On-board, autonomous, hybrid spacecraft subsystem fault and anomaly detection, diagnosis, and recovery

Dick Stottler, Sowmya Ramachandran, Christian Belardi, Rocky Mandayam
Stottler Henke Associates, Inc.

ABSTRACT

Future conflicts may involve attacks on space-based assets, physically and/or by interfering with their communications. In such situations it is imperative that the satellite already have onboard an autonomous ability to detect faults and other anomalies, determine the components involved and the root cause, determine the best method to recover mission capabilities, and schedule the required recovery plan, then adaptively execute it. Traditionally, Fault Detection, Isolation, and Recovery (FDIR) systems have utilized Model Based Reasoning (MBR), which requires knowledge of the subsystem design and the behavior of components down to the desired level of diagnosis. To the degree this information is readily available, it is important to make good use of it. However, the field of machine learning (ML) has shown that systems can also learn, off-line, the normal behavior of complex systems in many different environments and states, and then detect abnormal behavior in real-time. These system can also be trained with known abnormal states, and recognize these more specifically when they occur.

This paper will describe NASA-funded applications of MBR and ML systems to different spacecraft subsystems, including experiments with simulations and actual space hardware (in the lab), as well as additional techniques associated with automatic intelligent reconfiguration, planning and scheduling, and adaptive execution to create a complete high-level closed-loop approach to FDIR. This closed loop begins with the onboard monitoring of subsystem sensor data and associated commands. In the hybrid approach, this monitoring occurs in two ways: In the MBR portion of the monitor, a model of the spacecraft's subsystems, components, and their interconnections is used to compare the actual sensor values against what would be expected based on the current space environment, spacecraft state, and the commands. While allowing for noise, significant deviations of the sensor values are noted. Meanwhile, the ML portion uses a pre-trained Self-Organizing Map (SOM)-based architecture to produce high-resolution clusters of nominal system behavior that can distinguish between nominal and anomalous activity during online system monitoring. However an anomaly is detected, the system begins a diagnosis process. Similarly to the way a human engineer operates, the first step is usually to validate the anomalous sensor values. Often, sensors nearby (physically or in the connection diagram) can be used to confirm or refute the anomalous values. The latter case implies a sensor problem rather than a more significant fault. The next step is to reason upstream from the designated sensors to determine what components could potentially be at fault. If the sensors are adequately dense, the process may determine the faulty component or components. Frequently, however, there will be a set of possible candidate components, only some of which may be at fault. Most spacecraft systems have some degree of redundancy and, depending on the severity of the fault and the urgency with which capabilities must be restored, this redundancy can be used to either route around all of the possibly faulty components or be used to further refine the diagnosis by reconfiguring to isolate each specific candidate component, in turn, to determine which is at fault. These options can be automatically determined by using the graph of which components are connected to which others.

Once the at-fault component(s) is/are determined, the next step is to determine if full capabilities can be restored or if decreased capabilities will have to be accommodated. Especially in the latter case, replanning and rescheduling will likely be required. Usually a dynamic priority-based scheme is used to trade-off between different mission objectives, based on the current environment and tactical situation and optimizing constraint-satisfaction is used to meet as much of the desired tasking and as many of the mission objectives as possible within dictated timeframes and the new capability limits. The new task schedule is then passed to an adaptive execution system, often utilizing rules and/or finite state machines to execute the commands required to enact the schedule and to react to unexpected spacecraft subsystem responses. As the commands are executed, the sensors report new values resulting from the commands and/or from additional faults, and the entire cycle repeats itself, as necessary, to close the high-level loop. Current and future work will also be discussed applying these concepts to different spacecraft subsystems, including an experiment to be flown on the International Space Station (ISS), and capabilities being developed for Gateway and the exploration spacesuit.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

1. MOTIVATION

It is abundantly clear that in order to support the demands of future spacecraft missions, vehicles will need to be designed with a high level of on-board autonomy. While the traditional approach founded in predefined and conservative sequences of on-board executed commands can largely meet the requirements of many spacecraft operating in predictable environments, there are yet many forces driving the need for greater autonomy, including:

- Operations in less predictable contexts (e.g., deep space exploration and planetary robotic missions, automated space rendezvous maneuvers, an adversarial space environment);
- Increasingly demanding and diverse mission goals;
- Compressed timetables, budget restrictions, and tactical situations that demand higher vehicle availability and reliability; and
- Challenges posed by long communication delays and outages.

Obviously, Fault Management (FM) is an essential component of system autonomy. And while FM approaches have evolved over the years, a number of pervasive and persistent challenges remain and are in fact exacerbated by the growing complexity of our space science platforms—system complexity drives FM complexity (inclusive of structural complexity (e.g., the number of connected components); behavioral complexity (e.g., the variety of behaviors required and the complexity of the environment in which they take place); distributed complexity (e.g., the coordinated control of separate vehicles); and operational complexity (e.g., reliance on interactions between disparate systems & teams) [1]).

Since traditional FM envisions nominal operations that are punctuated by occasional occurrences of a defined set of anticipatable faults, it is relatively easy to recognize how expected system complexity would drive the number of such faults to levels that would make current FM design approaches impractical.

2. INTRODUCTION

An important spacecraft autonomy concept, as described in [2], considers spacecraft subsystem management as a different kind of (higher cognitive level) closed loop controller where the subsystem management problem is largely one of quickly responding to faults and other system changes in closed loop fashion at the appropriate time scale (e.g., sudden, dangerous faults must be responded to quickly while slow degradation of components inherently is more predictable and can be responded to at longer time scales). The closed loop consists of intelligent component and system characterization; fault detection, diagnosis, reconfiguration, replanning, and rescheduling; and adaptive execution. Our closed loop architecture is called MAESTRO (Management through intelligent, AdaptivE, autonomouS, faulT identification and diagnosis, Reconfiguration/replanning/rescheduling Optimization) and is designed to be easily applied to spacecraft subsystem management problems.

This process starts with fault detection, which can be performed with Model-Based Reasoning (MBR), Machine Learning (ML), or both. Each technique has its own benefits and disadvantages. We show how a hybrid system is very complementary, achieving the benefits of each and minimizing the weaknesses, verified with experimental results.

3. MODEL-BASED REASONING (MBR) DESCRIPTION

MBR is often used to detect faults and diagnose their causes by encoding the schematic information into a model of the subsystem being monitored, which includes the components (including sensors), 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 subsystem behavior and compare the simulated sensor output values to the actual sensor outputs. Significant deviations are used to detect faults. Then the model is used to reason which component faults are most likely to lead to the currently deviating sensor values. The set of possible faults (possibly including sensor faults) which explain the sensor values is the MBR diagnosis engine's output. The process of using the model to diagnose failures is considered somewhat analogous to the reasoning an engineer uses when using a schematic to try to diagnose the fault. The process can be made more efficient by various heuristics used by spacecraft engineers to quickly diagnose problems and include knowledge of which components are most likely to

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

fail (and how, e.g., mechanical relays tend to fail open while solid state relays tend to fail closed) and/or be the most likely explanation for certain types of sensor values. Because models encode the effects of contextual factors, they can be applied reliably across contexts. Model-based reasoning requires knowledge engineering efforts to encode these interacting effects.

There are several benefits of using MBR for Fault Detection and Diagnosis. MBR systems:

- Detect and diagnose anomalies never before encountered, as long as the modelled systems and components can support it.
- Will work well on Day One of in-space operations.
- Effectively utilize existing design knowledge of spacecraft subsystem engineers.
- Do not require large amounts of data (other than for software testing purposes).
- Can explain their reasoning and are human understandable, i.e., not a black box.
- Have models and code that can be validated for flight certification and predictable behavior.
- Can diagnose down to the lowest modelled component level.
- Can handle rare (but modelled) operating conditions such as use of a backup system.
- Can reconfigure, recalculate resources, and replan.

There are however some disadvantages to the approach. MBR systems:

- Are somewhat time-consuming to develop, especially in exquisite detail (effort proportional to the detail/size of the models).
- May need to have detection thresholds be somewhat larger, though those can be tuned or machine-learned with actual operations data.
- Are limited to known, designed, components and behaviors and known beyond-threshold deviations from these - they cannot discover unknown or unmodelled relationships.

4. BRIEF MACHINE LEARNING INTRODUCTION

Data-driven approaches using machine learning techniques overcome the problem of laborious, manual model development and, sometimes, computational inefficiency. Unsupervised approaches like clustering can automatically model patterns of nominal behavior from unlabeled data and monitor real-time telemetry for deviations from these behaviors [3]. Unsupervised learning has been shown to be very effective at detecting anomalies that have never been seen before. However, machine learning approaches cannot effectively trace the root causes for an anomaly for a variety of reasons: not having the information (i.e., labels) needed to discriminate between faulty and nominal behavior, and not having any insights into un-instrumented system components.

There are several benefits of using ML for Fault Detection. ML systems:

- Require very little development effort; no design knowledge or modelling time is required.
- Can discover unknown relationships (or unmodelled relationships because detailed models are too numerous/time consuming to develop) between sensor data in different modes (even cross-subsystem relationships), including detailed specific and subtle behavior in different nominal states or specific abnormal states.

There are however some disadvantages to the ML approach. ML systems:

- Require a lot of training data; while this can be simulation data, this will be model-based and therefore somewhat inaccurate; this can be mitigated by further training with operational data when it becomes available.
- Are Susceptible to “holes” in the training data.
- Cannot provide diagnosis, except in cases that have been previously trained for with labeled data.
- Cannot provide reconfiguration, replanning, or resource recalculations, except in cases that have been previously trained for.
- Are Black boxes that cannot, by themselves, explain their results, and can have unpredictable behavior in untested situations, which leads to possible issues with certification.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

5. HYBRID SYSTEM DESCRIPTION

A hybrid fault detection system uses both ML and MBR to monitor sensor values to identify anomalous behavior. When an anomaly is detected, there are a few possibilities: both systems detect it, only the ML system detects it, or only the MBR system detects it. In the first case, the system transitions to diagnosing the fault, using MBR, and taking further actions as described in the MAESTRO Description Section. If there is disagreement, the hybrid system attempts to resolve it. If the ML system detected the anomaly, it will include a quantitative list of the features (sensor values) that most contributed to the anomaly, these can be used to quickly retrieve the training data most similar to the current situation to determine if the ML system was trained for the current configuration. If not, it is likely the anomaly is not a real fault, just outside the bounds that the ML system was trained for. Alternatively, those same most important sensor values can be examined in the MBR system to determine if they are near fault detection thresholds, in which case, the fault is probably real and the thresholds just happened to be too high for the current circumstances. Additionally, the MBR system can provide the list of components that are upstream from the important sensors as well as which sensors could be explained by a fault in each of those components. A higher number of anomalous sensors explainable by a component increases the probability that that component is at fault. If the fault is determined to be real or likely to be real, MBR can be used to diagnose which component is most likely and reconfigure the system as necessary.

The third case is that the MBR detects the anomaly and ML does not. The first subcase is that the fault is real. This should only occur in the somewhat rare case that the ML system has not been trained on the current spacecraft subsystem configuration. When the spacecraft enters a configuration for which the ML has not been trained, it essentially will flag every telemetry packet as anomalous, forcing it to be ignored while the spacecraft remains in this untrained-for configuration. If a real fault occurs at this time, the ML system cannot by itself distinguish it from the entire (what it considers) anomalous situation it is in, although it may flag additional sensor data when the fault occurs. As before, the MBR will be used to diagnose and recover from the fault. The other subcase is that that fault is not real. This can only occur if the model is wrong in some way (missing something, something unmodelled, some behavior incorrectly modelled). As before, training data most similar to the current situation can be retrieved and, if it is very close, it gives confidence to the hypothesis that the fault is not real. Additionally, if the sensor data that the MBR system detected as anomalous is barely over the threshold or quickly returns to being below threshold then perhaps the thresholds were set too narrowly or perhaps there is more noise than was allowed for.

A hybrid system that simultaneously detects faults with both MBR and a ML approach can effectively combine the advantages of both technologies and mitigate their weaknesses. For example, if both approaches detect the same fault, that greatly increases the confidence in the result, especially since both systems will have been developed independently with radically different methods. And, after detection, the MBR can provide diagnosis, reconfiguration, capability recalculation, and replanning. Similarly, if both systems are not detecting a fault, this is independent agreement that nothing is wrong, i.e., that the behavior is nominal, based both on a priori modelling of the system and learned relationships between sensor values. In cases where there is disagreement, additional data can be retrieved from both systems to ascertain whether the partly detected anomaly is real or not.

The advantages of a hybrid MBR-ML detection and diagnosis system are that it:

- Utilizes a priori design knowledge when it exists and it is efficient to do so.
- Utilizes real or simulation sensor data when it exists and as it is produced.
- Utilizes very independently developed anomaly detection technologies for increased assurance.
- Can use its model to diagnose, reconfigure, recalculate, and replan in reaction to newly detected faults (whichever technology detected them) and can explain its reasoning.
- Can discover unknown or unmodelled relationships between sensor data in different modes. These relationships can span across subsystems.
- Can detect and diagnose anomalies never before encountered or trained for, using its MBR component and handle rare (but modelled) operating conditions.
- Can use the ML component to disambiguate overlapping MBR states. For example there is an SEU/latchup state in the LabSat where the increased power consumption is within range of a normal operating mode so that the MBR system cannot tell whether the component is behaving nominally or not, but the ML system could learn more subtle relationships in order to determine whether the state is nominal or anomalous.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

6. MAESTRO SYSTEM DESCRIPTION

Shown below in Fig. 1 is the high-level architecture for MAESTRO. In general, as shown, MAESTRO can support fully autonomous, closed loop execution. Nominally, the loop might begin with replanning goals, such as the need to reconfigure to restore power after a fault and subsequent safing operations. The Planner determines the best course of action (COA) (or several candidates). Each COA consists of the set of tasks that should be performed to fulfill its goals along with constraints and the resources that each task needs. For each COA, the scheduler generates an optimized schedule and reports back to the planner the feasibility (i.e., whether all the constraints could be met (such as whether available power will meet the requirements of all tasks, deadlines, etc.)) and the optimality possible. The Executive uses high-level priority and trade-off knowledge along with the impacts and ramifications of each COA to select the best COA, based on predefined trade-off criteria. Then the selected COA and its associated schedule are sent through MAESTRO's Interface to the Adaptive Execution (AE) module, which is based on SimBionic, and which executes the procedure associated with each task in the schedule. In order to adapt, AE also receives feedback in the form of sensor data. AE's adaptation may involve simply retrying the command, or other simple alternatives described in the procedure. These could also include predefined contingencies, actions to immediately take when something goes wrong, such as simple safing procedures for the spacecraft to minimize damage.

Based on the feedback, if a task fails or is significantly delayed, AE, through MAESTRO's Interface, informs the Scheduler and Planner so they can reschedule and/or replan. Alternatively, MAESTRO can be activated when an independent system first determines that there is a problem (and presumably safes the equipment). Meanwhile, the Diagnosis module is receiving the command and sensor data and generally monitoring the operation to detect a problem or fault. If it detects one, it then determines which component is malfunctioning or possibly a candidate set of possibly malfunctioning parts. These are sent to MAESTRO's Interface, which converts them into what the implications are in terms of Resources for the Planner/Scheduler. For example, a malfunctioning component likely means that some modeled resource is no longer available or has its capability reduced in some way, (e.g., there may be less power available than originally planned for). MAESTRO's Interface may create additional goals for the planner such as trying to perform noninterfering actions to refine the diagnosis, minimize equipment damage, and other problems associated with the fault. Based on the new situation and possibly new goals, the Planner and Scheduler combine to generate new COAs and associated new schedules appropriate for the new circumstances.

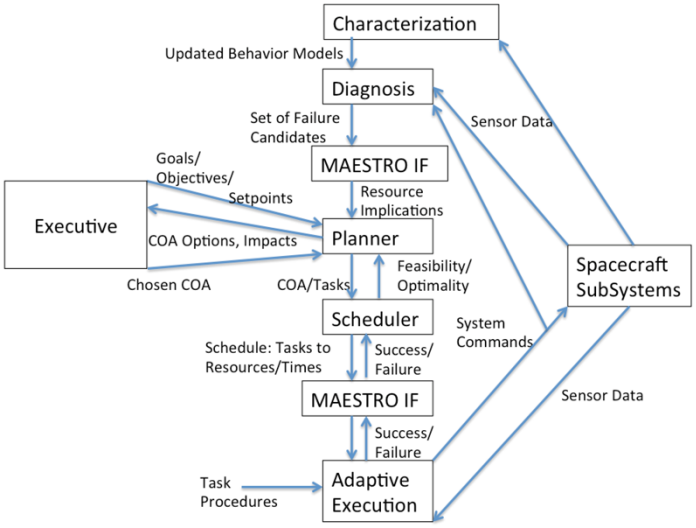

Fig. 1: MAESTRO High-Level Architecture

By incorporating the existing MAESTRO framework and integrating with the several Diagnosis, Planning, Scheduling, and Adaptive Execution software modules, the MAESTRO architecture can be applied to many different space craft subsystems to create a tailored MAESTRO closed loop system.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

MAESTRO includes facilities for translating the results of diagnoses (e.g., failed components) into the corresponding ramifications in the planning and scheduling models, e.g., which resources are no longer available (or only available at reduced quantity). For example, the failure of a specific photovoltaic (PV) panel may reduce the available generated power by half. A degraded set of panels may only be able to produce 80% of the power that they used to. A failed relay may mean that an entire part of the power distribution system is unavailable to route power and only a redundant power bus can be used.

Typically planning and scheduling systems both utilize some form of constraint based reasoning and a model of the planning and scheduling problem that includes resources, constraints, and actions, though optimally very different specific constraint-based technologies should be used for planning versus scheduling. Constraint based planning systems often use very unrestricted search over the set of constraints to investigate the full number of possibilities for achieving the objectives, given the constraints. This search is intractable unless the number of variables is kept small and therefore they tend to represent planned activity at a very high level, often referred to as a course of action. Given a specific course of action, a scheduler is used to fill in the specific details, the specific resources (including power) assigned for specific actions at specific times. Good heuristics exist that form the basis for highly efficient (even linear) algorithms for making near-optimal assignments of resources to actions, while obeying all applicable constraints. The scheduling system will need to predict power levels available using applicable formulas given the characterization of the electrical power system (EPS) components and current sensor values as well as planned actions.

Once a schedule is generated, it must be adaptively executed. MAESTRO can do this because the names of scheduled actions refer to behaviors defined in SimBionic. SimBionic behaviors are represented as hierarchical Behavior Transition Networks (BTNs), which are flow-chart like graphs that specify what task should be executed based on the flow of control through the network and values of variables referenced by transitions between tasks. The variables are often sensor values. The first action in a schedule might be to ensure that the applicable power bus is powered and then connected to the equipment to be used by the next action. This action might be represented as a BTN where the first task is to connect the specified bus to the specified source of power and when that is complete to connect that bus to the equipment. Both of these tasks might involve checking a relay and, if it is open, then issuing the command to close it and then checking that it actually closed—and if not, retrying the command once or twice before finally signaling a fault with the relay and failure of the task. Independent of any relay failures, transitions can exist that interrupt normal tasks when any sensed value reaches or trends toward a dangerous level, for example, a high or rapidly rising temperature or current level, or falling voltage, causing transition to various safing tasks like disconnecting suspect and nonessential equipment.

7. MACHINE LEARNING DESCRIPTION

We use Self-Organizing Maps (SOMs) as the basis for modeling system behavior. Once trained on normal system behavior, SOMs are powerful at detecting behavior previously not encountered in the training data (i.e., anomalies). Upon detecting anomalous behavior, our approach uses a supervised classification approach to determine a subset of measurands that characterize the anomaly. This allows it to help localize faults and thereby provide extra insight.

7.1 Self-Organizing Map Background

ADTM leverages the benefits of a Self-Organizing Map (SOM) for implicit data clustering and anomaly detection. Also known as a Kohonen map, a SOM is a two-layer artificial neural network (ANN) that uses unsupervised learning to produce a low-dimensional representation of the training samples [4]. Inspired by the way sensory input (auditory, olfactory, tactile, etc.) map to specific areas of the cerebral cortex, SOMs are also tuned to various patterns of input data during training (Fig. 2). Consistent with this analogy, the nodes in the output layer of a SOM are also called “neurons.”

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

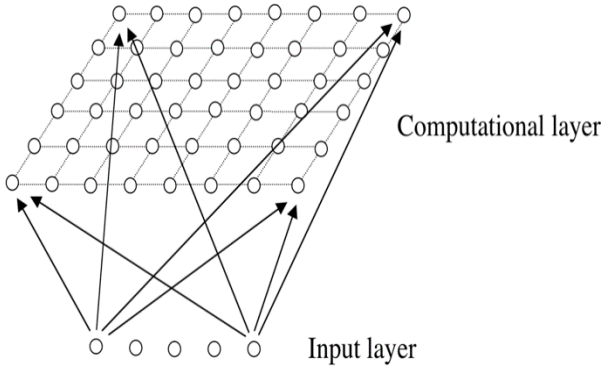

Fig. 2: Self-Organizing Map Diagram

The goal of training a SOM is to transform incoming inputs to a 1- or 2-dimensional map in a topologically ordered fashion such that points that are close together in the higher-dimensional input space are close together in the lower-dimensional output space as well. This mapping allows us to detect patterns of normal or anomalous behavior in a system, as different types of behavior map to different output units.

Specifically, the *N*-dimensional input data is fed into the SOM in the first layer and fully connected to a lattice of output neurons in the second layer. Each output neuron is associated with a *N*-dimensional weight vector. We represent an output neuron by its two-dimensional coordinate of its position in the grid. Like the clustering technique *k*-means, the two dimensions of the grid are parameters that are tuned during model validation.

Unlike *k*-means, however, the clusters learned during SOM training are topologically ordered through the following competitive learning process:

Each input vector is compared with the *N*-dimensional weight vector associated with each output node. The closest output node is chosen as the winner, or 'Best Matching Unit' (BMU), where 'close' is defined by a distance function (we chose Euclidean distance). Each BMU is associated with an entire neighborhood of related neurons whose weight vectors are also updated, though to a lesser extent, proportional to their distance to the BMU in the 2D output lattice.

In other words, entire neighborhoods of related neurons get updated in the direction of the input data that is closest to them, so that the topology of the *N*-dimensional input space is preserved in the 2-dimensional output space. We can think of this learning rule as pulling the weight vector associated with the BMU in the direction of the input vector. All neurons in the same 'neighborhood' are also dragged along, but to a lesser extent.

7.2 Anomaly Detection

Once trained on nominal data (as described above), the SOM maps new data seen online to the most similar weight vector of the output neurons, using Euclidean Distance as the similarity metric. Recall that we refer to the winning output neuron as the Best Matching Unit (BMU). The difference between the BMU's weight vector and the test point is the Minimum Quantization Error (MQE). A low MQE implies that the new sample closely aligns with a previously seen sample from the training data and is therefore nominal, whereas a higher MQE connotes that the point is anomalous, either because it contains a true fault or because it captures novel nominal behavior unseen during training.

8. EXPERIMENTS/RESULTS

A number of experiments were conducted based on the concepts described above for both real spacecraft hardware (residing in a lab) and the Martian Transit Vehicle (MTV) simulation developed by NASA's Johnson Space Center (JSC). In the subsections below, first the LabSat will be described followed by the experiments performed as well as their results. Then the MTV simulation will be described, at a high level, followed by the experiments run with it and the accompanying results.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

8.1 LabSat Description

We validated the performance of MAESTRO's fault detection, diagnosis, and recovery capabilities on a lab-residing CubeSat (named "LabSat"), originally designed after the Ionospheric-Thermospheric Scanning Photometer for Ion-Neutral Studies (IT SPINS) project by Montana State University (MSU) to study the nocturnal ionosphere.

The MSU LabSat was subdivided into three circuit boards. Board 1 was designated for power generation and storage (solar array simulators and batteries), while Boards 2 and 3 had redundant regulators and loads consuming power from Board 1.

The measurands for each board consisted of:

1. An outgoing voltage (in Volts) sensor for every component.
2. An outgoing current (in Amps) sensor for every component.
3. Three-state control switches connected to power-consuming loads on Boards 2 and 3. These loads were either powered by, depending on switch configurations, some combination of Solar Array 1 (SA1), Solar Array 2 (SA2), Battery 1 (BAT1), or Battery 2 (BAT2) from Board 1, and they were thus connected either to Power Bus 1, Power Bus 2, or neither (i.e., they were shut off).

Importantly, the LabSat enabled us to insert a variety of hardware faults, which we used as test sets to validate the performance of difference MAESTRO systems.

8.2 LabSat Experiments

Several examples are presented here. MAESTRO's fault detection submodule correctly recognized non-fault behavior 100% of the time in many diverse scenarios and operating conditions. This was possible by allowing for sensor noise by requiring that a sensor reading would not be considered anomalous until it was out of bounds from its predicted value two times in a row. Most of the time, the ML component agreed, thus increasing the system's certainty that there were no component faults.

In a few instances, the ML incorrectly designated nominal behavior as anomalous because it had encountered a configuration or operating mode that it had not encountered during training. In all these cases, the issue was obvious because it had either identified switch setting telemetry values as anomalous or had identified sensor values associated with a component that were obviously nominal for the operating mode that it was in. Additionally, retrieved similar telemetry showed that no data for either the switch settings or the specific problematic operating mode were in the training set. This showed how the combined system allowed MBR to cover any holes in the ML system's training data, a key benefit of a combined MBR-ML system.

MAESTRO and its MBR components were tested against the hundreds of faults that the LabSat is capable of exhibiting and correctly detected, diagnosed, and recovered from every fault successfully. A few examples are shown below.

One of the simplest is a single Single Event Upset (SEU) fault (in the CDH) as shown in Figures 3 and 4 below. As shown by the leftmost arrow in Fig. 3, CDH suddenly experiences a high current (purple rectangle, scale to the right). Also note that this is reflected by an increased current in the Regulated 3.3 V Power Bus 1 (REG 3v1 in Fig. 4 since CDH is being powered by Bus 1). MAESTRO detects the high current and correctly diagnosed the fault and correctly plans the solution, which is to cycle the power. This is performed with commands to switch the CDH switch to Bus 1 Power off for a time and then back on as shown by the rightmost arrow in Fig. 3. As shown in the CDH plot, current then returns to normal, i.e., a correct fix to the problem was executed.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

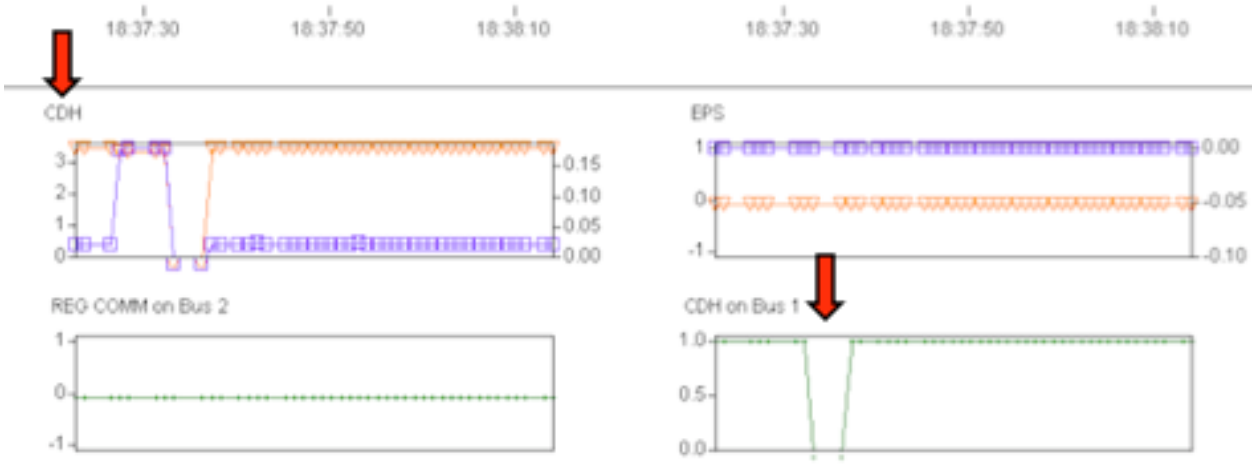

Fig. 3. CDH Current and Voltage Plots and CDH on Bus 1 Switch Plot

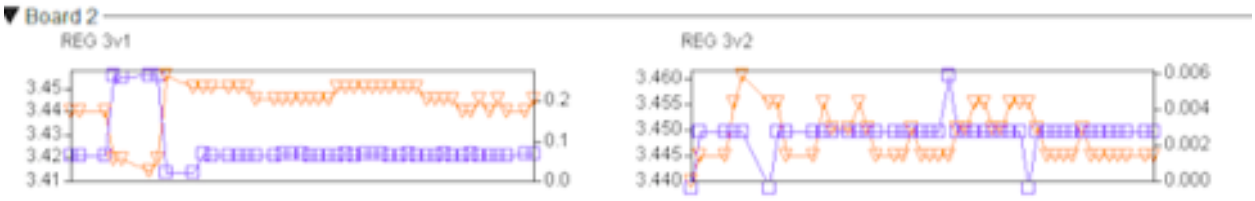

Fig. 4. Regulated 3.3 V Power Busses 1 and 2 Current and Voltage

Figures 5 and 6 show a two-fault scenario where one load experiences an SEU and another a stoppage. These are both correctly diagnosed and a reset task planned, scheduled, and executed as shown below. In one fault, the Reg Comm load experiences an SEU as evidenced by the high current (purple squares). The power is reset by commanding the REG COMM on Bus 1 to switch to 0 for a short time then back to one, which fixes the high current condition. Note that the voltage scale is on the left and the current scale is on the right).

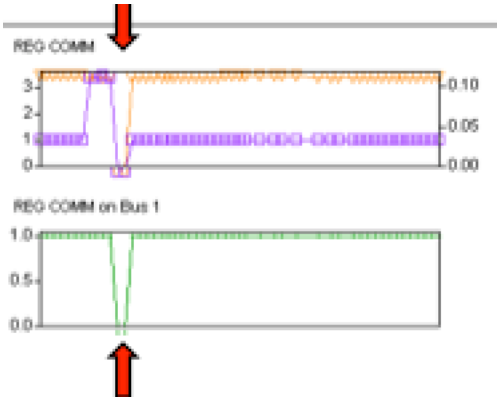

Fig. 5. SEU Fault on REG COMM as part of a 2-Fault Scenario

In the other (simultaneous) fault, CDH experiences a stoppage fault as indicated by the low (but non-0) current (purple squares). This is fixed by resetting the power by commanding the CDH on Bus 1 to switch to 0 for a short period of time then commanding it back on which fixes the problem as evidenced by a return to nominal current by the CDH.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

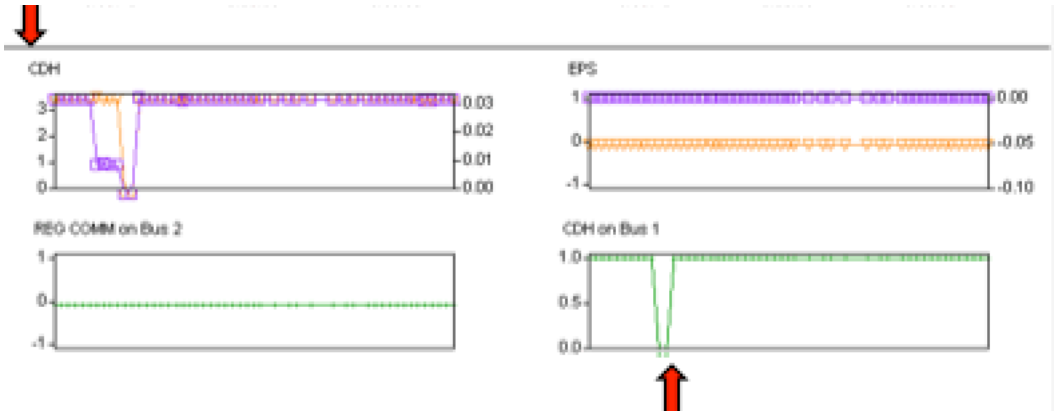

Fig. 6. CDH Stoppage Fault as part of a 2-Fault Scenario

A Regulator failure was also demonstrated. Reg 5v1 fails as shown in Fig. 7 with current and voltage both going to 0 (this is also reflected by voltages and current dropping to 0 for the CTIP load connected to the REG 5v1 Bus at the same time (in Fig. 8). At that time, both ADCS (on Bus 2) and CTIP (on Bus 1) were running simultaneously. When the fault occurs, MAESTRO immediately diagnoses the Bus 1 failure but allows the ADCS task to finish to completion, based on the convention of allowing all tasks that are unaffected by the fault to complete. MEASTRO reschedules all tasks (based on the convention that any interrupted task requires that all tasks be rescheduled), now allocating CTIP on Bus 2 as indicated by the CTIP on Bus 2 switch telemetry plot. Due to regulator constraints, the two tasks cannot run at the same time on Bus 2 so ADCS is forced to wait until CTIP completes before it can start.

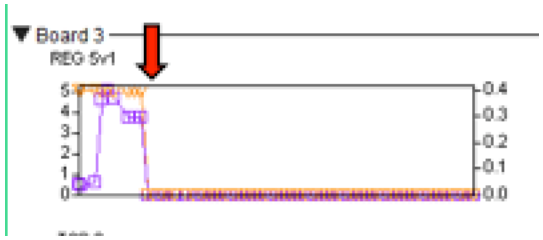

Fig. 7. 5.0V Regulated Power Bus Failure

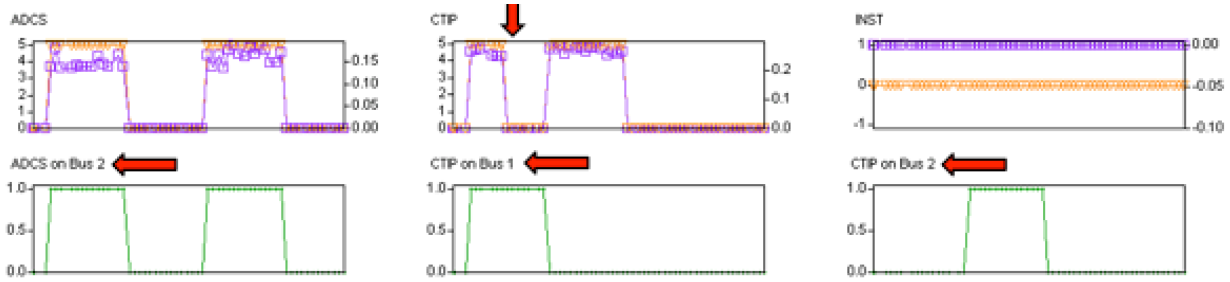

Fig. 8. CTIP and ADCS during REG 5v1 Failure

The REG 5v1 failure was also examined by the ML SOM system which had been trained on a variety of nominal data. It quickly correctly flagged the telemetry as anomalous along with flagging the 5v1 REG voltage sensor as the most important as well as flagging the sensors for the voltage of the two connected loads as second and third in importance, as shown in Table 1. With two independent identifications of an anomalous condition, certainty that a fault did exist was increased. Further, both the MBR and ML systems agreed on the affected sensors. The MBR system could then trace upstream and identify the fault as the 5v1 REG since its output was anomalous but its input voltage was normal. These examples showed the confirmation capability of a hybrid system.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Table 1. 5v1 Regulator Fault Localization

SOM	Salient Feature	Importance Score
SOM 1	N/A	N/A
SOM 2	N/A	N/A
SOM 3	Reg 5V1 V	38.6
	ADCS V	35.4
	CTIP V	17.1

8.3 MTV Simulation Description

The MTV Simulation simulates several subsystems in a hypothetical Mars Transit Vehicle including, Electrical Power System (EPS), Thermal Management System (EPS), Crew Metabolics, Pressure Control System, Water System, Environment Control and Life Support System (ECLSS), Crew Cabin and Airlock, etc. Most of our work was focused on the Thermal Control System which consists of 3 coolant loops, one internal and two identical external ones. The internal loop, The Internal Thermal Control System (ITCS), is shown below. ITCS's most important components are the two pumps and heat exchanger with the two external loops.

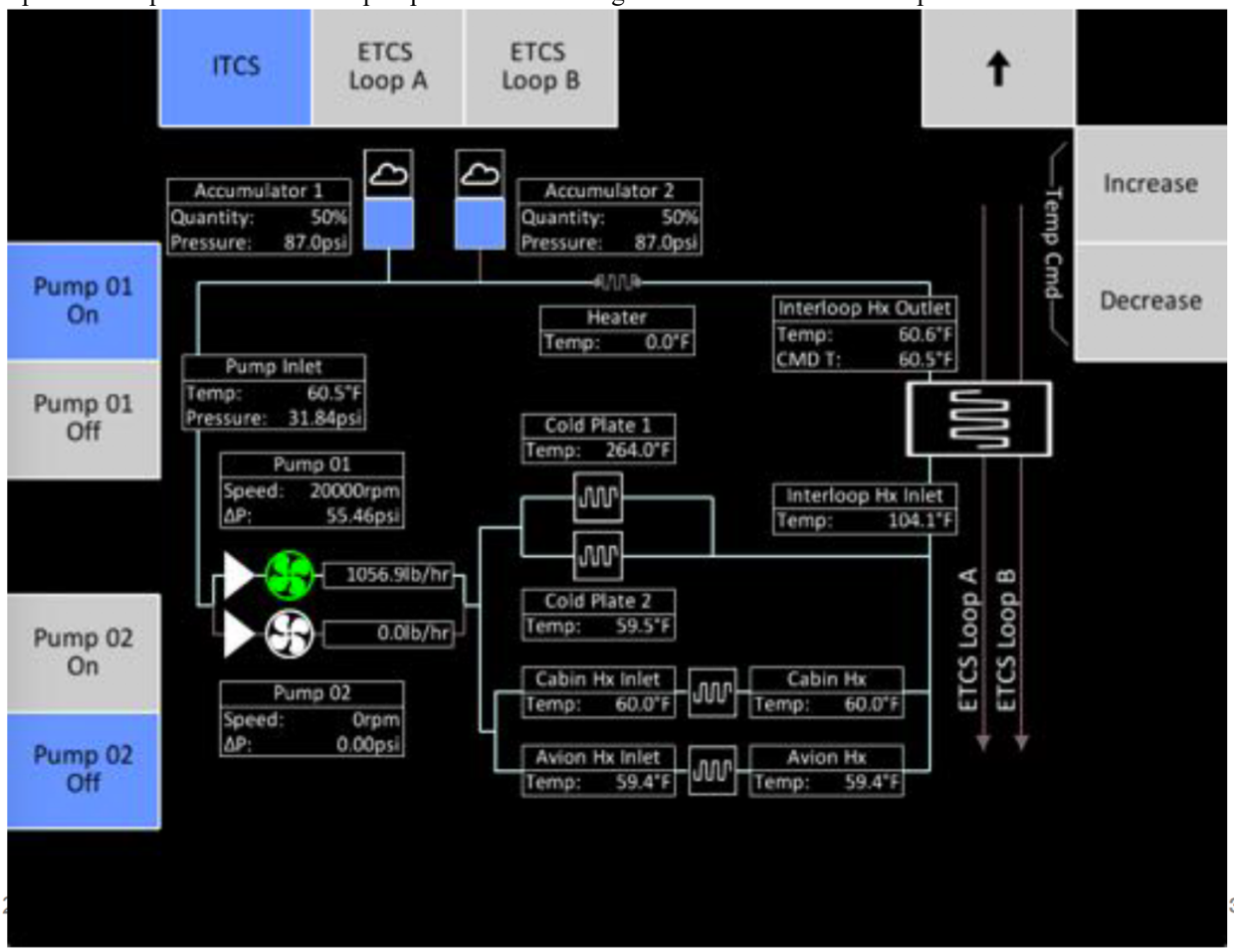

Fig. 9. ITCS Schematic

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

One of the external loops is shown below. It also includes two redundant pumps and the Heat Exchanger in the ITCS is also shown.

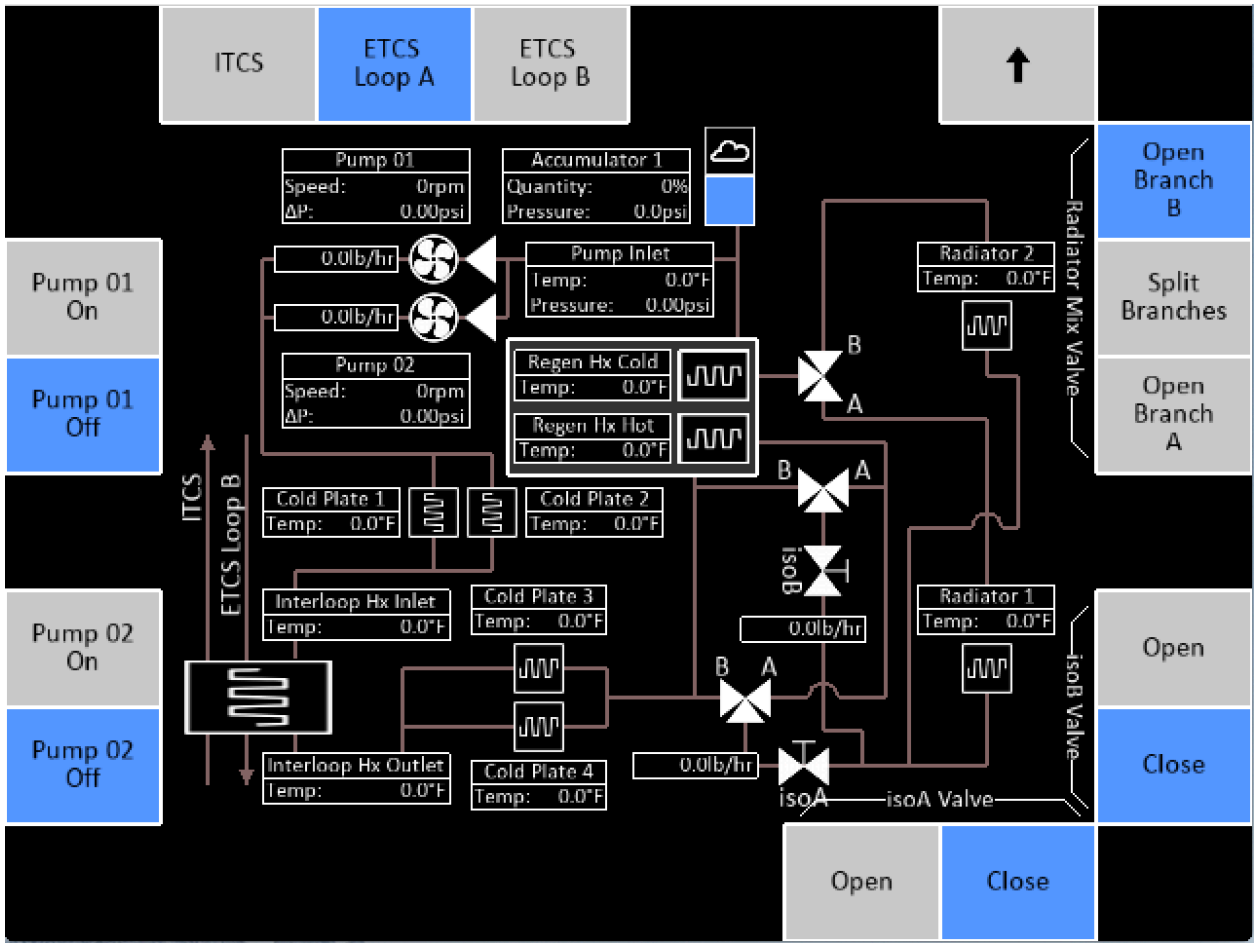

Fig. 10. ETCS-A Schematic

8.4 MTV Experiments and Results

Of course, the MBR component of MAESTRO would always perfectly classify nominal behavior and detect and diagnose all faults in the MTV simulation by just using the same model that the MTV simulation used. A much more interesting experiment would be if the ML system could learn nominal behavior from the simulated sensor values alone, and then correctly flag data resulting from simulated faults as anomalous. To test this we performed several experiments.

In the following ML experiments we use a Self-Organizing Map (SOM) to learn nominal behavior from subsystem sensor data. We use the SOM's learned representation for in-pattern (i.e., nominal) behavior to detect anomalies. Detected anomalies are then characterized by a feature-wise comparison of the anomalous data with the SOM to calculate feature importance. The feature importance identifies which dimensions of the data are contributing most to the anomalous classification.

The first experiments focused on ETCS-A. The MTV has a built-in ability to clog a pump and/or make a closed valve leak. In both experiments, the ML system first correctly identified normal behavior as nominal and then correctly classified the data associated with each simulated fault as anomalous, correctly identifying the most

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

relevant sensor values so that the MBR component could trace upstream to identify the faulty pump in one case and the faulty valve in the other, again showing the advantage of having a symbolic model associated with the ML detection system.

We then did a set of experiments designed to show the synergy behind the MBR and ML approach. Recall the trade-off between level of effort and amount of detail in an MBR system. To illustrate this, we assumed that the MBR system, although having a complete model of the components, had no notion of temperature.

In the first of the hybrid experiments, we repeated the ETCS-A experiment, but with no sensor information available with regards to the pumps. Specifically we removed the output pressure and flow rate sensors for both pumps. This meant that there was no direct way to tell if the pumps were faulty or not and therefore no way for an MBR system, by itself, to detect and diagnose the faulty pump.

We used data from the ETCS A to train and evaluate our system. For this experiment we had a training set of nominal data, a test set of nominal data, and a test set of fault data. In order to create this scenario, we removed all features from the datasets that pertain to pumps 1 and 2. This include the pumps' delta pressures, flowrates, and RPMs. The fault data captures the ETCS A system with a blockage in pump 1 and the redundant pump 2 turned off. The model is trained on the nominal training set and evaluated against the two test sets. We find that it successfully detects all anomalies (See Tables 2 and 3).

Table 2. SOM accuracy detecting the pump 1 blockage without direct sensor data from pump 1.

	Accuracy		
Dataset	**Nominal Training Set**	**Nominal Test Set**	**Fault Test Set**
SOM	98.64%	100.0%	100.0%

Table 3. SOM precision, recall, and f1 score on both test sets.

	Precision	**Recall**	**F1 Score**
SOM	100.0%	100.0%	100.0%

This fault manifests in ETCS A primarily as out of pattern temperature readings immediately downstream of the pump, allowing the MBR system with just a model of the connections and without knowledge of temperature behavior to correctly hypothesize that the pump is faulty (See Fig. 11).

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

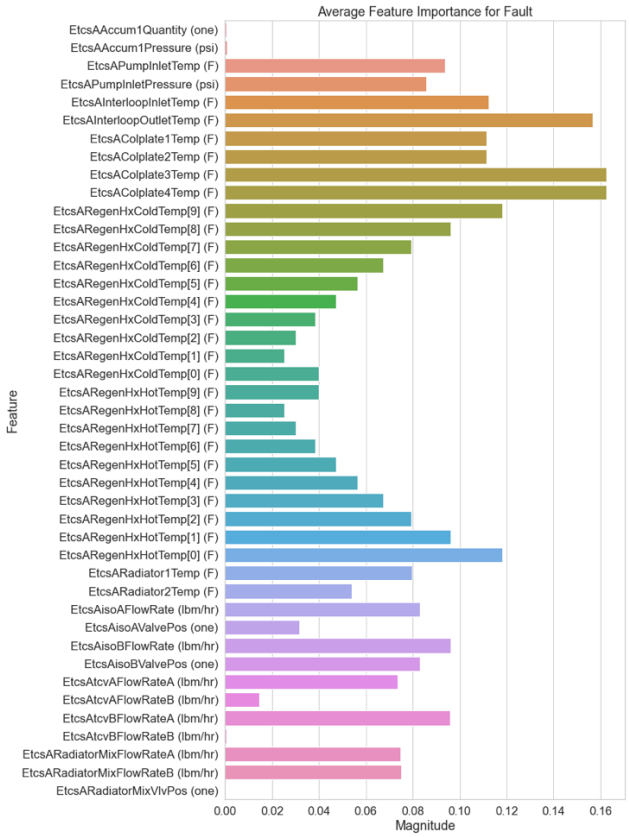

Fig. 11. Average feature importance for ETCS A calculated over the fault test set

Recall that one of the advantages of a ML system was the ability to discover previously unknown relationships. We executed a series of experiments to showcase this capability and the leverage of this power possible with a hybrid system.

This first experiment was designed to see if the ML system could discover how the heat exchanger in the ITCS was supposed to work. We used data from only the Internal Thermal Control System (ITCS) to train and evaluate our system. For this experiment we had a training set of nominal data, a test set of nominal data, and a test set of fault data. The specific fault is a failure in the heat exchanger between the ITCS and the rest of the Thermal Control System (TCS). The model is trained on the nominal training set and evaluated against the two test sets. We find that it successfully detects all anomalies (See Tables 4 and 5).

Table 4. SOM accuracy discriminating between nominal and anomalous ITCS data.

	Accuracy		
Dataset	**Nominal Training Set**	**Nominal Test Set**	**Fault Test Set**
SOM	98.75%	98.25%	100.0%

Table 5. SOM precision, recall, and f1 score on both test sets.

	Precision	**Recall**	**F1 Score**
SOM	99.65%	100.0%	99.83%

This fault manifests in the ITCS as out of pattern temperature readings. We find that our feature importance calculation successfully identifies temperature sensors in the ITCS as the most salient features (See Fig. 12), specifically the Heat Exchanger Outlet Temperature was identified as most important, making it trivial for the MBR system to identify the heat exchanger as the most likely faulty component, with only knowledge of which sensors are connected to which components.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

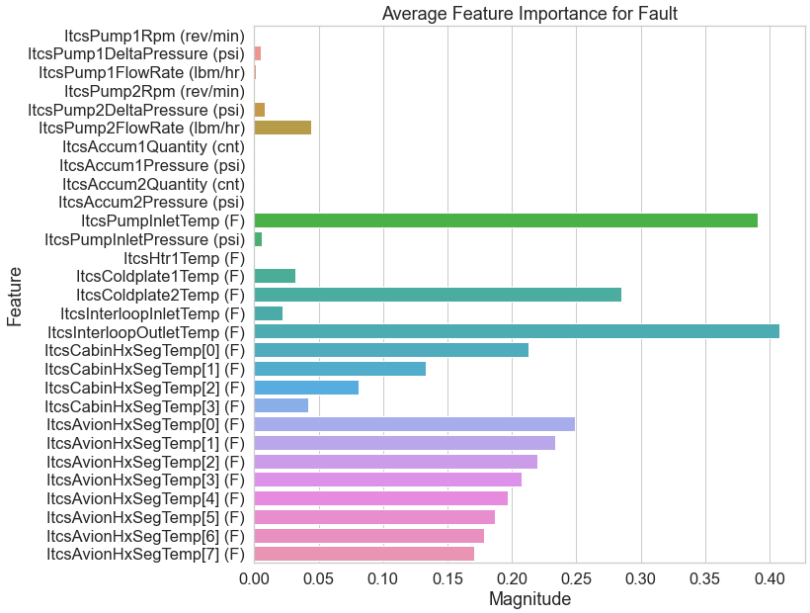

Fig. 12. Average feature importance for ITCS calculated over the fault test set.

The next experiment was to assess our machine learning approach's ability to discover the relationship between connected subsystems. We used data from the ITCS and the External Thermal Control System A (ETCS A) to train and evaluate our model. For this experiment we had the same setup as the previous experiment: a training set of nominal data, a test set of nominal data, and a test set of fault data. The specific fault for this experiment is a failure in the heat exchanger between the redundant External Thermal Control System B (ETCS B), and the rest of the Thermal Control System. None of the data used in this experiment is from the ETCS B, however, so that ETCS B represented an unknown phenomenon in this experiment since the failure in ETCS B will affect the relationship between ITCS and ETCS A in an unknown way. The model is trained on the nominal training set and evaluated against the two test sets. We find that it successfully detects all anomalies (See Table 6 and 7).

Table 6. SOM accuracy discriminating between nominal and anomalous ITCS & ETCS A data.

	Accuracy		
Dataset	**Nominal Training Set**	**Nominal Test Set**	**Fault Test Set**
SOM	98.75%	98.00%	100.0%

Table 7. SOM precision, recall and f1 score on both test sets.

	Precision	**Recall**	**F1 Score**
SOM	99.60%	100.0%	99.80%

This fault would manifest in the ITCS and ETCS A as out of pattern temperature readings. We find that our feature importance calculation successfully identifies temperature sensors in the ITCS and ETCS A as the most salient features (See Fig. 13), specifically the heat exchanger outlet temperatures are the most important, clearly allowing the MBR to identify something wrong (and unmodelled) with the one shared component of both subsystems.

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

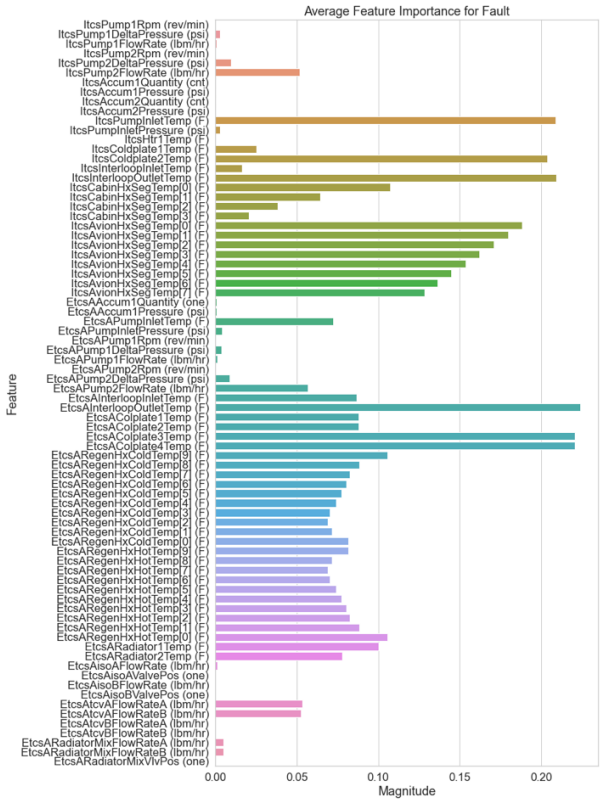

Fig. 13. Average feature importance for ITCS & ETCS A calculated over the fault test set.

And finally, we wanted to determine if our ML approach could learn relationships and then extrapolate them to unseen situations, specifically the SOM's ability to extrapolate delta temperature relationships. We used data from the ITCS and ETCS A to train and evaluate our model. For this experiment we had a training set of nominal data, and a test set of nominal data. The training set captures the system with a variety of different heat loads on and off, but never all at once. The test set captures the system only with all heat loads on at the same time. Note that this means that the situation in *all* of the test data (all loads on at once) is not represented in *any* of the training data. This also means that temperatures in the test data were generally higher than found in the training data. The model is trained on the nominal training set and evaluated against the nominal test set. We find that it successfully recognizes the nominal test set as nominal (See Table 8).

Table 8. SOM accuracy recognizing new nominal data.

	Accuracy	
Dataset	**Nominal Training Set**	**Nominal Test Set**
SOM	99.00%	98.00%

While this result is exciting, SOM generalizability beyond its training set is limited. Further analysis of this result is required. It does show a case where an MBR system monitoring temperature sensors, might indicate a problem if its thresholds for detection were too low. In this case, the ML system would correctly indicate that there was not a problem, lowering MAESTRO's certainty of a fault.

9. FUTURE WORK

We are currently pursuing several avenues to further develop this work. We have a NASA contract to apply MAESTRO, including both MBR and ML components to the Gateway EPS. As part of this effort we will also more tightly integrate the ML and MBR components and finish our integration of MAESTRO components with NASA's core Flight System (cFS). We are also in process of developing an experiment of some MAESTRO capabilities, fielded on and making improvements of MSU's radiation-tolerant processing using Field Programmable Gate Arrays (FPGAs) to be flown on the International Space Station (ISS) next year. These MAESTRO capabilities are

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

also being incorporated into an Astronaut agent for the next generation exploration spacesuit (xEMU, exploration EVA Mobility Unit).

10. CONCLUSIONS

In order to support the demands of future spacecraft missions, vehicles will need to be designed with a higher level of on-board autonomy. Autonomously handling faults requires a high-level of abstraction, closed-loop system consisting of intelligent component and system characterization; fault detection, diagnosis, reconfiguration, replanning, and rescheduling; and adaptive execution. Perhaps the most important component of this closed loop is fault detection and there are two common technologies used for this purpose, Model-Based Reasoning and Machine Learning. A hybrid system that combines both techniques effectively combines the advantages of both technologies and mitigates their weaknesses. The advantages of a hybrid MBR-ML detection and diagnosis system are that it:

- Utilizes a priori design knowledge when it exists and it is efficient to do so.
- Utilizes real or simulation sensor data when it exists and as it is produced.
- Utilizes very independently developed anomaly detection technologies for increased assurance.
- Can use its model to diagnose, reconfigure, recalculate, and replan in reaction to newly detected faults (whichever technology detected them) and can explain its reasoning.
- Can discover unknown or unmodelled relationships between sensor data in different modes. These relationships can span across subsystems.
- Can detect and diagnose anomalies never before encountered or trained for, using its MBR component and handle rare (but modelled) operating conditions.
- Can use the ML component to disambiguate overlapping MBR states.

The concept of the Hybrid MBR-ML fault detection and diagnosis system was validated in several experiments with real spacecraft hardware and with simulated spacecraft subsystems and the advantages of the combined approach verified.

11. ACKNOWLEDGMENTS

This work was performed under contracts awarded and administered by National Aeronautics and Space Administration Agency (NASA).

12. DISCLAIMERS

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA.

13. REFERENCES

[1]*Coalescing NASA's Views of Fault and Health Management*. Talk presented at the 2012 NASA FM Workshop.
[2] Robinson, P. (2013). *Fault Management (FM) as a Controller.* AIAA Infotech@Aerospace Conference, Boston, MA, 2013.
[3] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. In *Springer Series in Statistics*. https://doi.org/10.1007/978-0-387-84858-7
[4] Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics 43:59-69. doi: 10.1007/bf0033728

Copyright © 2020 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com