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

Dick Stottler, Sowmya Ramachandran, Chris Healy, Abhimanyu Singhal, Evan Finnigan
Stottler Henke Associates, Inc.

ABSTRACT

An important component of Space Situational Awareness (SSA) / Space Domain Awareness (SDA) is knowledge of the true status of friendly assets and whether any assets are under attack. Therefore, it is important to be able to detect faults and other anomalies, and determine the components involved and the root cause and whether that root cause is likely an external attack. Because these attacks may be both physical and/or by interfering with communications, it is also imperative that the satellite already have onboard an autonomous ability to, after diagnosis and root cause determination, determine the best method to recover mission capabilities, schedule the required recovery plan, and 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 systems can also be trained with known abnormal states, and recognize these more specifically when they occur.

This paper will describe progress on this work since our last paper presented at AMOS 2020. This includes further development and generalization of the hybrid approach to fault detection, diagnosis, and recovery; applying that approach to additional subsystems including a hardware model of the gateway electrical power system (EPS), to the exploration EVA Mobility Unit (xEMU) Portable Life Support System (PLSS) CO2 Removal and Thermal Management Subsystems, International Space Station (ISS) Urine Processing Assembly (UPA), and NASA Ames' Graywater Recycling System; and integrating components with NASA's core Flight System (cFS) and NASA JSC's IRIS architecture.

The additional subsystems brought new challenges to be overcome. For example, the xEMU PLSS CO2 Removal subsystem included complex cyclic behavior and states due to the use of twin amine beds that were each alternatively switched between exposure to the ventilation system to remove CO2 and exposure to the vacuum of space to off-gas the previously absorbed CO2, thus refreshing their absorbing capability. These challenges led to the development of a third, independent method for detecting anomalies, based on an analogy to thermodynamic variables: the Model-Agnostic Thermodynamic variable Anomaly Detection (MATAD) system performs automatic Characterization and Diagnosis of subsystem anomalies.

The hybridization emphasizes the benefits of each approach and mitigates the disadvantages. The benefits include the ability to detect and diagnose anomalies never before encountered; working well on Day One of in-space operations; effectively utilizing existing design knowledge; succeeding without large amounts of data; explaining the reasoning and being human understandable; being flight certifiable; behaving predictably; diagnosing down to the lowest modelled component level; handling rare but modeled operating conditions; execute very quickly; discover unknown and subtle relationships (even across subsystems); and provide extra certainty of the diagnosis when all three approaches agree.

1. FAULT DETECTION OVERVIEW

Fault Detection includes the three independent technologies for fault detection: MBR, SOMs, and MATAD. The Fault Detection module receives fault detection notifications from each of the three technologies, over time, and executes a Confirmation/Reconciliation procedure which considers each input (across both the three technologies and across different periods of time) and forwards the combined result to a Diagnosis module, which uses the given information and MBR to identify the specific faulty component and likely root causes, if possible, or a candidate set of culprits, otherwise. In the latter case, the Diagnosis module can often automatically narrow down this candidate

set, over time, to the one responsible faulty component. In addition to their usual role detecting faults, SOMs can also supply diagnostic information, to the degree that the current fault happens to be one of the known faults that the SOMs were trained for. Otherwise, the SOMs will simply identify the telemetry data as anomalous and indicate which features are most important for this determination, which can aid Diagnosis. Additionally, a Characterization module will be monitoring the behavior of components over time and updating its models and sending these updated models to the MBR modules to allow them greater precision in their determinations. Automatic planning, scheduling, and execution components determine a recovery plan, schedule the necessary actions, then adaptively execute the schedule.

2. MODEL-BASED REASONING (MBR)

One approach to detection, diagnosis, and characterization is Model-Based Reasoning (MBR). Model-based detection and diagnosis systems encode detailed and explicit descriptions of the interrelated factors that affect subsystem behavior. These models typically represent the world as a collection of components, where each component is characterized by attribute values and one or more modes. Constraints specify required relationships among attribute values and modes, and constraint violations are used to identify components in faulty modes. For example, current, voltage drop across a resistor, and resistance are constrained by an algebraic relationship. If the sensed voltage, current, and known resistance do not obey this relationship, then either one of the sensors or the resistor might be at fault. Similarly, there is a relationship between expected pressure increase across a pump or fan, its speed (e.g., in RPMs) and the flow rate. To the degree this relationship is violated, something may be wrong with one of the sensors or the pump or fan itself. The same is true with other components such as CO2 removal amine beds, heat exchangers, and specific chemical reactors. Model-based reasoning systems have traditionally been used to diagnose faults in engineered systems. System components are characterized by nominal and faulty modes, and constraints on interconnected components are based on physical laws and design intent. Because models encode the effects of contextual factors, 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. MBR engines can be extremely fast and do not necessarily require a large amount of memory or compute power, even for complex models.

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 some kind of fault. The model is then used to reason which component faults are most likely to lead to the currently deviating sensor values. The set of possible faults (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 fail and how (e.g., mechanical relays tend to fail open while solid state relays tend to fail closed), and/or are the most likely explanation for certain types of sensor values. MBR engines identify one specific fault and/or a set of possible faults. Note that the MBR engine does this every time new data arrives. However, as mentioned later, it may wait to issue a detection until other points in time or other modules are consulted.

3. SELF-ORGANIZING MAPS (SOMS)

Our Machine Learning approach to fault detection uses a combination of SOMs (Self-Organizing Maps) and Case-Based Reasoning to detect known and unknown faults, and it is extremely accurate at doing so. Because data is generally not (as) available for fault scenarios, it is expected that SOMs are generally trained on nominal (i.e., non-faulty) data. Then, given a new data point, the SOM will classify the point as nominal or anomalous. However, anomalous does not necessarily mean faulty. There is the obvious case of false positives, but there is an even deeper issue. Anomalous only means "different from the training data" – so if the SOM was not trained on data that reflected a similar enough operational setting to the current data sample, the SOM would classify the current data sample as anomalous. However, if the SOM was trained on data from all possible nominal scenarios, then an anomaly is likely a fault. In this case, the SOM will not indicate which type of fault, but merely that it is an (unknown) fault. Note that in this situation, the SOM **detects** but does not **diagnose** a fault.

On the other hand, sometimes enough data is available for one or more (but not all) fault scenarios. In this case, multiple SOMs will be trained – one SOM for the nominal case, and one SOM for **each** faulty case. For example, if we had training data for nominal data and two types of faults, we would train 3 SOMs total. For simplicity, we will call them nominal-SOM, faultType1-SOM, and faultType2-SOM. Now, each SOM is trained to classify "normal vs abnormal" **for its case** but what "normal" and "abnormal" mean for each SOM is different – "normal" refers to what

the SOM was trained on and "abnormal" refers to all else. Given a new data sample, nominal-SOM will classify it as nominal or not, faultType1-SOM will classify it as faultType1 or not, and faultType2-SOM will classify it as faultType2 or not. When a new data sample arrives, it is evaluated by each SOM, and one SOM is deemed to be the "match". If the match is nominal-SOM, then the data is deemed nominal. If the match is faultType1-SOM, then the data is deemed to be of faultType1. If the match is faultType2-SOM, then the data is deemed to be of faultType2.

The "match" is determined using **Case-Based Reasoning (CBR)**. CBR is an AI methodology that solves problems by retrieving solutions to previous similar problems and altering them to meet current needs. In this context, there is a nominal case, and one or more faulty cases. The core attribute of a case is a SOM. A case will include other pertinent information such as an explanation of the operational mode and, where necessary, a recommended set of actions to address or mitigate the situation. SOMs serve as the index into the case-bases for retrieving cases resembling the situation of interest. We use the Minimum Quantization Error (MQE) as the similarity metric for this purpose. CBR will first use the MQE measure to find the closest reference case. It will then compare the MQE from the target data to predefined upper and lower thresholds of the training data MQE for the reference case to test whether the two situations are similar. If the target MQEs fall outside the threshold range, this is a sign that there is a significant deviation between the two. This implies that no case can be found that matches the MQE for this target case (i.e., we have encountered a novel anomaly). On the other hand, if the target MQEs fall within the threshold range of the reference SOM's MQE, then the target and reference situations are similar enough for effective knowledge transfer. (To avoid confusion, we note that this CBR matching process is similar but different and distinct from the process that occur within a SOM which finds the closest matching SOM output node. SOM output node matching will be described in the SOM description section.) Note that for each SOM, the threshold can be adjusted (again, based primarily on the benefit/cost of True Positives, True Negatives, False Positives, and False Negatives). An improvement of this scheme will instead group data into batches of size N (an adjustable parameter), and have every SOM evaluate every sample in the batch. As mentioned before, this is to combat random noise/spikes. In this context, the number of consecutive abnormalities found, the fraction of the batch deemed anomalous, and/or an average MQE of the abnormal samples within a batch can be used instead of the MQE for one sample to find the match. Note that in this application of SOMs, where sufficient data is available for one or more fault types, the SOMs in combination with CBR do **detection** and **potential diagnosis** (depending on whether the SOMs+CBR think the supposedly faulty situation is a known fault or an unknown fault).

4. HYBRIDIZING MBR AND SOMS

Hybridizing MBR and SOMs emphasizes the benefits of each and minimizes the disadvantages. Generally, the benefits of MBR coincide with the disadvantages of SOMs, and vice versa, so the two complement each other well. In addition, because MBR and SOMs are very different technologies (especially in that one is model-based and one is model-free), their agreement provides extra confidence in the result (compared to using only MBR or only SOMs). A hybrid fault detection system uses both SOMs 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 SOMs detect it, or only the MBR system detects it. Another factor informing the resolution between MBR and SOMs is MATAD, which is described in greater detail in the next section. If the MATAD analysis of the sensor associated with the anomaly detected by MBR or by the SOMs indicates an abnormality, that is confirmation of MBR or of SOMs respectively.

5. MODEL-AGNOSTIC THERMODYNAMIC VARIABLE ANOMALY DETECTION (MATAD)

The central idea behind MATAD is that of a Thermodynamic Variable (TD). Note that in this context, a TD is a name for a mathematical object and **does not** necessarily refer to quantities such as temperature and pressure that are associated with thermodynamics specifically. This concept is inspired by thermodynamics but not limited to it. A TD is a function of the most recent n datapoints, with n varying for each TD. A simple example of a TD is the variance across the last 100 received datapoints. MATAD, across a series of training runs, learns the upper and lower limits for all these variables across all sensors. Values above these limits are marked as deviations, and the severity of the deviations are measured in terms of the standard deviation of the variables themselves. Examples of TDs include mean, mode, min, max, variance, and max jump between two samples. Note that every TD is defined over the last n samples. Therefore, one TD may be "mean over the last 100 samples" and another may be "mean over the last 500 samples." Because looking at multiple time scales can be helpful, MATAD will maintain multiple "versions" of each type of TD where each version corresponds to a different n. In addition to looking at the time-series data stream as-is, MATAD also performs a Fourier Transform (FT) of the stream, from which it extracts

additional TDs such as average frequency, min frequency, max frequency, peak frequency, and amplitude of peak frequency.

6. EXPERIMENTS/RESULTS

6.1 Mini Gateway Electrical Power System (EPS) Hardware

6.1.1 System Description

Montana State University (MSU) built a 1/1000 power scaled hardware setup that mimicked the Gateway Electrical Power System (EPS) for the Gateway Power Propulsion Element (PPE) and Habitation And Logistics Outpost (HALO), the high-level block diagram of which is shown below.

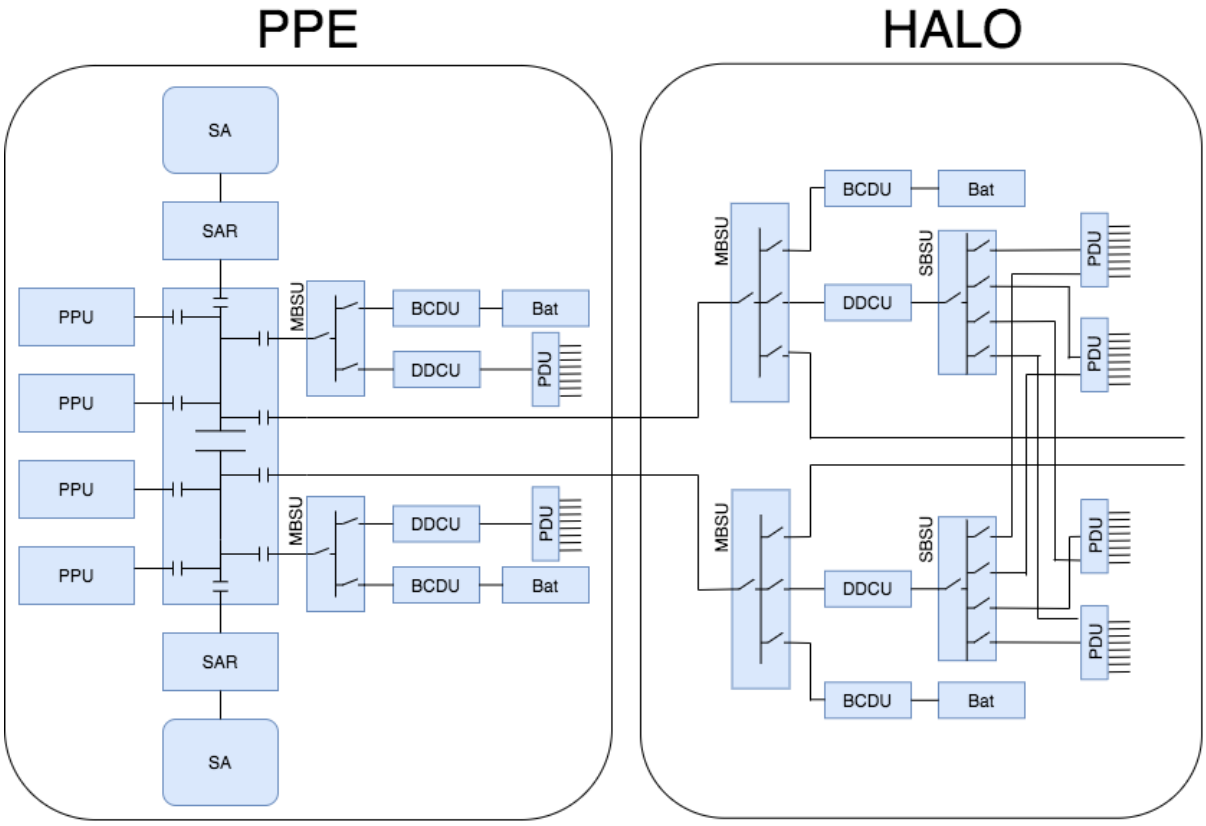

Fig. 1: Gateway EPS Block Diagram

MSU used four boards, roughly corresponding to the “top half” of the PPE (Board 1), the “bottom half” of the PPE (Board 2), the “top half” of the HALO (Board 3), and the “bottom half” of the HALO (Board 4), where the halves refer to the PPE+HALO image above. An incomplete schematic of Board 1 is shown below. It does not include the switches and loads branching off the PDU, and it excludes several sensors.

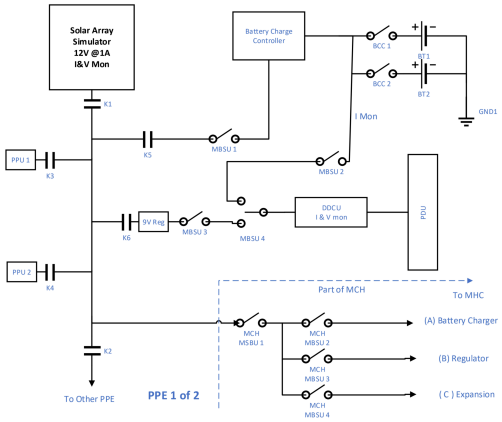

Fig. 2: Board 1

Board 1 and Board 2 are identical, and together mimic the PPE. A partial schematic of Board 3 is shown below. Similar to the schematic of Board 1, it excludes the switches and loads branching off the PDUs as well as several sensors.

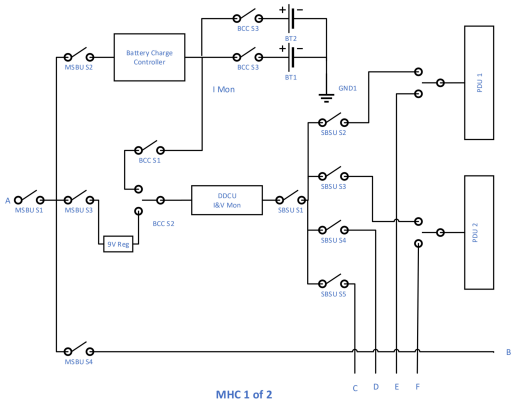

Fig. 3: Board 3

Board 3 and Board 4 are identical except for the loads branching off the PDUs, and together mimic the HALO (which we call the MHC here onwards). The four wires extending to the bottom of the schematic connect with Board 4.

The MBR component could diagnose:

- Full and partial solar array faults. The Solar Array provides up to 1 A at around 13 V. If the loads “demand” more than 1 A at 13V, the Solar Array still limits the current to 1 A and its voltage sags to less than 13 V. When the solar array is faulted, its current limit drops to below 1 A. For a full fault, the current limit drops to 0 A. There are also partial faults – where the current drops below normal but above 0.
- Battery faults. The battery is faulted when it is outputting no current.
- PPU faults. The PPU is faulted when it draws zero power (when it is on, and there is no solar eclipse).
- Broken switches. A switch is broken when its state does not match the commanded state. For example, if the switch is commanded to be open but the switch stays closed, this is a fault.
- Unresponsive loads. A load is unresponsive if its configuration (which we call “mode”) does not match its commanded configuration. For example, a comms load could be in Rx or Tx mode (and it draws much more power when in Tx mode).

- Single Event Upsets (SEUs). Single Event Upsets (SEUs) are caused by an ionizing particle hitting an electronic device. SEUs manifest as increased power draw in at least one mode of the load.
- High power faults. Some loads fail by drawing too much power.
- Low power faults. Some loads fail by drawing too little power.
- Zero power faults. Some loads fail by drawing zero power.
- DDCU trip faults. The DDCU switch is supposed to open when the power draw exceeds 5 W. A fault would be when the DDCU opens "early" when power draw is below 5 W.

There are dozens of significant components on the boards, such that the above fault types could manifest themselves in hundreds of different individual faults.

6.1.2 Experiments and Results

In addition to finding "planned for" faults (injectable faults that MSU designed into the hardware/software), our MBR module actually found truly unexpected/unknown faults.

In one case, MBR alerted us that both PPU 1 and PPU 2 were faulted in that they were drawing unexpected amounts of power (based on the commanded modes of the PPU). The Solar Array voltage and current were normal. MBR compared the expected power with the actual power. MBR found a fault that was not planned for (i.e., there was a known failure mode for the PPUs (0 W power draw), but MBR indicated that the PPUs were drawing non-zero unexpected power). Upon visual inspection of the boards, PPU 1 was physically touching PPU 2 (it was not designed this way). When then PPUs were separated, the MBR alerts disappeared. We assume someone bumped the boards, bending a wire.

In another case, MBR alerted us that the Solar Array voltage was very low (the actual voltage was around 8 V, and the expected voltage being around 13 V), even though very few/no loads were connected. Upon inspecting the boards, we found that a power supply had been off (accidentally). Upon turning the power supply back on, the MBR alerts disappeared.

For the Gateway EPS we conducted an experiment in which we trained a SOM on a limited/constrained nominal dataset and applied the SOM to detect DDCU cutoff faults and fully faulted solar array. We set the threshold for the MQE to be at the "99 percentile" meaning that the threshold was set such that on average 99% of the nominal data would be classified as nominal. The higher the percentile, the fewer false positives (high precision); the lower the percentile, the fewer false negatives (high recall). The results are shown in Table 1.

Table 1. SOM Gateway EPS Experiment Results

	Precision	**Recall**	**F1 Score**
DDCU Cutoff	92%	100.0%	95.8%
Fully Faulted Solar Array	81.8%	100%	90.0%

For the DDCU Cutoff, the SOM correctly identifies the unexpectedly toggled switches as the most important feature. For the Fully Faulted Solar Array faults, the SOM identifies the battery charger output current as the most important by a **narrow margin**, followed by the solar array current. This actually makes sense, as the battery charger output is always zero when the solar array current is zero, and it is some fraction of the solar array current (depending on whether the PPUs are on and on which loads are on) when the solar array current is nonzero.

6.2 Exploration EVA Mobility Unit (xEMU) Portable Life Support System (PLSS) CO2 Removal and Thermal Management Subsystems

6.2.1 xEMU PLSS System Description

The xEMU is a spacesuit, which is like a small spacecraft. The xEMU Portable Life Support System (PLSS) consists of the Primary Oxygen Supply, Secondary Oxygen Supply, CO2 Removal Loop, Primary Thermal Management System, and Secondary Thermal Management System. We have an in-house simulation of the CO2 Removal Loop, the Primary Thermal Management System, and the Primary Oxygen Supply. Below is a cross-section labeled diagram of the xEMU PLSS.

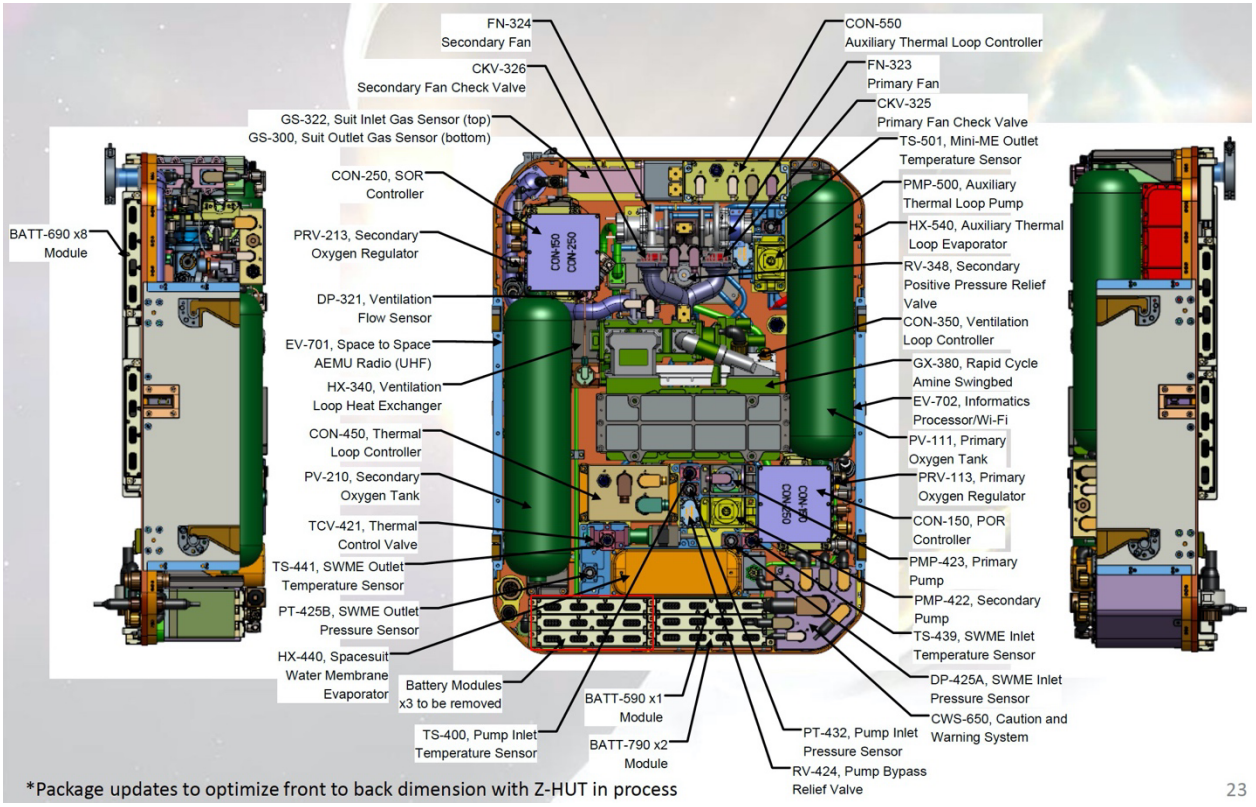

Fig. 4: xEMU PLSS Diagram

Below is a schematic of the xEMU PLSS:

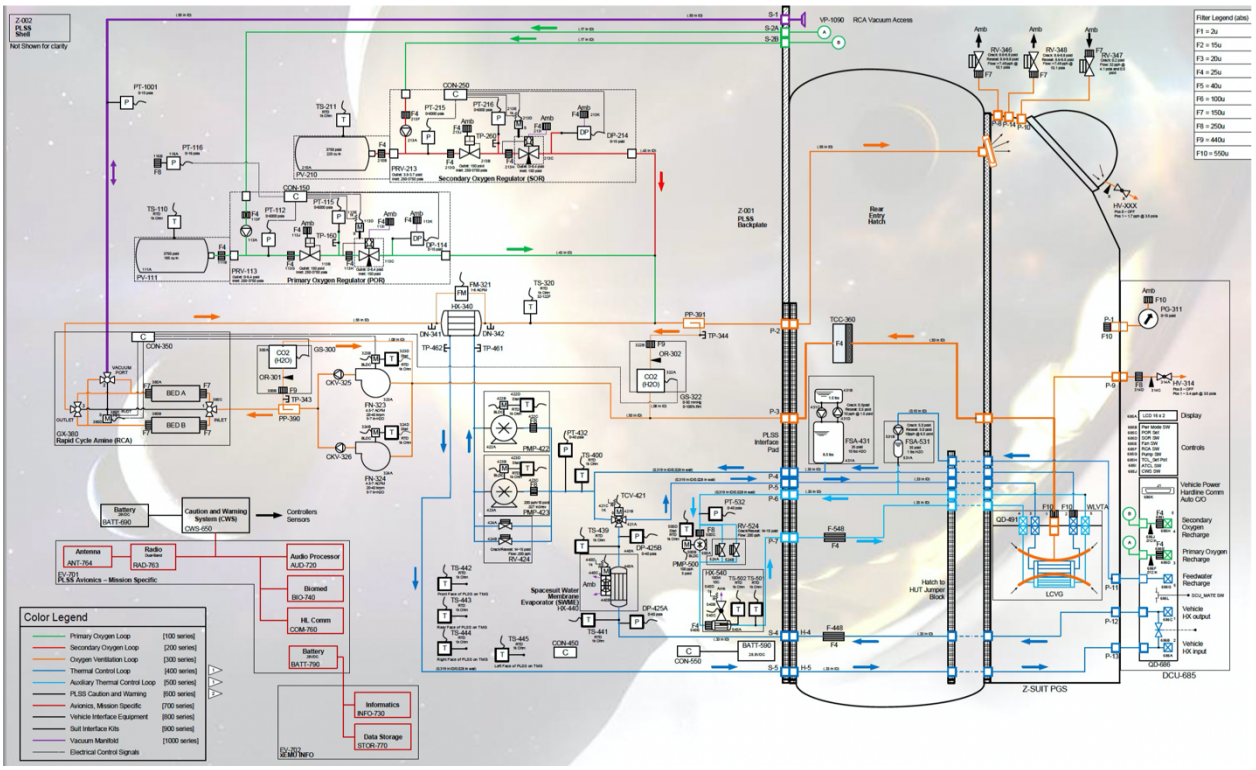

Fig. 5: Schematic of the xEMU PLSS

6.2.2 Experiments and Results

We simulated several faults including:

- Broken Fans. There are two fans that can push air through the CO2 Removal Loop. They can be broken (i.e., inefficient) slightly (e.g., operating at 99% of expected RPM) or significantly (e.g., operating at 50% of expected RPM).
- Inefficient Bed A. The two RCA Beds absorb CO2 from the air. They can, however, be operating at reduced efficiency. This can mean slightly reduced (e.g., reduced by 2%) or severely reduced (e.g., reduced by 50%). This fault makes Bed A inefficient.
- Inefficient Bed B. This is the same as the above fault but for Bed B.
- Jammed Bed. There is a bed valve that switches between Bed A and Bed B. Under normal operations, the valve first connects one bed, say, Bed A, to the ventilation loop in order to absorb CO2 from the air until it is saturated. Then, the valve switches so Bed B takes over the task of CO2 absorption. Meanwhile, Bed A off gases (releases its CO2 to space). When Bed B is saturated, the valve switches Bed A back. This process continues. A Jammed Bed means this valve is stuck (i.e., stops switching between Bed A and Bed B). This means the bed is stuck to the ventilation loop and will become saturated and stop absorbing CO2, which is disastrous (the astronaut would soon have to off gas manually using the helmet valve, which also means off gassing large quantities of oxygen along with the CO2 since they are mixed in the air, greatly shortening the EVA and requiring an EVA termination).
- Broken Sensors. This refers to sensors failing high, low, "frozen," and "random" (the four typical modes of failure).

MBR correctly finds Broken Fans, Jammed Bed, and Broken Sensors. MBR does not **yet** find the Inefficient Beds for a few reasons. First, inefficient beds are a "grayscale" fault – it does not make sense to consider a 99% efficient bed and a 0% efficient bed as both just "inefficient." One must set a threshold, which means trading off false positives and false negatives. To reduce false negatives, the threshold must be tighter. This brings us to the second point – the CO2 concentration is **highly variable** not only between astronauts but even for a single astronaut. This means too tight a threshold will have many false positives when in fact the deviation is due to the astronaut's varying metabolism, for example. This issue can be at least partially addressed with the **characterization** element of MBR. However, characterizing astronauts' CO2 levels was not yet implemented. Moreover, the MATAD method **does** correctly detect inefficient beds! A more accurate model would include auto-tuning for selected components.

We conducted an experiment with the SOMs where no fault data was used in training. The SOMs were tested on the above fault cases. When analyzing a specific data sample, SOMs do not directly consider neighboring samples (in time, e.g., the samples before or after the sample of interest). SOMs do evaluate in batches, but this currently consists of independently applying the SOM to each sample in the batch, and then taking an aggregate sum of how many samples were flagged nominal vs anomalous. Therefore, since there is no direct linkage in time, we **do not expect** the SOMs to detect anomalies for the Inefficient Bed A and Inefficient Bed B faults. However, the SOM **did** detect some of the inefficient beds. Note that the inefficient bed categories include significantly inefficient and slightly inefficient beds, and the slightly inefficient faults are harder to detect. Future work includes taking advantage of time in SOM-based analysis. Table 2 shows the results of the experiment:

Table 2. SOM results for xEMU PLSS

Fault Set	Accuracy	F1-Score	Precision	Recall	Top 2 Sensors Most Correlated with Anomaly
Broken Fans	98	98.15	97.7	100	324B
Inefficient Bed A	57.8	68.06	94.1	53.3	PP390 PP391
Inefficient Bed B	76.3	84.29	95.7	75.3	PP390 PP391
Jammed Bed	85.6	90.9	96.0	86.3	DP114 PT112
Broken Sensors	87.5	93.0	98.7	88.0	PP390 PP391

The MATAD approach finds all the above faults all the time, with no false negatives. It is very impressive.

In addition to the above faults, we also tested MBR on the following faults in the Primary Thermal Management System:

- Broken Pump. Broken Pumps are similar to Broken Fans in that they can be broken (i.e., inefficient) slightly or significantly.
- Jammed SWME. The SWME regulates the water temperature (keeps it cool). The SWME exposes a small portion of the water to vacuum. Some of the water is vaporized, which draws heat from the water that is not vaporized, causing the temperature of the remaining unvaporized water to drop. The amount of water that is vaporized (and therefore the resulting temperature drop) is controlled by the size of the SWME aperture using closed-loop temperature control. Under normal operations, if the incoming water temperature is higher, the aperture is bigger, and if the incoming water temperature is lower, the aperture is smaller. If the SWME is jammed, the aperture stays constant, resulting in a constant temperature drop (regardless of incoming temperature).
- Broken Sensors. This is exactly the same as the four sensor failure modes for the CO2 Removal Loop.

MBR successfully finds all these faults. Similarly, MATAD also finds all of these faults all the time, with no false negatives.

6.3 International Space Station (ISS) Urine Processing Assembly (UPA)

We were provided with data from the Urine Processing Assembly (UPA) subsystem of the International Space Station (ISS). The data was captured over a period of time that included multiple faults occurring at different times and affected a range of components. We were provided with a list of anomalous events and the dates on which they occurred. Figure 6 shows the form in which this information was provided. Note that this is not complete ground truth information as we were only told which dates the faults occurred and not the precise time segments where they happened. Nor did the list specify that there were no other faults or anomalies on other days. We analyzed the data using ADTM to see if it could detect the anomalies mentioned. We were given nominal data that we could use to train SOMs and analyze the rest of the data to look for the known faults.

	GMT year	GMT day	
	2015	7	Last on PCPA SN002, tube rupture
		8	First on PCPA SN003
		292	FCPA SN007, high motor current, harmonic drive design
		300	First on FCPA SN004 (first planetary drive design)
		300a	few attempts necessary to get motor calibrations correct
	2016	(April 2016 to Oct 2017)	UPA operating in degraded state with DA leaking pretreated urine into distillate
		297	Last on PCPA SN003, tube rupture
		303	First on PCPA SN006
	2018	87	PCPA SN006, high motor current (legacy, harmonic drive)
		90	PCPA SN004R (first planetary design install)
		151	FCPA SN004, high motor current (planetary drive design), UPA to STOP, severe alert
		151a	two restart attempts with no joy
		153	first on FCPA SN005R

Fig. 6: UPA Data Format

Table 3. The set of fault conditions and experiments for the UPA data

Experiment	Condition
1	PCPA Tube Rupture
2	Anomalous P4 Swings
2.5	High Motor Current (PCPA)
3	High Motor Current (FCPA)

We were given data that captured the different conditions in the above table. We regarded each one as an experiment with ADTM analysis. For each experiment, we used ADTM to train a SOM on data from a day that did not have any faults indicated. Note that the data provider did not explicitly say that lack of association of any data segment with a

fault in their notes necessarily means the data segment is nominal. However, we forced to make that assumption in order to have training data. Our assumption was that even if the training data contained anomalous segments, they would not be representative of the target faults. Having trained a SOM, we then use it to analyze the test data (the data segments with known faults) and look for anomaly flags.

Lacking precise and complete ground truth data, we were unable to use a principled search for optimal hyperparameters and training time. We used a standard SOM size of 32 by 32 neurons. We trained the networks for 30 epochs.

The data was partitioned into months. For each known fault, we used the data for the month as the test data. For example, the data annotations indicated that a known fault occurred on GMT days 7 and 8 in 2015. Therefore, we used the data from January 2015 as the test data set for this fault.

Considering the anomaly flags raised by ADTM, we observed many false positives (where a false positive is any anomaly that does not fall within the approximate time segments with known faults). Many of the false positives were of extremely short duration, indicating a high amount of noise in the anomaly detection. To mitigate this issue, we smoothed the anomaly flags by averaging over a moving window. We used a window size of 600 (representing 10 mins of real-time data). Finally, a hyperparameter that had a significant impact on the results was the MQE (minimum quantization error) threshold used for detecting anomalies. The default is to compute the MQE for the training data and use the 99-percentile value as the threshold. We found that this tended to produce many false positives for some of the experiments. Using a 100-percentile (i.e., maximum value) as the threshold mitigated this effect. However, for other experiments, using this more stringent criteria led to the known anomalies not being detected. We therefore adjusted the threshold in these cases to the 99-percentile value. Ideally, we would pick an optimal threshold value using a hyperparameter search approach using completely labeled data. As mentioned earlier, that was not possible for this data set.

6.3.1 PCPA Tube Rupture

For this data set we used an MQE threshold of 100-percentile value of the training MQE. The following figures show a plot of the anomaly flags raised by ADTM versus the ground truth. The blue plot shows ADTM's flags with a 1 indicating an anomaly. The orange plot shows the ground truth, again with a value of 1 representing an anomaly. We generated "ground truth" data by marking all points in the date range associated with a fault as anomalous. Note that this is a coarse approximation of the ground truth for a few reasons:

1. It could be that some fault events lead to system-wide shut down or other adaptive behavior from the system to restore near normality. ADTM may not consider those behaviors faults per se because of these behaviors being represented in the training data.

2. The fault may be intermittent, appearing instantaneously or for a short duration during the noted time segment. We made no assumptions about the duration of faults while creating the "ground truth" labels.

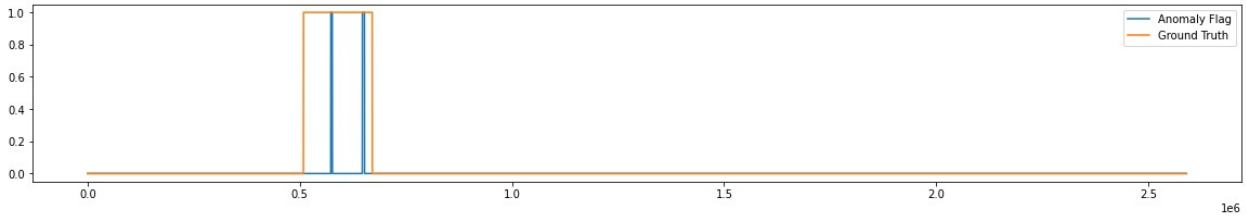

Fig. 7: Anomalies detected for January 2015 PCPA tube rupture event

Figure 7 shows the results for the PCPA tube rupture event in January 2015. ADTM is able to detect anomalies in the period of the event and it detects no others. Note that the faults detected by ADTM do not span the entire duration. However, as mentioned earlier, it is not known if the faulty behavior spans the entire duration. It is encouraging that ADTM can detect a faulty event and alert the user to take necessary actions.

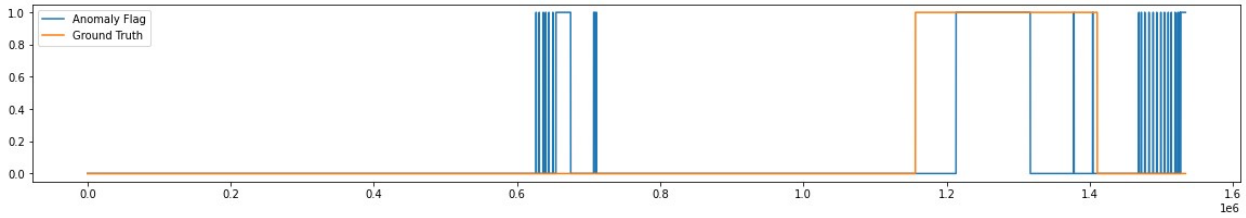

Fig. 8. Anomalies detected for October 2016 PCPA tube rupture event

Figure 8 shows the results for the same fault in October 2016. Here, ADTM flags anomalies for a large portion of the time segment where the fault was active. It also identifies anomalies at other times. These may be false positives.

6.3.2 High P4 Swings event

For this experiment we used an MQE threshold of 100-percentile value of the training MQE. Note how ADTM is unable to detect the anomaly that started on GMT Day 22. It could not detect this anomaly even after lowering the MQE threshold to a 90-percentile value of the training MQE. This is the only time that ADTM could not detect an anomaly (i.e., produced a false negative outcome). It does successfully flag anomalies on the other days with known faults. However, it does raise alarm on several other days which could potentially be false positives.

Figure 9 shows the results of the analysis by ADTM. It does flag anomalies but only towards the latter part of the time segment indicated by the SME (closer to Day 30). Thus, it is unable to detect this fault closer to when it arises. There are no false positives for this period.

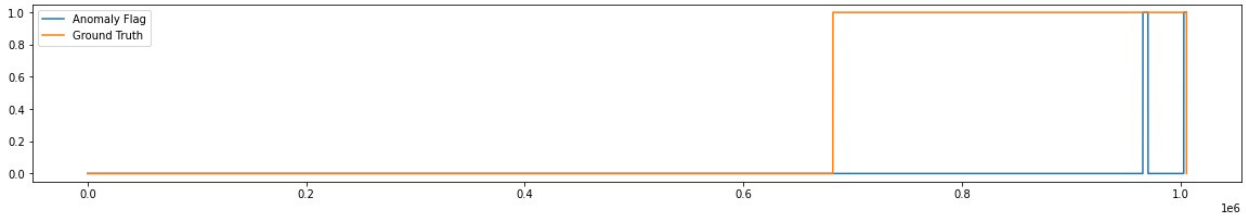

Fig. 9: Anomalies detected for January 2020 High P4 Swings event

The High P4 Swing event started in January 2020 and continued into February. The figure below shows the ADTM results for February 2020. Again, ADTM is able to detect anomalies during the period the fault was active but not for the entire duration and, again, it is not clear if these are instances of false negatives or if indeed the fault was intermittent. Additionally, ADTM detects anomalies after the fault was resolved. We were not able to verify if these are instances of false positives.

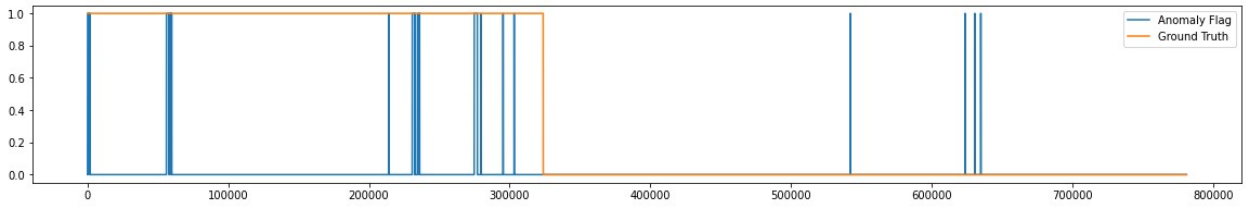

Fig. 10: Anomalies detected for February 2020 High P4 Swings event

6.3.3 High PCPA Motor Current

For this experiment we used an MQE threshold of 99-percentile value of the training MQE. Using a 100-percentile value led to false negatives. With the 99-percentile threshold, ADTM was able to flag anomalies on the days with known faults.

Figure 11shows the results of ADTM's analysis. As before, ADTM detects anomalies in the time period of interest. However, it does not cover the entire period. We expect that a high current fault will be intermittent, but we do not have ground truth information on the timing of these faults within that period. ADTM also flags some potential false positives. This is a case where we used a less stringent MQE threshold (99-percentile vs maximum) to help ADTM

detect true positives, but this led to an increase in false positives. Enhancements to the approach that improve sensitivity without sacrificing precision is a desirable direction for future work.

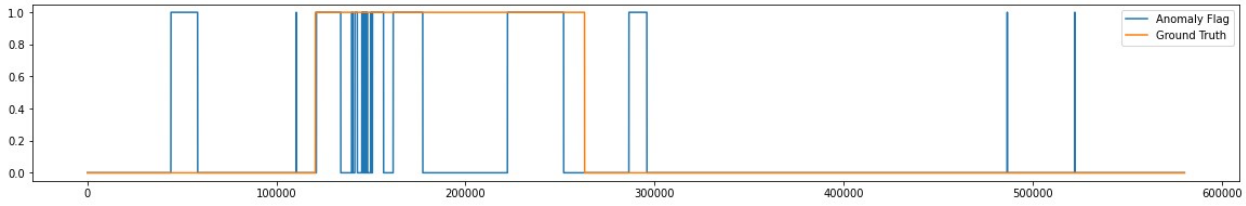

Fig. 11: Anomalies detected for March 2020 PCPA High Motor Current event

6.3.4 High FCPA Motor Current

For this experiment we used an MQE threshold of 99-percentile value of the training MQE. Figure 12 shows that ADTM successfully flagged anomalies in the time period of the fault in 2018. It also found many intermittent anomalies that are possibly false positives.

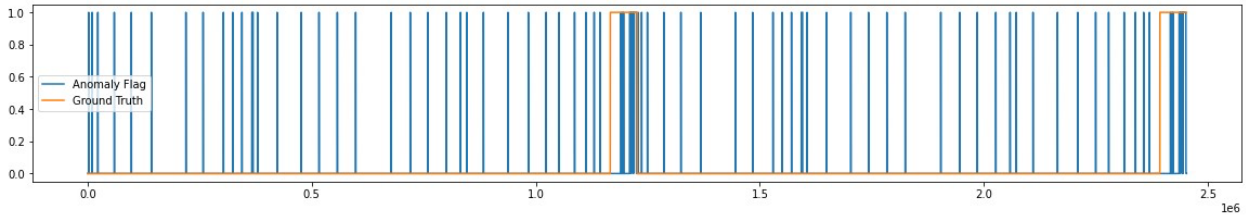

Fig. 12: Anomalies detected for May 2018 FCPA High Motor Current event

Figure 13 shows that ADTM successfully flagged anomalies in the time period of the fault in 2015. It also found many intermittent anomalies that are possibly false positives.

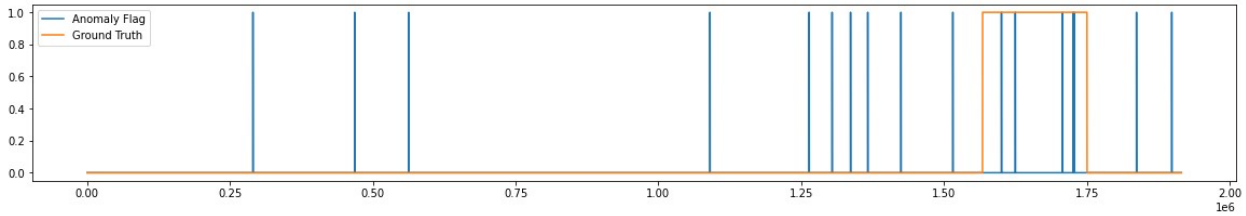

Fig. 13: 2015 Anomalies Flagged

A theme across all experiments is that the SOMs can successfully detect anomalies when they arise, except in the case of the High P4 Swing fault. However, it detects anomalies intermittently, i.e., in most cases, the anomalous segment identified by ADTM is a subset of that identified based on the SME provided information. It is not clear how much of that should be classified as false negatives since we do not have information on the nature of the faults (whether they were intermittent or if they lasted the entire duration). ADTM also identifies anomalies outside of the SME-provided time periods. Again, it is not clear how many of these instances should be classified as false positives because the SME mentioned that there might be abnormalities beyond what she has noted. Nonetheless, this can be taken as evidence that ADTM has the potential for raising false alarms. This is a good example of how a hybrid system that utilized MBR would very likely suppress these false alarms.

6.4 NASA Ames' Graywater Recycling System

We achieved the following significant results through the NASA ARC Graywater Recycling System experiment:
1. Our SOMs accurately detected a known fault in a real-world NASA ARC Graywater Recycling System dataset.
2. Our SOMs accurately localized the known fault to two salient measurands that we identified as contributing the most to deviation in the fault test sets. These measurands were confirmed by our NASA ARC point of contact and SME Michael Haddad as important sensors to detect as anomalous, given the nature of the fault.
3. Our SOMs accurately detected and localized the fault across two interconnected subsystems, one downstream of the other, through training two SOMs on separate subsystems within the Graywater Recycling System. This is a

critical step towards successfully cross-correlating faults that have cascading effects from one subsystem to another and proves the feasibility of our design in which we intend to train many SOMs of varied granularity on distinct subsystems and components and then aggregate their results to show how measurands across subsystems are affected by a fault or degradation pattern over time. This is critical to developing long-term monitoring capabilities that track the evolution of faults or component deterioration over time.

We acquired data from NASA engineers in October 2018 on the Graywater Recycling System installed at Stanford University. The analysis we ran on such data advanced our research significantly as the data documented a real system failure. Dr. Michael Flynn, Principal Investigator in water recycling technology development at NASA ARC, described the failure in an email:

> *"The data files are interesting because they document a system failure. During the run the FO membranes became fouled with bacteria and the system shut down due to a low OA tank float alarm. Water was added to the OA tank and the system was restarted but then shut down again.*
>
> *This is probably the most common problem we have encountered. In this case it occurred because the FO membranes became fouled by a bacterial sludge. It is an indication that the FO membranes need to be back flushed. It seems to occur every 3 to 4 weeks. It would be an important event to predict."*

6.4.1 Graywater Recycling System Description

Figures 14 – 16 display the Graywater Recycling System schematics provided by Dr. Flynn. Figure 5 is the subsystem that contained the fault; the FO membranes and OA tank implicated in the anomaly are highlighted by green boxes. Beyond detecting the fault, our algorithm was also able to output the specific sensors that contributed most to the anomaly, as validated by our SME.

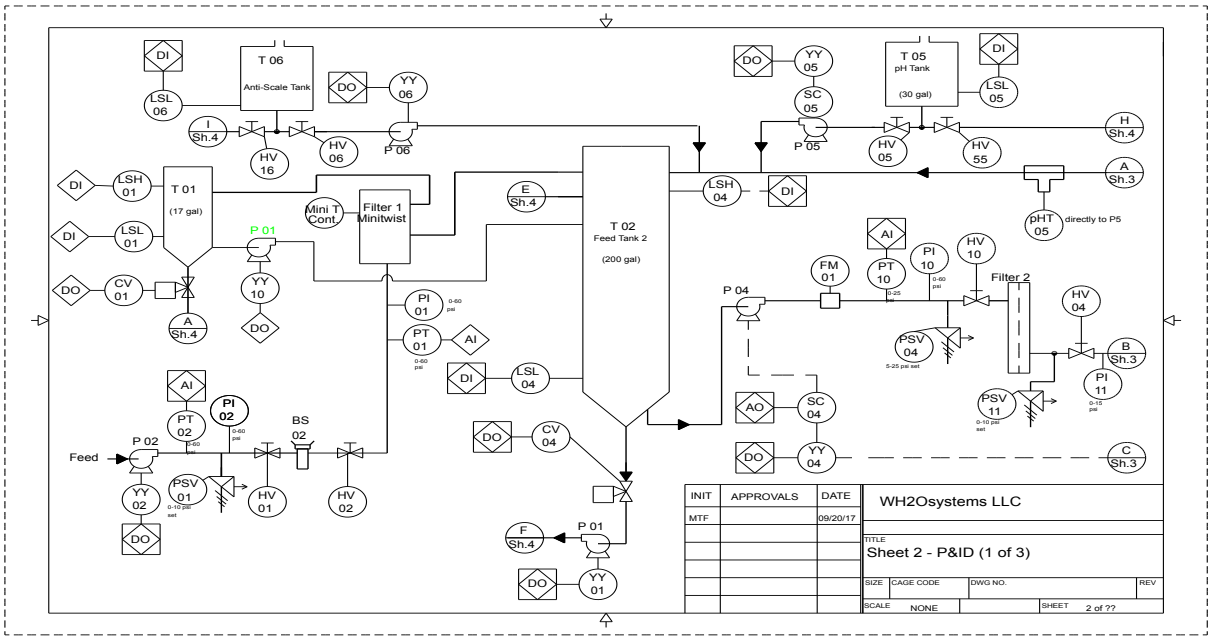

Fig. 14. Stanford Graywater Recycling Schematic 1/3

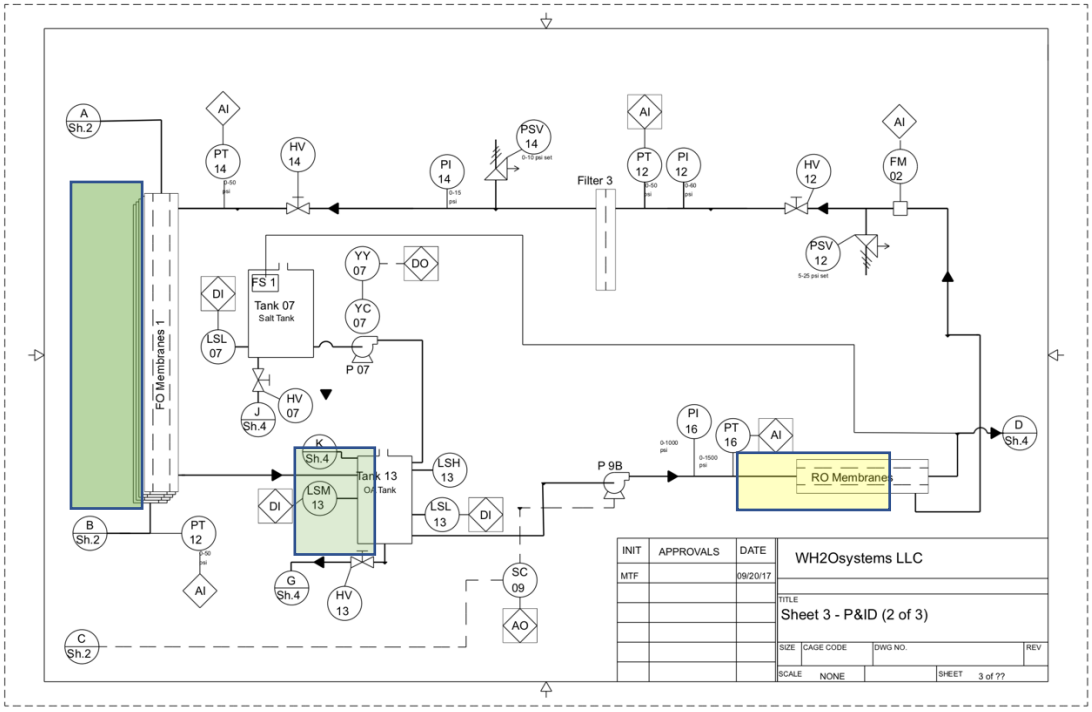

Fig. 15. Stanford Graywater Recycling Schematic 2/3
FO Membrane 1 and OA Tank implicated in fault, in green. RO Membranes affected by fault downstream, in yellow.

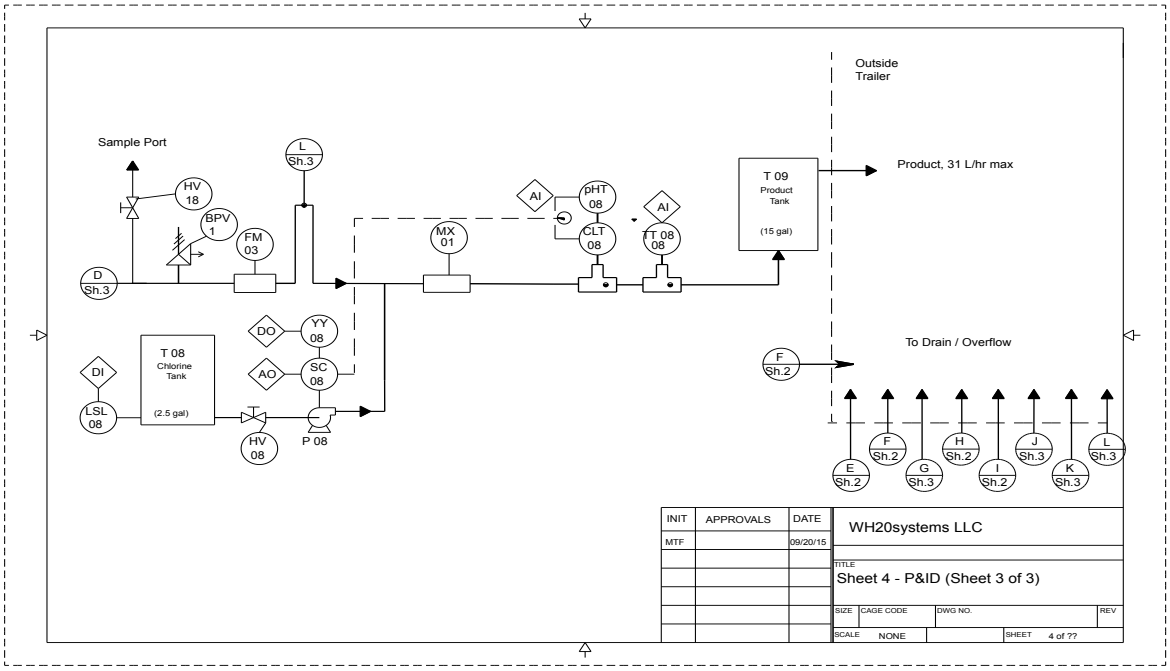

Fig. 16. Stanford Graywater Recycling Schematic 3/3

The data we received decomposed the Graywater Recycling System into two subsystems (which are defined by sensors spread out across the schematics in Figures 14 – 16), which we refer to as “Subsystem 1” and “Subsystem 2.” Their respective features are listed in Table 4.

Table 4. Graywater Recycling Subsystem Sensors

Subsystem	Sensors
Subsystem 1	- TURN SYSTEM ON/OFF INDICATOR - CV01, CV04 - P01, P02, P03, P04, P05, P06, P07, P08, P09 - PUMP4 SPEED SET SCALED - RO PUMP SPEED SET SCALED - PT01, PT02, PT10, PT12, PT14, PT16 SCALED - LSH01, LSH04, LSH13 - LSL01, LSL04, LSL05, LSL06, LSL07, LSL08, LSL13 - LSM13 - PT11 SCALED FP, PT14 SCALED FP

6.4.2 Graywater Recycling System Experiments and Results

For each subsystem, we received two days' worth of nominal data and four days of faulty data. We divided the nominal data into a training set and a "nominal test set" for each subsystem, the latter of which was used to compare against the fault test sets.

For each test set, our SOMs flagged a sample point as anomalous based on its MQE score, using the 99th percentile of the training MQEs as a threshold. The results are displayed in Table 5. Note, the percentage of anomalies detected in the "fault test set" represents the percentage of True Positives with respect to the known fault. The percentage of anomalies in the "nominal test set," however, do not necessarily correspond with the rate of False Positives, as the nominal test set for Subsystem 1 was implicated in novel nominal activity that was not captured during training; thus it contained unknown anomalies that our algorithm successfully detected, though these were not necessarily faults.

Table 5. % Anomalies detected in Graywater Recycling System data with 99% confidence interval

Subsystem	Test Dataset	Percentage Anomalies Detected
Subsystem 1	A. nominal test set B. fault test set	A. 12.2% B. 99.86%
Subsystem 2	A. nominal test set B. fault test set	A. 0% B. 84%

From Table 5, we see that the SOMs flagged >99% and 84% in the fault test sets of Subsystem 1 and Subsystem 2 respectively. By comparison, the SOM trained on Subsystem 1 flagged ~12% of the nominal test set as anomalous, while the SOM trained on Subsystem 2 detected no anomalies in the nominal Subsystem 2 test set.

Combining the ML with MBR in a hybrid system should largely or completely eliminate the false alarms for Subsystem 1.

7. IRIS INTEGRATION

The Intelligent Response and Interaction System (IRIS) is a JSC architecture which itself is intended for use in several different spacecraft missions, including the Gateway. IRIS uses the MQTT message bus to share data and we have integrated some of our xEMU PLSS modules, including MBR Diagnosis, with JSC's MQTT bus. This was simply a matter of publishing messages to their MQTT bus and subscribing to messages from it.

8. core Flight System (cFS) Integration

One of NASA's previous and currently ongoing investments leveraged in this proposed effort is cFS. "The core Flight System (cFS) is a platform and project independent reusable software framework and set of reusable software applications. There are three key aspects to the cFS architecture: a dynamic run-time environment, layered software, and a component-based design. It is the combination of these key aspects that makes it suitable for reuse on any number of NASA flight projects and/or embedded software systems at a significant cost savings." (https://cfs.gsfc.nasa. gov/). As mentioned above, this Diagnosis research is part of an autonomous, closed-loop system that detects faults, diagnoses them, determines the root cause and the best method to recover mission capabilities, schedules the required recovery plan, and adaptively executes it, called MAESTRO. NASA's Glenn Research Center requested that we integrate the EPS Scheduling component of MAESTRO with cFS. This involved publishing messages and subscribing to cFS's Software Bus (SB). The scheduler and its integration were successfully tested with GRC's systems (which were already integrated with cFS)

9. CURRENT/FUTURE WORK

We have recently begun an effort to operationalize and apply this work to a network of ground-based satellite antennas. Initially we are focused on a pilot project to develop the technology for one specific, high-value, frequently-problematic component. Once the technology is proven in the pilot, we will extend it to all components. We also plan to integrate the complete set of diagnostic components with JSC's IRIS architecture and generalize our modules as needed. JSC has enthusiastically supported this plan. We expect to apply these technologies to several other applications as the opportunities arise.

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 (from MBR).
- Utilizes real or simulation sensor data when it exists and as it is produced (from ML).
- 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 (from MBR).
- Can discover unknown or unmodelled relationships between sensor data in different modes. These relationships can span across subsystems (from ML).
- 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. Furthermore, though preliminary, the Model-Agnostic Thermodynamic variable Anomaly Detection (MATAD) system shows great promise as a third independent anomaly detector.

11. REFERENCES

[1] R Stottler, S. Ramachandran, C. Belardi, R. Mandayam. On-board, autonomous, hybrid spacecraft subsystem fault and anomaly detection, diagnosis, and recovery, *Ica AMOS 2019 Proceedings*, 2020.