore crucial to detect faults such as vehicle instability due to regolith shift or collapse as soon as they start to give ample time for corrective actions. Accuracy is also an important consideration. While false negatives are widely acknowledged as undesirable, false positives can also lead to adverse outcomes for the VOLT. For example, trying to correct a VOLT's ROSA to a proper level, when it is in fact already within its stability tolerance, can itself lead to

---

[1] Image provided by Astrobotic

[2] Image provided by Astrobotic

undesirable loss of stability.

We will present the details of MAIFLOWER and demonstrate its accuracy and speed of detection on several simulated VOLT fault scenarios.

2. RELATED WORK

Traditionally, Fault Detection, Isolation, and Recovery (FDIR) systems have used Model Based Reasoning (MBR), which requires knowledge of the subsystem design and the behavior of components down to the desired level of diagnosis [3].

Modern developments in fault detection use data driven, Machine Learning (ML)-based, approaches. A unified approach to anomaly detection simplifies to an unsupervised learning task aimed at developing a valid model for the majority of the data points. Anomaly detection methods typically use unsupervised clustering to find high density regions, and data that does not fit well into these high-density regions are identified as anomalous. Anomaly detection differs from the classification task, because typically the training data only represents nominal operations, and the anomalies may often not be fully known in advance [4].

There are a number of ways to model nominal vs off-nominal data. Some of these are traditional AI approaches such as principal component analysis (PCA) [5], K-Means [6], and Gaussian Mixture Models (GMM) [7]. There are also one class classification methods such as support vector machines (SVM) [8], which classify nominal data into one class and everything else into an off-nominal class.

The modern methods for learning models of nominal data include Variational Autoencoders (VAE) [9] and Generative Adversarial Networks (GAN) [10]. These are generative approaches that use neural networks to learn to generate realistic output from noise and conditional inputs. The VAE or GAN learn a model of the distribution of expected system behavior from nominal system behavior data. Sensor readings that deviate from this model by more than a prespecified threshold are considered off-nominal or anomalous.

In order to successfully integrate with the VOLT system, there are a number of requirements that MAIFLOWER must be able to meet, including a need for quick reaction times in order to detect and correct leveling errors during ROSA deployment. MAIFLOWER also needs to be capable of detecting very gradual changes that are hard to discern over sensor noise. For example, the ROSA makes only one full rotation every month which it does in 9-degree increments which take 30 seconds each and occur every 8 hours. The traditional ML approaches described above cannot handle this range. TRIAD uses both the time and frequency domains to detect anomalies and therefore can handle both slow- and fast-moving behaviors.

There are a family of related technologies, created by Stottler Henke for anomaly detection using our Management of consumables Adaptive Execution, SynchronizaTion, Replanning/rescheduling, Optimization system (MAESTRO) framework [2]. The MAESTRO framework includes interfaces for diagnostic engines, adaptive execution systems, planners and schedulers. Versions of MAESTRO have been applied to the Lunar Gateway [2], the Cryogenic Test Bed at NASA's Goddard Space Flight Center and the next generation Exploration Extravehicular Mobility Unit (xEMU) space suit. MAIFLOWER adds to MAESTRO by integrating both MBR and TRIAD.

3. METHOD

VOLT Background

The VOLT is comprised of two major subsystems; a mobile base known as the Astrobotic Mobility Platform or AMP, and the Roll Out Solar Array, or ROSA. The ROSA is attached to the AMP via a 4 DOF gimbal, which provides leveling and sun-tracking capability for the array. The AMP is composed of a rigid chassis which contains avionics compute and power electronics enclosures, as well as the primary vehicle thermal system. Four spoked wheels are attached to the chassis via folding legs that are stowed for launch and deploy once the VOLT reaches the lunar surface. The front legs have additional degrees of freedom which enable steering and roll axis pivoting to traverse uneven terrain. The ROSA primarily consists of four elements: 1) a rigid root tube which provides the structural attach point to the gimbal and pointing actuator, 2) the integrated modular blanket assembly (IMBA) to which the solar cells are affixed, 3) the mandrel which is a cylindrical element used to spool and unspool the array, and 4) two composite slit booms which provide structural support to the array as it deploys. In its deployed configuration, the ROSA stands 20 meters tall, with 10kW of solar cells occupying the top half of the blanket.

There are a number of sensors on the VOLT that MAIFLOWER will use to detect and diagnose faults. The VOLT has three IMUs, one on the top of the ROSA, another on the top of the gimbal, and another on the bottom of the gimbal. There are 12 cameras, two on each side of the AMP to provide stereo vision for navigation and two upward facing cameras for monitoring ROSA deployment. There are four strain gauges to measure the loads on each of the wheels. Additionally, there are position and torque sensors on each of the four wheels, the pointing actuator, the steering motor, on each of the three arms of the gimbal and in the mandrel.

There are a wide range of faults that a general flight model (FM) architecture needs to be capable of detecting, diagnosing and reacting to. The most critical mechanical fault relevant for MAIFLOWER is the ROSA on VOLT exceeding its tolerance of +/- 3 degrees from the local gravity vector during deployment operations. Exceeding this requirement results with unsuccessful deployment of the ROSA to generate power, and in the worst case could lead to a dynamic instability causing the vehicle to tip due to off-nominal center of mass. During deployment, acceleration and velocity measurements must be constantly monitored to assess whether the ROSA is within its safe operating limits. Soil failure can also result in the ROSA exceeding its safe operating limits. Examples of soil failure include soil collapse and soil sliding. Both of these issues can happen after initial gimbal leveling and during deployment operations.

Sensor faults are a subset of mechanical faults, and sensor accuracy can be compromised by a variety of factors. All sensor outputs will contain some amount of noise (e.g. thermal noise detected by the sensor, electrical noise, or sensor drift); however, sensors can also be damaged by an impact, or by a shock caused by upstream electrical components. A faulty sensor can exhibit a range of behaviors, from producing random values to getting stuck at a single value or getting stuck at zero.

There are also a number of electrical power system (EPS) faults that can affect the 64 photovoltaic solar panel modules (SPMs) that make up the 10kW ROSA. Cells in the SPMs can experience physical damage due to radiation induced charge buildup or micrometeoroid debris, as well as relays faulting open or closed and sensor failures. Failure of one or more SPMs to generate power may necessitate real-time action to shed loads or immediately reschedule tasks using power in order to prevent over discharging the batteries or otherwise overtaxing the EPS. Excess energy generation can also be an issue. When solar panels are body-mounted, excess energy (which becomes heat) is a less of a concern because the rover body provides a ready heat sink. In the case of the ROSA, heat buildup is also not an issue because the ROSA will radiate heat into space through its back side.

MAIFLOWER also has an ability to monitor the voltage and current into and out of the batteries to intelligently manage their charging and discharging cycle. This battery management system is expected to extend battery life and to preserve maximum charge capacity. If the batteries are mistreated by being over, undercharged, or by being charged outside of the recommended temperature range, they will degrade more quickly. However, even well-treated batteries will eventually fail. In this context, a battery failure is when the battery has 80% or less of its original charge capacity. This degradation will be manifested by the observation that charging current appears to charge the battery more quickly than before. Another indication of a degraded battery is the bus voltage sagging more quickly [11].

The next subsection will describe the MAIFLOWER approach for handling these types of faults.

MAIFLOWER System

MAIFLOWER uses hybrid MBR and TRIAD, which emphasizes the benefits of each approach and mitigates the disadvantages. The benefits of the hybrid system include the following:

- detects and diagnoses anomalies never before encountered
- provides high accuracy on “Day One” of operations
- effectively utilizes existing design knowledge
- detects and diagnosis faults with a small amount of data
- produces detection and diagnosis results that are easy for humans to interpret
- provides a rigorously certifiable software implementation
- has predictable behavior
- diagnoses down to the lowest modelled component level
- handles rare but modeled operating conditions
- has a very quickly executing software implementation
- discovers unknown and subtle relationships (even across subsystems)
- provides a high degree of certainty in the diagnosis when both approaches agree

During normal operations, MAIFLOWER monitors onboard sensor values to characterize the systems. Characterization means automatically learning nominal behavior of subsystem components so that MAIFLOWER will be prepared to detect deviations from this nominal behavior which could be potential faults. As a system starts failing, MAIFLOWER will first detect the problem and immediately proceed to fail-safe state to minimize damage. Next, MAIFLOWER will diagnose the problem and determine the root cause. Finally, MAIFLOWER will plan actions to mitigate the problem.

Any intelligent, adaptive system must inherently be a closed loop system because it must sense what is occurring and make appropriate decisions to take suitable actions, and then sense the effects of those actions. The first part of this sense-decide-act loop involves perception or understanding the situation from raw sensor values.

MAIFLOWER uses hybrid MBR and TRIAD for this perception task. MBR encodes the schematic information of subsystems, which includes the components (including sensors themselves), their normal behavior and known abnormal modes of behavior, and the connections between components. During normal operations, the model is used to simulate the current behavior and compare the simulated sensor output values to the actual sensor outputs. Significant deviations are used to detect the existence of a fault, and then the model is used to determine which component faults are most likely to lead to observed pattern of deviation from the expected sensor values. The set of possible faults, including sensor faults which explain the unexpected sensor values, is the MBR diagnosis engine’s output. TRIAD is a model-free module to detect and diagnose faults. It intelligently synthesizes the feature functions of many time-series anomaly detection algorithms, including state-of-the-art methods like Convolutional Neural Networks (CNNs) and Transformers. TRIAD creates feature encodings based on all of the feature functions and then performs threshold-based detection to find off-nominal behavior. Our initial version of MAIFLOWER is specifically focused on anomaly detection, and diagnosis; automation for mitigating faults is planned in the future.

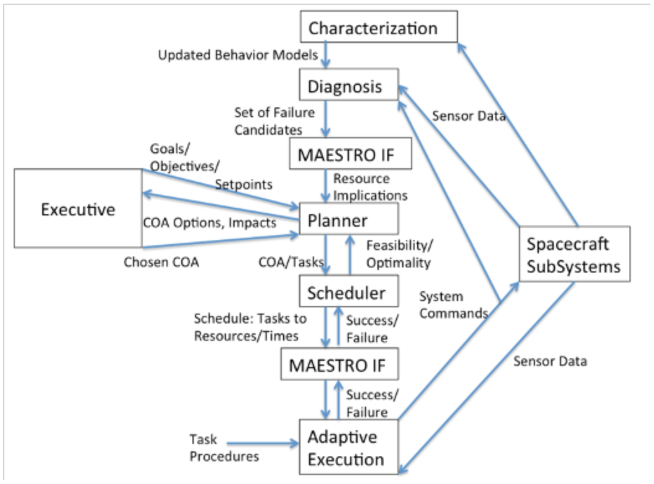

Figure 3. System Diagram of the MAIFLOWER Framework

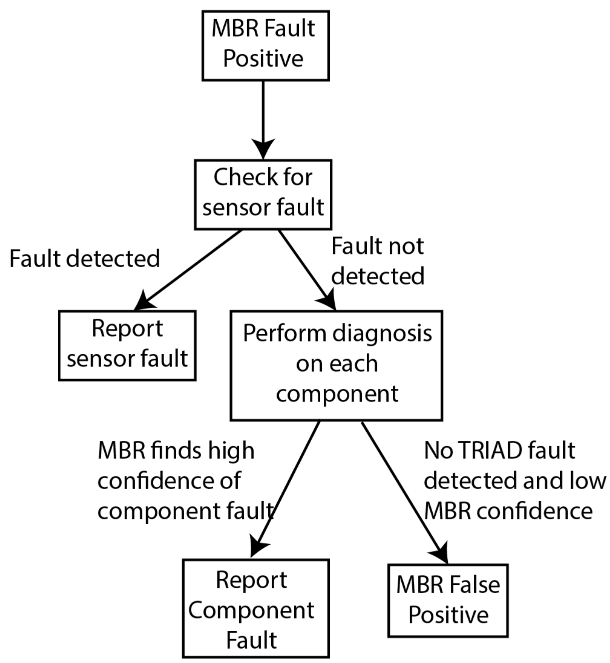

Figure 4. Decision tree showing how MAIFLOWER finds false positives and various faults

Figure 3 shows the high-level system design of MAIFLOWER. MAIFLOWER is a closed loop system that takes sensor data from the spacecraft subsystems and uses it both to diagnose anomalies and to characterize the system's nominal behavior. When MAIFLOWER detects and diagnoses a fault, it will first safe the spacecraft, i.e., it will implement an immediate action to prevent potentially mission ending damage. The planner and scheduler modules then add tasks to the spacecraft's active mission plan to mitigate faults and adjust the tasks already in the mission plan such that the subsystems affected by the fault are circumvented. The adaptive execution model sends commands to the spacecraft itself, which will run those commands, and produce for telemetry data, closing the loop. Note that the MAIFLOWER version described here focuses on the diagnosis block of this diagram.

Model-Based Reasoning

MBR based diagnosis systems encode detailed and explicit descriptions for the interrelated factors that affect sensor readings. These models typically represent the world as a collection of components, where each component is characterized by attributes and one or more possible modes of operation. The model also contains constraints that specify required relationships between attribute values and modes. Constraint violations are used to identify components in faulty modes. For example, forces, torques, translational and angular accelerations, velocities and positions are constrained by physics. If the sensor measurements of these attributes do not obey these relationships, either the sensors, or the physical component itself must be at fault. MBR models can represent nominal conditions as well as known fault modes. The nominal condition itself may need to model more than one mode of operation. In this case, the MBR's task will be to assess the system's current mode of operation based on sensor data and commands issued to the system.

Models encode the effects of contextual factors, so they can be applied reliably across contexts, such as the current environment, configuration, and sent commands. Model-based reasoning requires knowledge engineering efforts to encode these interacting effects. However, the knowledge engineering burden can be significantly reduced by automatically compiling existing schematic diagrams of systems. MBR engines can be extremely fast and do not require a large amount of memory or compute power, even for complex models.

MAIFLOWER encodes a low fidelity model of the VOLT for MBR to estimate predicted sensor measurement values, which can then be compared to the actual sensor values. Note that typically the models used for MBR are lower fidelity that the real system in order to manage the computational and modeling costs. Because the VOLT is expected to be dug into the lunar regolith and not moving, the MBR model is very simple. In this model, the expected sensor values are fixed and based on the static forces of the system. In other words, if the VOLT is on a flat surface, the force on each wheel is expected to be ¼ of the gravitational force acting on the vehicle.

The simple model of the VOLT used by MBR attempts to mimic the real system (or realistic simulation if MAIFLOWER needs to be tested without access to a physical system). MAIFLOWER receives sensor telemetry data from the real system which it will compare to the corresponding sensor values in its simplified MBR model of the VOLT. MAIFLOWER also issues commands identical to those issued to the real system to its simplified MBR model of the VOLT to ensure that the simplified MBR model is in the same operational mode as the real system. The MBR module calculates expected sensor values using its internal model while considering the phase of the mission. It compares the expected values to the actual values from telemetry and notes significant deviations. Cross checking is then performed for each anomalous sensor to look for sensor faults or errors. If such errors are detected, MBR reports this finding and completes the loop. If sensor errors are ruled out as the cause of anomalous telemetry, the MBR module steps through each component to check for major faults by comparing the expected behaviors for each known fault mode. If the anomaly is due to a major fault, the module reports its findings about the root causes and returns to the top of the loop for the next telemetry cycle. MAIFLOWER uses TRIAD to remove ambiguity that can occur while diagnosing faults with MBR. Figure 4 shows how MAIFLOWER determines whether anomalous sensor values are caused by a diagnosable fault or if there was a MBR false positive.

TRIAD

As mentioned previously, MAIFLOWER complements MBR with TRIAD, a data-driven approach. TRIAD is a novel method for anomaly detection and diagnosis based on the idea of thermodynamic variables (TDs). TRIAD uses sensor data from nominal spacecraft operations to train its fault detection capability. TRIAD then trains its fault diagnosis capability with off-nominal data produced during different fault situations. Thermodynamic variables are aggregate values that describe the state of a system. Temperature and pressure are two familiar examples of thermodynamic variables. TRIAD uses aggregate values including but not limited to mean, variance, maximum value, minimum value, and maximum distance between consecutive sensor readings. See Appendix A for a full list of all aggregate values used by TRIAD. Each aggregate value will be generated over the last *n* samples. For example, one function for generating a TD, referred to as a "feature-generating function", is the mean over the last 100 samples at a specific sampling rate (30 msec is a typical sampling rate for MAIFLOWER). Because analyzing the data at multiple time scales can be helpful, TRIAD will have generated function for the same type of aggregate value, but with different numbers of samples. For example, the version of TRIAD used in MAIFLOWER has generated functions for computing them mean over the past 10, 100, 250 and 500 samples. In addition, TRIAD sequentially applies multiple different feature-generating functions to the data to generate richer features. An example of this benefit is when a feature-generating function doesn't provide much insight into the raw input data. However, when applied to the frequency domain data created by taking a Fourier transform of that raw data, it can consistently uncover anomalies that might otherwise go unnoticed. The current version of TRIAD uses a pool of feature-generating functions (that generate the TDs specified in Appendix A), as well as sequences of feature-generating functions that were designed specifically for MAIFLOWER. Future versions of TRIAD could instead automatically generate sequences of feature-generating functions until TRIAD consumes an allotted computation budget. These sequences would be automatically curated to reduce redundancy in the features TRIAD generate.

TRIAD is trained on fault-free training dataset to construct quantile-based empiric distributions over each of the generated TDs. When the system is online, out-of-distribution instances of features are detected as anomalies. The thresholds for what constitute out-of-distribution values for each TD are set depending on the desired sensitivity. TRIAD uses a validation data set to tune this threshold, moving along a tradeoff frontier between false positive and false negative rates. Typically, the challenge with approaches like TRIAD is that they cannot characterize a system's behavior when it is operating in an environment where they have not been trained, though they can still detect it. Using TRIAD in combination with MBR, as described in the next subsection, helps address this challenge at least partially. When TRIAD is operating in an environment where it has not been trained, there are two possibilities. First, TRIAD is dealing with data on which it has not been trained but that environment or mode is modeled by MBR and therefore be characterized and diagnosed. Second, neither TIRAD nor MBR model that space. We are investigating techniques like continual learning to address this case. Continual learning enables ML models to train on all available data without catastrophic forgetting [12]. Catastrophic forgetting is a phenomenon where neural networks lose accuracy on the environments or modes it was trained on in the past after being trained on a new environment or mode [13].

TRIAD diagnoses anomalies using vectorized recordings of previously known anomalies. It develops "zones" in the high-dimensional feature space, each of which corresponds to a type of fault. The version of TRIAD in MAIFLOWER uses manually designed boundaries to split the high dimensional space into zones corresponding to each fault type. Both nominal and fault data were generated from the simulation model of the VOLT and ROSA. We are unlikely to get fault data from the real system for this application and we will continue to use simulated fault data in the future, albeit from an updated higher-fidelity simulation model. In a future version of TRIAD, it will instead use unsupervised learning to automatically find the zones in the high-dimensional space that correspond to distinct anomalies. Specifically, TRIAD will fit a probability distribution in the high-dimensional space for each fault type by maximizing the log-probability of parameterized Gaussian mixture distributions. Gaussian mixtures are highly expressive distributions, especially in contexts where target distributions can be high-dimensional and multi-modal. This allows for incomplete sets of "cues" for specific types of faults to remain actionable, which can be helpful in the case of broken sensors. Using these parameterized distributions, TRIAD will assign a probability to each class of known anomaly in the event of a real fault, in addition to a probability that the fault is of an unknown type.

Both the anomaly detection and anomaly diagnosis components of TRIAD can be retrained with new data in the face of new operational circumstances with no development overhead, allowing TRIAD to continually learn and keep pace with new contexts, incorporating new knowledge of specific fault types over time. Given the infeasibility of generating fault data from the real system, we will use simulated data to demonstrate retraining capabilities.

On top of the statistical functions and other basic generating functions, TRIAD is also designed to incorporate data-driven anomaly detection methods, including state-of-the-art anomaly detection methods that use transformers and CNNs. To accomplish this, these neural network based approaches will be trained on a variety of downstream tasks including detection of synthetic faults, prediction of future data [14], and reconstruction accuracy [11]. Then, feature encoding layers of the neural network will be subject to threshold-based detection. This capability will be implemented and tested in the future work.

Hybridizing MBR and TRIAD

As mentioned, MAIFLOWER uses hybrid TRIAD and MBR to identify anomalous behavior in spacecraft telemetry data. This approach leverages the benefits of each while minimizing their disadvantages. When an anomaly is detected, there are a few possibilities: it is detected by both systems, only by TRIAD, or only the MBR system. TRIAD will detect all data streams that are "different" from the data it was trained on, and as such, it may flag sensor values as anomalies even when they are not. These false positives from TRIAD will be counteracted by MBR which only flags anomalies for sensor values that violate modeled physical constraints of the system. Because MBR and TRIAD are very different algorithms (especially in that one is model-based and the other, model-free), their agreement provides extra confidence in the result (compared to using MBR or TRIAD alone).

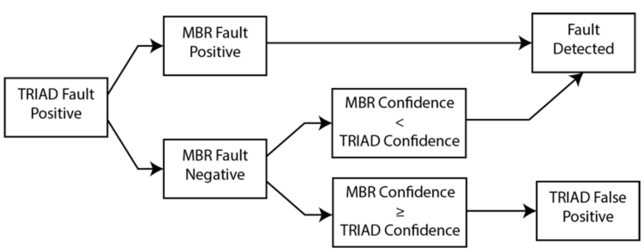

Figure 5. Diagram of the MAIFLOWER false positive detection system

When TRIAD and MBR disagree, MAIFLOWER will need to choose which result to use as the final conclusion. Both TRIAD and MBR provide a confidence value along with their fault detection and diagnosis results. If TRIAD and MBR disagree, this tie will be broken by whichever has higher confidence. See Figure 5 for a graphical representation of how MAIFLOWER handles TRIAD false positives. As previously mentioned, MBR and TRIAD have complementary strengths. They also have complimentary regimes in which they provide a high confidence in their diagnosis.

Simulation of VOLT and ROSA

While the goal is to evaluate MAIFLOWER on real hardware data in the near future, the team have conducted a set of evaluations using a simulated model of the Astrobotic VOLT to generate both nominal and fault data. Note that the model used for the simulation is different from the one used for MBR. The two were developed independently with a firewall ensuring that one did not bias the other. Furthermore, our objective was to follow the convention that the simulation model is a higher fidelity representation of the real system compared while the model used for MBR is a lower fidelity representation; this is consistent with the MBR approach in general where a model is very likely to be a simplification of the real system.

The higher fidelity simulation models normal force from the lunar surface, the associated coefficient of friction between the wheels and soil, and gravity as well as the stabilizing and attachment interactions between the AMP and the ROSA. Figure 6 illustrates how the AMP and ROSA are modeled, where the two components are modeled with springs interfacing between them. In Figure 6, springs A and B affect fore-aft motion while springs C and D affect port-starboard motion. Spring E fixes the ROSA to the rover body and has a very high spring constant to model a nearly rigid connection. Springs A and B have a spring constant of 28000 N/m, springs C and D have a spring constant of 21000 N/m, and spring E has 5000000 N/m. The ROSA can sway, and this oscillation effect is controlled by spring constants that differ between the fore-aft and port-starboard directions. The swaying of the ROSA causes an equal and opposite reaction on the mobility platform, thereby producing a rocking motion where the legs briefly leave the ground in time with the ROSA's oscillations. Even if the mobility base isn't rocking enough to lift any legs off the ground, the ROSA's oscillations will be reflected in the normal forces between the legs and the landscape. In other words, the physical simulation will account for the increased force on certain legs when the ROSA's oscillations shift the system's center of mass toward them.

The simulation model calculates normal forces and friction with an efficient linear optimization. This process first calculates how impulses affect all the legs of the rover, assuming partially inelastic collisions with the landscape. This calculation produces a normal force magnitude and vector for each leg. The friction forces on each leg are computed using those normal forces. All relevant faults can be simulated with our model. The shape and incline of the landscape can be dynamically adjusted to simulate soil collapse. To simplify the physics, the coefficient of friction between the wheels and the lunar regolith is dynamically updated to simulate soil slippage. Small offsets can be added to the orientation of the upper gimbal to simulate an offset of the ROSA to the local gravity vector.

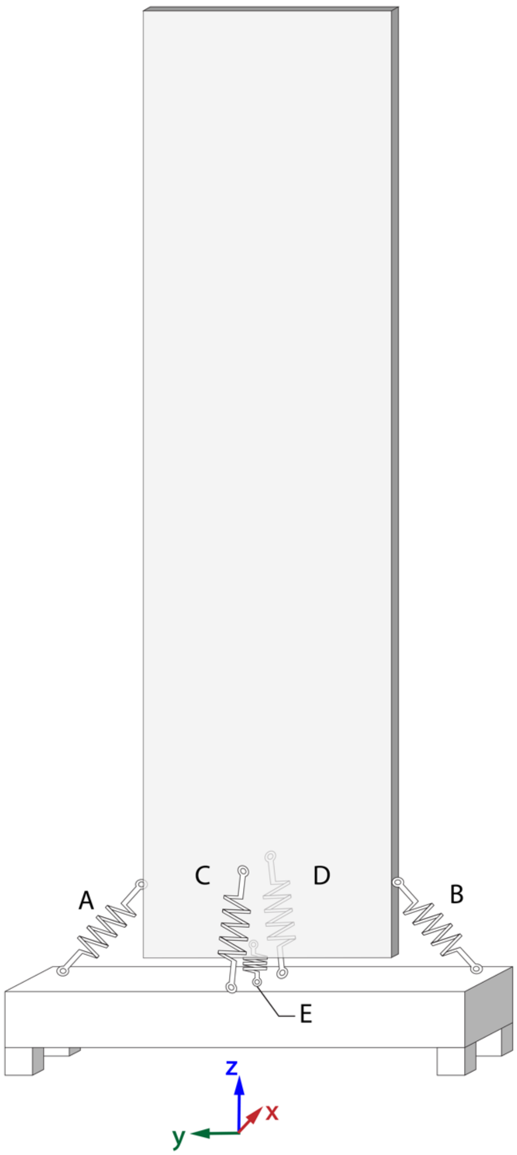

Figure 6. Spring based model with coordinate reference shown at the bottom

Evaluation

We evaluated MAIFLOWER in nine different scenarios which are listed in Appendix B. The evaluations started by perturbing our physics-based simulation model in some way (e.g. soil collapse, soil slipping) and recording data. We then used both TRIAD and MBR on the generated data to detect and diagnose the relevant fault. A successful response to the scenario occurs when either both MBR and TRIAD agree, or when they disagree but MAIFLOWER was able to successfully break the tie by choosing the outcome from the method with a higher confidence value.

4. Results and Discussion

We validated our simulation in nine diverse scenarios. Let us first consider the asymmetric soil collapse scenario. In this scenario, soil collapses beneath all wheels except the front left wheel, triggering oscillations in the ROSA in both the X and Y directions. Refer to Figure 6 to see the X and Y directions on the VOLT. Figure 7 shows the position of the top of the ROSA after the asymmetric collapse.

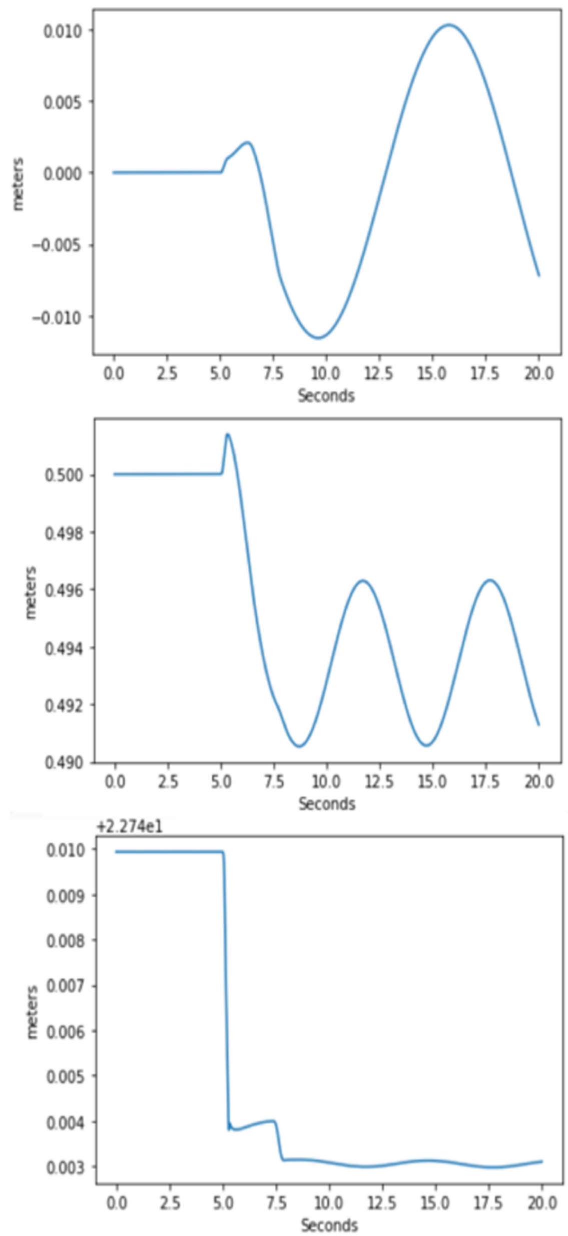

Figure 7. X (top), Z(middle), Y (bottom) positions of the ROSA tip after the asymmetric soil collapse. The damping coefficient was set to a very low value in the simulation. The initial shorter peak in the first sub-figure represents the initial fall.

It is not possible to show the graphs for every sensor for each scenario. Instead, we show graphs for select sensors for each scenario to give a sense for the data generated by the simulation and the impact of the faults on different sensors. Note that the X and Y directions display large magnitude oscillations while the Z direction displays a large step change followed by a small magnitude oscillation. These oscillations are of varying frequencies, as the spring constants holding the ROSA in place along its stable and unstable axes are different. The back right wheel hits the ground first, and the VOLT briefly sways, held by two wheels, until the back left wheel makes contact with the ground at around 7.5 seconds, stabilizing the system. The normal force induced on each wheel post collapse oscillates—this is due to the swaying of the ROSA.

In this scenario, the MBR module cross checks the four sensors measuring the force on the wheels with each other to determine if one of the sensors is faulty. Having confirmed that all four are showing anomalous behaviors, it concludes that sensor faults are not the cause of the anomaly. MBR then inspects the list of anomalous sensors for any patterns that indicate particular causes of failure. Here, it finds that the four sensors associated with the wheels and the two gimbal accelerations are anomalous and determines that the soil must be undergoing an asymmetric collapse. In our simulation test run, MBR took 2.7 milliseconds to process, measured from when the telemetry data is received to when the diagnosis for the current step has been made. TRIAD notices an anomaly 0.15 seconds after the collapse, determines the nature of the anomaly correctly instantly, and confirms with confidence 0.3 seconds after collapse.

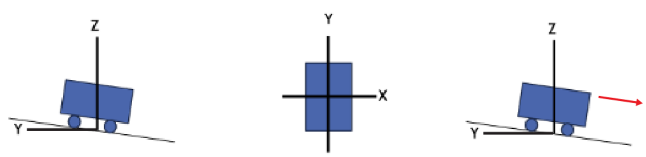

Figure 8. Sketch of soil slipping scenario, which labels the X, Y, and Z axes and shows the direction of slipping with the red arrow in the right most sketch

We will now discuss the symmetric soil slipping scenario. Refer to Figure 8 for a visual representation of how the rover is sliding down the slope. The red arrow in the right most sketch is the direction of motion. In this scenario, the rover lies on a 6-degree incline slope. Friction between the wheels and the lunar surface gives out, and the rover begins to slip. Friction is recovered in discrete shocks, which causes the rover to shudder. See Figure 9 for a visual representation showing VOLT picking up speed as it begins to slip, followed by velocity oscillations as the wheels repeatedly regain and lose traction and the final state where the vehicle again has zero velocity. The VOLT in total slides roughly 2.5 meters in the Y direction and 0.25 meters downward. Note that the slipping fault is triggered at 5 seconds and that before that the rover is stationary. MAIFLOWER will detect and diagnose the fault from the unexpected movement within 0.3 seconds after the sliding begins. Figure 8 shows a red arrow in the direction of the rover slipping down the slope in the Y direction and down the slope.

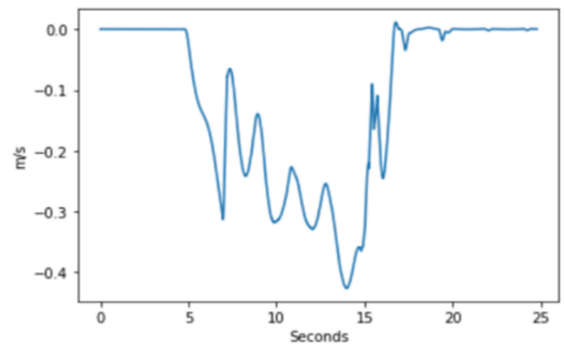

Figure 9. Y component of velocity in symmetric slipping scenario. The negative velocities occur because the slope goes downward in the -Y direction.

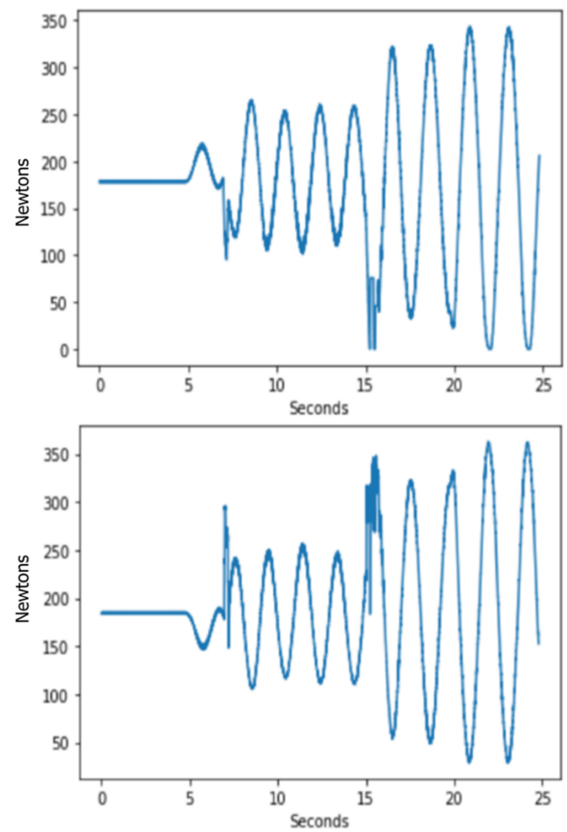

Figure 10. Normal force exerted on front (top image) and back (bottom image) wheels in the symmetric slipping scenario.

Note in Figure 10 that the normal force on the wheels oscillates -- this is a downstream effect of the oscillation of the ROSA. In this scenario, MBR again detects and diagnosis the error by inspecting the anomalous sensors for patterns that indicate specific faults, and it completes the diagnosis in less than 4 msec. TRIAD detects an anomaly at 0.18 seconds after slipping begins and correctly determines that the wheels are under soil slippage. TRIAD makes an accurate high-confidence prediction of 0.30 seconds after the VOLT begins to slip.

For all scenarios, the MBR loop takes between 2 and 4 ms to perform. In our simulation run, MBR properly detected and diagnosed all major faults and detected all sensor errors. TRIAD was highly successful in detecting faults and anomalies. TRIAD detected each anomaly with standard noise within 0.15 seconds and offered a diagnosis in 0.33 seconds. See Appendix C for a full table of runtimes for both MBR and TRIAD in all scenarios.

5. Conclusion

The goal of this effort was to develop a general fault detection system that can intelligently detect and diagnose faults across a wide range of subsystems that are present on unmanned spacecraft. The next goal of this effort is to apply MAIFLOWER's fault detection and diagnosis capabilities to real hardware. Towards this goal, Astrobotic has tested the VOLT at the NASA Glenn Research Center (GRC) Simulated Lunar Operations Laboratory (SLOPE) lab in summer 2024, and the team is currently working on training, validating and testing MAIFLOWER using that data. We are also beginning to investigate autonomous safing actions (i.e. actions that put the system in a safe mode). In the future, we will further test and mature the system for deployment onboard VOLT in a lunar mission.

Although, as described earlier, the component technologies of MAIFLOWER have been applied to a wide array of spacecraft subsystems, this is the first involving a primarily mechanical system. We have found three key benefits from applying MAIFLOWER to a mechanical system. First, MAIFLOWER provides a generalized modular fault management architecture that can quickly be spun up for any number of different subsystems. Second, MAIFLOWER provides autonomous, high-speed anomaly detection along with "root cause" analysis by correlating time-series data across subsystems. Correlating across subsystems enables the detection of cascading impacts of a single fault on a spacecraft as a whole. Third, Astrobotic's VOLT, which provides a real hardware platform to test on will prove MAIFLOWER's feasibility for adaptation for other spacecraft. The VOLT system will first be tested in the NASA SLOPE lab and then on the lunar surface.

Appendices

A. Aggregate Values

This appendix contains a list of all aggregate values that TRIAD analyzes to find anomalies. TRIAD has a generating function that takes time-series of telemetry data as input and produces a specific aggregate value. Note that there is a distinct generating function for different number of samples.

- Mean (taken over 10, 100, 250, 500 data points)
- Variance (taken over 10, 100, 250, 500 data points)
- Maximum value (taken over 10, 100, 250, 500 data points)
- Minimum Value (taken over 10, 100, 250, 500 data points)
- Maximum Distance Between Consecutive Sensor Readings (taken over 10, 100, 250, 500 data points)
- Maximum Consecutive Identical Readings (taken over 10, 100, 250, 500 data points)
- Average Fourier Frequency (calculated using Fast Fourier Transform) (taken over 10, 100, 250, 500 data points)
- Peak Frequency (calculated using Fast Fourier Transform) (taken over 250, 500 data points)
- Second Highest Peak Frequency (calculated using
- Fast Fourier Transform) (taken over 250, 500 data points)
- Peak Frequency Amplitude (calculated using Fast Fourier Transform) (taken over 250, 500 data points)
- (calculated using Fast Fourier Transform) (taken over 250, 500 data points)
- Dot Product Similarity Between Two Sensors (taken over 250, 500 data points)
- Dot Product Similarity Fast Fourier Transform of Two Sensors (taken over 250, 500 data points)

B. Prototype Scenarios

This appendix contains detailed descriptions of all of the scenarios that we used to test MAIFLOWER. We simulated each of these scenarios using our physical simulator and then produced sensor data for every sensor on the VOLT that we then analyzed with our hybrid MBR and TRIAD fault detection and diagnosis system.

- Symmetric Soil Collapse: Soil collapses 1cm under both wheels on one side of the rover at 5 seconds. This induces oscillation in the ROSA but does not result in tipping.
- Asymmetric Soil Collapse: Soil collapses beneath all wheels barring the front left wheel, triggering oscillations in the ROSA in both the X and Y directions.
- Soil Collapse Causes VOLT to Tip: Soil collapses half a meter along its fragile axis, causing the rover to tip. The collapse itself doesn't tip the rover – it destabilizes the ROSA, and its ensuing oscillation while the VOLT is less steady causes the collapse.
- Symmetric Soil Slipping: The rover lies on a 6-degree inline. Friction gives out, and the rover begins to slip. Friction is recovered in discrete shocks which causes the rover to shutter. The VOLT in total slides roughly 2.5 meters in the Y direction and .25 meters downward.
- Asymmetric Soil Slipping: The rover slips backwards and to the left, at a 6-degree incline. In

this scenario, friction is returned to the system faster (creating more dramatic oscillations in the ROSA) but is never fully recovered. This lower fraction creates subtle slipping and more erratic normal force distributions even after the VOLT is no longer sliding as the torque normally exerted by friction helps to counteract the oscillation of the ROSA. As a result of this, the VOLT itself briefly tilts at around 18 seconds – this corresponds to two nearly simultaneous peaks along the two axes of the ROSA's oscillation, which is enough to lift two legs up. Upon Recovering, the VOLT briefly tips in due to increased friction by this point, the VOLT no longer tips.

- ROSA Leveling Error: The upper gimbal was unable to completely level the ROSA before deployment and is deploying it at an angle offset by one degree. The offset produces subtler oscillation in the ROSA, as the equilibrium angle increases while the ROSA deploys. This scenario concerns the ROSA mid-deployment, at a height of 15 meters (by contrast with its final height of 20 meters)
- Frozen Sensor Error: Strain sensor on the back left leg freezes.
- Zeroes Sensor Error: The angular velocity readings coming from one of the horizontal cameras drop to zero. This could be due to the camera's vision being obscured, a hardware fault in the camera itself, or a failure in the flow algorithm used to derive angular velocity measurements.
- Varied Sensor Error: In this scenario, a broken IMU on the lower gimbal freezes some measurements and drops other measurements to zero.
- Simultaneous Sensor Error and Soil Collapse: A soil collapse is triggered and there is a simultaneous sensor error for the lower gimbal IMU pitch measurement which is relevant to the diagnosis of this collapse. Scenarios like these are important considerations as physical faults such as soil collapse can cause sensor failure.

C. TRIAD Scenario Runtimes

Table 1 shows the amount of time until TRIAD detects and diagnoses each fault after it is triggered. This includes both the TRIAD run time and the time required to gather enough data to detect and diagnose a fault. Note that in some scenarios TRIAD detects and diagnoses an error at the same time, and there is no delay between the detection and the diagnosis.

Table 1. The amount of time until TRIAD detects and diagnoses each fault after it is triggered, including TRIAD run time and the time needed for sufficient observations to confirm the fault

Scenario	Detect (s)	Diagnose (s)
Sym. Soil Collapse	.15	.27
Asym. Soil Collapse	.15	.3
Soil Collapse Causes VOLT to Tip	.15	.3
Sym. Soil Slipping	.18	.3
Asym. Soil Slipping	.15	.36
ROSA Leveling Error	1.49	1.49
Random Sensor Error	.15	.27
Frozen Sensor Error	.33	.33
Zeroed Sensor Error	.33	.33
Varied Sensor Error	.27	.33
Simultaneous Sensor Error and Soil Collapse	.15	.33

D. MBR Scenario Runtimes

Table 2 shows the amount of processing time required for MBR to detect and diagnose the fault in each scenario after the fault was triggered.

Table 2. The amount of processing time required for MBR to detect and diagnose the fault in each scenario after the fault was triggered

Scenario	Detect and Diagnose (ms)
Sym. Soil Collapse	1.7
Asym. Soil Collapse	3.6
Soil Collapse Causes VOLT to Tip	4
Sym. Soil Slipping	3
Asym. Soil Slipping	3
ROSA Leveling Error	2.9
Random Sensor Error	3.5
Frozen Sensor Error	

Zeroed Sensor Error	3.2
Varied Sensor Error	
Simultaneous Sensor Error and Soil Collapse	

ACKNOWLEDGEMENTS

The material is based up on work supported by NASA under Contract No. 80NSSC24CA098. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official view of politics, either expressed or implied, of NASA.

AUTHOR DISCLOSURE STATEMENT

The software frameworks/products MAESTRO, TRIAD and MAIFLOWER described in this work are proprietary to Stottler Henke Associates. The VOLT system is proprietary technology to Astrobotic. The ROSA system is proprietary technology to Redwire.

REFERENCES

[1] J. D. Frank, "Artificial Intelligence: Powering Human Exploration of the Moon and Mars".

[2] D. Stottler, S. Ramachandran, C. Healy, A. Singhal, and E. Finnigan, "On-board, autonomous, hybrid spacecraft subsystem fault and anomaly detection, diagnosis, root cause determination, and recovery," 2022.

[3] P. Robinson, M. Shirley, D. Fletcher, R. Alena, D. Duncavage, and C. Lee, "Applying Model-Based Reasoning to the FDIR of the Command and Data Handling Subsystem of the International Space Station," Feb. 2003.

[4] L. Ruff *et al.*, "A Unifying Review of Deep and Shallow Anomaly Detection," *Proc. IEEE*, vol. PP, pp. 1–40, Feb. 2021, doi: 10.1109/JPROC.2021.3052449.

[5] M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nat. Rev. Methods Primer*, vol. 2, no. 1, pp. 1–21, Dec. 2022, doi: 10.1038/s43586-022-00184-w.

[6] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982, doi: 10.1109/TIT.1982.1056489.

[7] D. Reynolds, "Gaussian Mixture Models," S. Z. Li and A. Jain, Eds., Boston, MA: Springer US, 2009, pp. 659–663. doi: 10.1007/978-0-387-73003-5_196.

[8] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Their Appl.*, vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 10, 2022, *arXiv*: arXiv:1312.6114. doi: 10.48550/arXiv.1312.6114.

[10] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[11] S. Ramachandran, M. Rosengarten, and C. Belardi, "Semi-Supervised Machine Learning for Spacecraft Anomaly Detection & Diagnosis," in *2020 IEEE Aerospace Conference*, Mar. 2020, pp. 1–10. doi: 10.1109/AERO47225.2020.9172454.

[12] S. V. Eeckt and H. Van hamme, "Unsupervised Online Continual Learning for Automatic Speech Recognition," Jun. 18, 2024, *arXiv*: arXiv:2406.12503. doi: 10.48550/arXiv.2406.12503.

[13] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/pnas.1611835114.

[14] H. Ren *et al.*, "Time-Series Anomaly Detection Service at Microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 3009–3017. doi: 10.1145/3292500.3330680.

BIOGRAPHY

***Richard Stottler** co-founded Stottler Henke in 1988 as a software company dedicated to providing practical solutions to difficult problems by skillfully drawing upon a large repertoire of artificial intelligence technologies. Under his leadership, Stottler Henke has grown steadily and profitably into a 60-person research and software development company with distinctive expertise in intelligent tutoring systems, intelligent simulation, automated planning and scheduling, and intelligent knowledge management. Dick provides technical leadership in the design and development of intelligent tutoring systems, intelligent planning and scheduling systems, and automated design systems. He combines a strong applied research record in artificial intelligence with practical experience in rapid and efficient knowledge engineering. He also led the development of intelligent planning systems for NASA space shuttle missions and aircraft assembly and automated scheduling for the International Space Station. Dick has written or presented dozens of papers and articles for publications such as the proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). He received his BS in engineering from Cornell University and his MS in computer science (artificial intelligence) from Stanford University.*

***Dr. Sowmya Ramachandran's** research focuses on the application of artificial intelligence (AI) and machine learning (ML) such diverse applications as automated fault management and advanced, adaptive training systems. In this capacity, she has worked extensively with generative AI models and natural language processing techniques and tools. She is currently a project manager of project to develop a personalized research assistant tool using large language models (LLMs). She also heads a NASA-funded effort to use AI and ML for detecting and diagnosing anomalies for NASA's lunar landing mission. Her past research efforts include diverse applications such as use of language models for automatic generation of multiple-choice test items from documents, an automated ML-based, intelligent health management system for deep space exploration habitats, and an after-action review tool for large team training exercises that used machine learning techniques to automatically group chat stream messages into coherent topics. She has published extensively in conferences of high repute and served on program committees for leading education and training conferences. She has recently co-authored a book titled "Handbook of augmented reality training design principles", published by the Cambridge University Press. Dr. Ramachandran holds a Ph.D. in Computer Science from The University of Texas at Austin.*

***Evan Finnigan** is an artificial intelligence software engineer at Stottler Henke Associates, Inc. He has built case-based reasoning, fault management, training, scheduling and planning software for a wide variety of complex real-world problems. Before working at Stottler Henke, Evan worked as a student researcher designing and building assistive robotics.*

***Abhimanyu Singhal** is an artificial intelligence software developer at Stottler Henke Associates, Inc. He has significant experience in artificial intelligence and machine learning. He has applied these techniques to many challenging domains incusing ship classification, the xPLSS spacesuit, and naval networking with multibeam antennas.*

***Chris Healy** is an artificial software engineer at Stottler Henke Associates, Inc. His expertise includes using convolutional neural networks and Generative Adversarial Networks to tackle a variety of transfer learning and domain shift problems. He also has expertise in a variety of uncertainty estimation and anomaly detection techniques in computer vision, including Monte-Carlo dropout networks, latent encodings, and null space analysis.*