

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/238108187>

General Purpose Data-Driven System Monitoring for Space Operations

ARTICLE in JOURNAL OF AEROSPACE COMPUTING INFORMATION AND COMMUNICATION · OCTOBER 2012

DOI: 10.2514/1.54964

CITATIONS

11

READS

36

7 AUTHORS, INCLUDING:



David L. Iverson

NASA

37 PUBLICATIONS 158 CITATIONS

SEE PROFILE



Rodney Martin

NASA

32 PUBLICATIONS 115 CITATIONS

SEE PROFILE



Mark Schwabacher

Google Inc.

45 PUBLICATIONS 980 CITATIONS

SEE PROFILE



Lilly Spirkovska

NASA

39 PUBLICATIONS 301 CITATIONS

SEE PROFILE

General Purpose Data-Driven System Monitoring for Space Operations

David L. Iverson^{*}, Rodney Martin[†], Mark Schwabacher[‡], Lilly Spirkovska[§], William Taylor^{**}
NASA Ames Research Center, Moffett Field, California, 94035

Ryan Mackey^{††}
California Institute of Technology, Jet Propulsion Laboratory, Pasadena, California, 91109

J. Patrick Castle^{‡‡}
Stinger Ghaffarian Technologies, NASA Ames Research Center, Moffett Field, California, 94035

Modern space propulsion and exploration system designs are becoming increasingly sophisticated and complex. Determining the health state of these systems using traditional methods is becoming more difficult as the number of sensors and component interactions grows. Data-driven monitoring techniques have been developed to address these issues by analyzing system operations data to automatically characterize normal system behavior. The Inductive Monitoring System (IMS) is a data-driven system health monitoring software tool that has been successfully applied to several aerospace applications. IMS uses a data mining technique called clustering to analyze archived system data and characterize normal interactions between parameters. This characterization, or model, of nominal operation is stored in a knowledge base that can be used for real-time system monitoring or for analysis of archived events. Ongoing and developing IMS space operations applications include International Space Station flight control, satellite vehicle system health management, launch vehicle ground operations, and fleet supportability. As a common thread of discussion this paper will employ the evolution of the IMS data-driven technique as related to several Integrated Systems Health Management (ISHM) elements. Thematically, the projects listed will be used as case studies. The maturation of IMS via projects where it has been deployed, or is currently being integrated to aid in fault detection will be described. The paper will also explain how IMS can be used to complement a suite of other ISHM tools, providing initial fault detection support for diagnosis and recovery.

I. Introduction

AS modern space propulsion and exploration systems improve in capability and efficiency, their designs are becoming increasingly sophisticated and complex. Determining the health state of these systems using traditional parameter limit checking, model-based, or rule-based methods is becoming more difficult as the number of sensors and component interactions grows. Data-driven monitoring techniques have been developed to address these issues by analyzing system operations data to automatically characterize normal system behavior. System health can be monitored by comparing real-time operating data with these nominal characterizations, providing detection of anomalous data signatures indicative of system faults or failures.

^{*} Computer Engineer, Intelligent Systems Division, Mail Stop 269-4

[†] Computer Engineer, Intelligent Systems Division, Mail Stop 269-1

[‡] Computer Scientist, Intelligent Systems Division, Mail Stop 269-3

[§] Computer Engineer, Intelligent Systems Division, Mail Stop 269-3

^{**} Computer Engineer, Intelligent Systems Division, Mail Stop 269-2

^{††} Senior Researcher, 4800 Oak Grove Drive, Mail Stop 126-147

^{‡‡} Computer Scientist, Intelligent Systems Division, Mail Stop 269-3

Data-driven techniques have a number of advantages over other methods for monitoring complex space vehicles. Unlike model-based systems, the developer does not need to understand or encode the internal operation of the system. The knowledge required to monitor the system is automatically derived from archived data collected during system operation. Unlike rule-based systems, data-driven systems do not require system analysts to define nominal relationships among sensors. Analysts can and often do determine these relationships for systems with few sensors; it is more difficult to analytically determine the nominal relationship among a large number of sensors. Data-driven techniques are not limited to low-dimensional spaces and can work as effectively with dozens of parameters as they do with a few. Knowledge bases formed by data-driven techniques are also easy to update. As the operating envelope of the monitored system is expanded, data-driven techniques can be quickly retrained to incorporate the new behavior into the knowledge base. The expertise and time-consuming process of updating a model or rule base to maintain consistency with the new operation is not required. Furthermore, as the training data set is updated with additional operations data, the resulting nominal behavior models tend toward improved system characterization.

Several data driven software tools have been successfully applied to aerospace operations for both real-time system monitoring and archived data analysis¹. One such tool, Inductive Monitoring System² (IMS), uses a data mining technique called clustering to analyze archived system data and characterize nominal interactions between selected parameters. This characterization, or model, of normal operation is stored in a knowledge base that can be used for real-time system monitoring or analysis of archived events. In monitoring, system data is periodically compared with the nominal IMS model to produce a measure of how well current system behavior matches normal behavior defined by the training data used to construct the knowledge base. The degree of deviation from expected nominal system behavior is summarized with a single scalar “distance from nominal” value for each tested data sample. Significant deviations from the nominal system model can provide alerts to call attention to potential system malfunctions or precursors of significant failures. IMS also provides information pertaining to the relative amount of contribution from each monitored parameter to any detected deviations. This can be helpful in isolating the cause of the anomaly.

This paper discusses the use of the IMS data-driven technique as related to several Integrated System Health Management (ISHM) projects. The scope of IMS-based monitoring applications continues to expand with current development activities. Successful IMS deployment in the International Space Station (ISS) flight control room has led to generalization and applications in other ISS flight control disciplines. It has also generated interest in data-driven monitoring capability for Constellation, NASA’s program to replace the Space Shuttle with new launch vehicles and spacecraft capable of returning astronauts to the Moon, and then on to Mars. Several projects are currently underway to evaluate and mature the IMS technology and complementary tools for use in the Constellation program. These include a vehicle system management experiment for the Air Force TacSat-3 satellite, ground systems monitoring for NASA’s Ares I-X and Ares I launch vehicles, and fleet supportability applications. This paper describes the maturation of IMS via these projects and also explains how IMS can be used to complement a suite of other ISHM tools, providing initial fault detection support for diagnosis and recovery.

II. Data Mining for Space Operations

Many space operations organizations maintain extensive archives of operational data from both spacecraft and ground support systems. Data mining methods can be used to analyze the data found in these archives and extract information about typical parameter behavior and parameter interactions. In particular, data driven anomaly detection techniques can process the data to find unusual events, or outliers, in data for a given subsystem. These anomaly detection techniques can also automatically analyze archived nominal system data to characterize normal system performance. Comparing incoming real-time data to a nominal model can inform the user when current system behavior differs from previous system performance.

A. Distance-Based Anomaly Detection

One powerful feature of many data driven anomaly detection techniques is the ability to simultaneously analyze multiple parameters. This feature allows them to discover and model interactions between related parameters that might be difficult to notice when monitoring the parameters individually. A basic data structure used for distance-based analysis is a vector of parameter values (Fig. 1). Vectors containing N

values are treated as points in an N -dimensional vector space. An appropriate distance metric is used to calculate the distance between these points. The familiar Euclidean distance metric has proven effective in many applications, though other metrics may also be useful. The underlying premise of distance-based anomaly detection is that anomalous data points will fall a significant distance away from typical, nominal data points.

<i>Pressure A</i>	<i>Valve 1 Position</i>	<i>Pressure B</i>	<i>Valve 2 Position</i>	<i>Pressure C</i>	<i>Temperature 1</i>	<i>Temperature 2</i>
2857.2	86.4%	1218.4	96.2%	1104.1	49.8	37.6

Figure 1. Sample Data Vector

For system health monitoring applications, vector parameters are typically instantiated with concurrent sensor values collected from a time slice of the data stream. Additional computed (derived) values or parameter values from previous data samples can be included in the vector as well. For instance, increased system insight can often be obtained by incorporating values in the vector such as the rate of change of a pressure, the difference between two related temperature sensors, or the difference between commanded and actual values for a set point or actuator position. Also, augmenting the vector with values collected during previous time slices can implicitly capture short term system operation patterns and trends. Flight controllers and engineers familiar with the monitored system can often suggest useful telemetry and derived parameters to use in the health monitoring vectors. Before processing, these vector values are typically normalized by applying z-score normalization or a similar method to each of the parameters.

In some cases, it may be advantageous to increase or decrease the significance (weight) given to certain vector parameters. For instance, if maintaining a specific operating pressure is critical to a system, the weight of that pressure value could be increased so a small deviation in the pressure would manifest as a larger change in the associated vector parameter, increasing monitoring sensitivity to variations in that parameter. Conversely, if the monitored system is not particularly sensitive to a certain parameter, such as ambient temperature, the weight of the ambient temperature value could be decreased to reduce the chance of unnecessary alarms when that parameter value changes by an insignificant amount.

Each monitored system will present unique characteristics. Techniques are currently under development to assist with parameter selection and weighting based on automated data analysis. However, normalization and parameter weighting schemes may sometimes benefit from manual tuning to suit the situation or meet particular monitoring goals.

B. IMS: Inductive Monitoring System

The Inductive Monitoring System (IMS) is a distance-based anomaly detection tool that uses a data driven technique called clustering to extract models of normal system operation from archived data.² IMS works with vectors of data values as described in the previous section. During the learning process, IMS analyzes data collected during periods of normal system operation to build a system model. It characterizes how the parameters relate to one another during normal operation by finding areas in the vector space where nominal data tends to fall. These areas are called nominal operating regions and correspond to clusters of nearby, similar points found by the IMS clustering algorithm. IMS represents these nominal operating regions as hyper-boxes in the vector space, providing a minimum and maximum value limit for each parameter of a vector contained in a particular hyper-box. These hyper-box cluster specifications are stored in a knowledge base that IMS uses for real-time telemetry monitoring or archived data analysis. Figure 2 shows an overview of the IMS method.

1. IMS Learning Process

In general, the number and extent of nominal operating regions created during the IMS learning process is determined by three learning parameters: the “maximum cluster radius” can be used to adjust the size and number of clusters derived from a fixed number of training data points, the “initial cluster size” is used to adjust the tolerance of newly created nominal operating regions, and the “cluster growth percent” is used to adjust the percent increase in size of a nominal operating region when incorporating new training data vectors. More specifically, the learning algorithm builds a knowledge base of clusters from successively processed vectors of training data. As such, the clustering approach is incremental in nature, which distinguishes it from well-known methods such as k-means clustering where the resulting clusters are independent of the ordering of the vectors. With the processing of each new training data vector, the

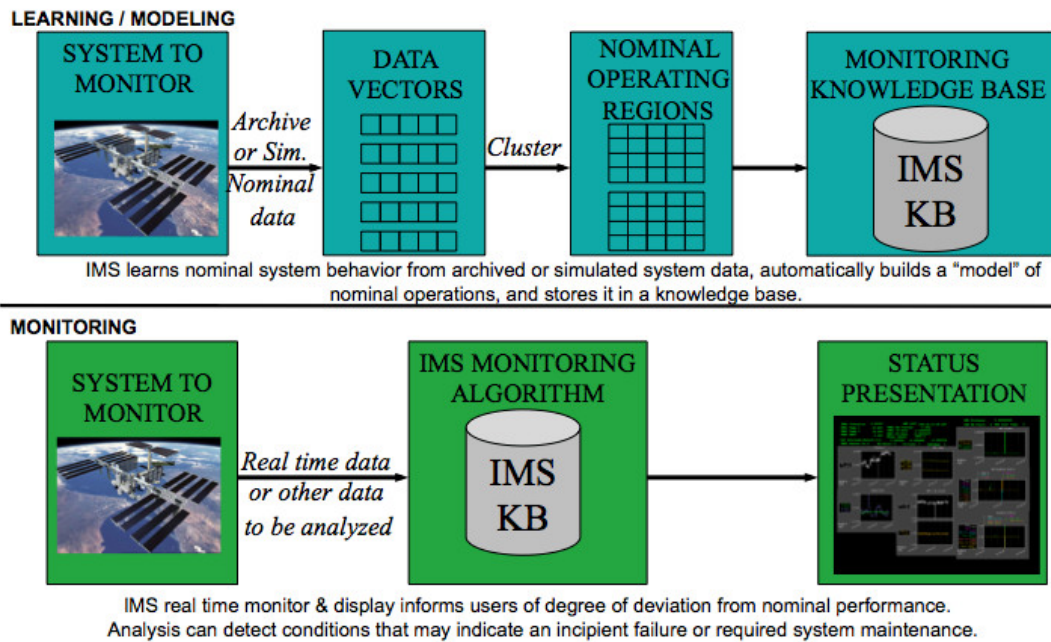


Figure 2. Inductive Monitoring System (IMS) overview

distance from this new vector to the centroid of the nearest cluster in the knowledge base is computed. If this distance is below a pre-specified value, the “maximum cluster radius”, the new vector is summarily incorporated into that cluster. The upper or lower limits for each affected dimension of the cluster are expanded respectively according to the “cluster growth percent” parameter to reflect the inclusion of the new vector. This incremental, inductive process gives IMS an advantage over other clustering methods such as k-means, since it tends to group temporally related points during the learning process. The grouping of temporally related points may also aid in discovering distinct system operations, which makes IMS more amenable to the specific goal of monitoring time series data for system operations.

The “cluster growth percent” parameter is used to adjust the learning rate. It establishes a fixed “growth” percentage difference for expansion of each dimension when updating previously formed clusters. This “cluster growth percent” learning parameter is therefore clearly proportional to the learning rate, due to the increased number of training data points that will be assigned to each new cluster per iteration for higher values of the “cluster growth percent” parameter. Naturally, the number of clusters in the knowledge base for a given training data set will increase as the “maximum cluster radius” and “cluster growth percent” values are decreased. Therefore, an inverse relationship between the maximum cluster radius and the number of clusters in the knowledge base exists. This dependence can be exploited to regulate the final size of the knowledge base in order to accommodate resource limitations in the computers running IMS.

If the distance between a newly processed vector and the centroid of the nearest cluster in the knowledge base is above the pre-specified “maximum cluster radius” value, a new cluster is created. The formation of a new cluster is accomplished by creating a hyper-box whose dimensions are based upon forming a window around each element of the new training data vector. The window is defined by introducing the “initial cluster size” parameter which is used to adjust the learning tolerance. This “initial cluster size” learning parameter represents a fixed percentage of the value for each dimension of the new training vector. As such, it relates directly to the size of newly established clusters, otherwise known as the “learning tolerance.” The “initial cluster size” and “cluster growth percent” learning parameters also act as buffers which enable a provisional allowance for manufacturing sensor tolerances and for sensors that may have suffered from deterioration due to wear. Furthermore, these learning parameters provide increased coverage to compensate for training data that may not fully characterize the nominal performance envelope.

2. IMS Monitoring Process

During the monitoring operation, IMS reads and normalizes real-time or archived data values, formats them into the predefined vector structure, and searches the knowledge base of nominal operating regions to see how well the new data vector fits the nominal system characterization. After each search, IMS returns

the distance from the new vector to the nearest nominal operating region, called the composite distance. Data that matches the normal training data well will have a composite distance of zero. If one or more of the data parameters is slightly outside of expected values, a small non-zero result is returned. As incoming data deviates further from the normal system data, indicating a possible malfunction, IMS will return a higher composite distance value to alert users to the anomaly. IMS also calculates the contribution of each individual parameter to the composite deviation, which can help identify and isolate the cause of the anomaly.

3. IMS Development Environment

An IMS development environment is under construction to facilitate production of IMS monitoring applications. It centers on a graphical user interface (GUI) that consolidates common IMS monitoring application development tasks (Fig. 3). These tasks include selecting useful data parameters and parameter weights, extracting and validating data from archive files, detecting and removing spurious and off-nominal data points from the archives, and building IMS knowledge bases. This development environment is intended to enable IMS end users, such as flight controllers or mission engineers, to conveniently develop their own IMS monitoring applications without consulting data mining specialists.

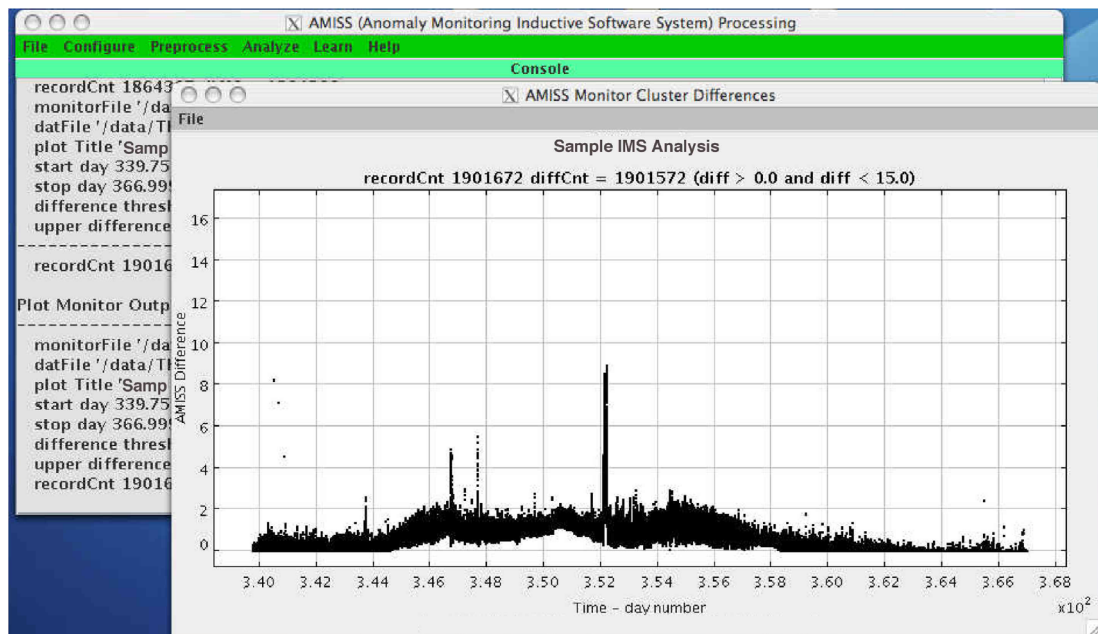


Figure 3. IMS development environment GUI showing composite results review and data editing tool

III. Space Operations Applications

The IMS software tool has been deployed in NASA mission control to support real-time telemetry monitoring and has generated interest in data-driven monitoring capability for other NASA programs. Several projects are currently underway to evaluate and mature the IMS technology and complementary tools for use in NASA's Constellation program. These include a vehicle system management experiment for the Air Force TacSat-3 satellite and pre-launch ground diagnostics for NASA's Ares I-X development flight test. Additionally, IMS is under evaluation for ongoing Ares I ground support equipment health monitoring and Constellation fleet supportability tasks.

A. International Space Station Mission Control

Thus far, IMS has been deployed in NASA's International Space Station (ISS) flight control room in support of two flight control disciplines: Attitude Control and Thermal Operations. The ISS Control Moment Gyroscope (CMG) attitude control system consists of four large gyroscopes, each mounted in a

gimbal system that can rotate the CMG about the two axes perpendicular to the gyroscope spin axis (Fig. 4). The CMGs operate as non-propulsive attitude control devices that exchange momentum with the ISS through induced gyroscopic torques.

As they have aged, some of the CMGs have degraded enough to malfunction and require replacement. Given their history, the ISS Attitude Determination and Control Officer (ADCO) flight controllers are interested in detecting early symptoms of degradation in the CMGs. A deployment of data-driven system health monitoring applications in the ISS flight control room is assisting with that task.

Working with ADCO flight controllers, 9 telemetered and 4 derived CMG parameters were selected for real-time monitoring. Seven to ten months of archived data was analyzed for each of the four CMGs. The data was sampled at a 1 Hz rate, formed into vectors of 13 values, and four IMS monitoring knowledge bases were constructed from the collected data. Each CMG was analyzed individually to capture its unique characteristics.

The IMS monitoring application was integrated with the NASA Mission Control data server software to access real-time telemetry in the ISS flight control room. Four IMS processes, one per CMG, run on the ADCO flight control console to provide continuous real-time monitoring. Once per second, each process compares incoming telemetry data with the appropriate CMG knowledge base and returns the amount of overall deviation, if any, from the nominal training data. It also returns the contribution of each individual parameter to any deviation to aid in isolating the source of the deviation. These IMS results are published back to the Mission Control data server for access and monitoring by other Mission Control software applications. IMS composite distances are plotted on ADCO console displays and automated alerts issued if significant deviations occur.

Successful deployment and certification of the IMS CMG monitoring system led to further development of real-time data-driven monitoring for ISS subsystems. The IMS CMG application was generalized to accept an arbitrary number of user-selected input parameters and to run on any controller console in the ISS flight control room. The resulting tool, called AMISS for Anomaly Monitoring Inductive Software System, has been applied to the thermal operations (THOR) domain, monitoring subsystems of the ISS External Thermal Control System (ETCS).

The ETCS is used to dissipate heat onboard ISS. Excess thermal energy from inside the ISS is transferred to liquid ammonia cooling loops in the ETCS. The heated ammonia is then circulated to radiators and cooled as thermal energy is released into space (Fig. 5). ETCS systems are separated into two independent loops with three major subsystems each: the Pump Module (PM), the Ammonia and Nitrogen Tank Assemblies (ATA/NTA), and the Radiators (RAD). The PM circulates coolant through the ETCS, the ATA/NTA stores reserve ammonia coolant and maintains pressure within the ETCS systems, and the RAD system controls the flow of coolant to each of three thermal radiators.³

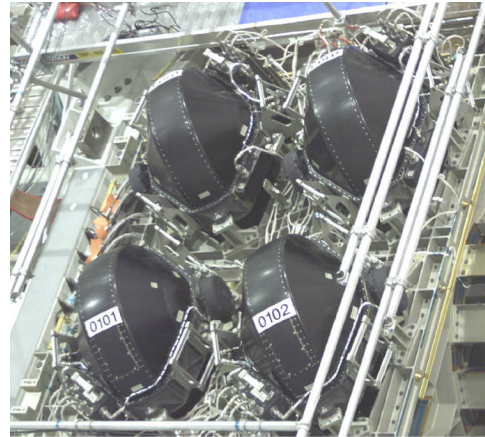


Figure 4. ISS Control Moment Gyroscopes



Figure 5. ISS external view showing ETCS radiators at lower left and lower right.

AMISS knowledge bases were constructed from a year of archived ETCS operational data, one knowledge base for each major ETCS subsystem, and two knowledge bases covering all pertinent parameters in each ETCS loop. The subsystem modules monitor for anomalies that occur within each subsystem while the full loop modules also watch for anomalies that are only apparent in subsystem interactions. The ETCS AMISS applications ran on the THOR ISS flight control consoles during recent ISS build activities to provide real-time system health information. Full operational certification of these applications is expected in 2009.

C. TacSat-3

The TacSat-3 Vehicle System Management (TVSM) project is a software experiment to demonstrate fault and anomaly detection algorithms and diagnosis tools integrated with executive and adaptive planning functions in a spaceflight context. The demonstration is based upon flight software and test data from the Air Force Research Laboratory's upcoming TacSat-3 satellite. The TVSM software package has the ability to monitor spacecraft subsystems such as power and guidance, navigation, and control (GN&C). It analyzes data in real-time to demonstrate detection of faults and unusual conditions, diagnose problems, and react to threats to spacecraft health and mission goals. The experiment will demonstrate the feasibility and effectiveness of ISHM technologies by combining a data-driven anomaly detection algorithm (IMS) with a model-based diagnosis tool (Qualtech Systems Inc. TEAMS-RT) and rule-based data analysis and recovery recommendations (implemented with Jet Propulsion Laboratory's (JPL) SHINE tool). The TVSM software has the ability to run both open and closed loop, the latter demonstrating feedback to reactive capabilities by the flight software. Following the completed demonstration on flight-like hardware, the TVSM software package will be proposed for a follow-on flight experiment on TacSat-3 itself.

D. Ares I-X

As previously mentioned, the Constellation program is developing vehicles to replace the Space Shuttle after its retirement, scheduled for 2010. The first vehicle will consist of the Ares I crew launch vehicle and the Orion crew exploration vehicle. The Ares V cargo launch vehicle and the Altair lunar lander will follow. Ares I-X, scheduled to launch in the summer of 2009, will be the first uncrewed test flight of the Ares I crew launch vehicle. The Ares I-X test vehicle will be powered by a single, four-segment reusable solid rocket booster (SRB), like those used on the Space Shuttle, modified to include a fifth inactive segment to simulate the Ares I five-segment booster. The Ares I-X vehicle will also include mock-ups of the upper stage, the Orion crew module, and the launch abort system to simulate the integrated spacecraft. It will be assembled in the Vehicle Assembly Building (VAB), launch from pad 39B, and will be controlled and monitored from Hangar AE, all at NASA Kennedy Space Center (KSC).

The Ares I-X Ground Diagnostic Prototype is currently under development⁶ to detect and diagnose faults in the Ares I-X first-stage thrust-vector control (TVC) and the associated ground hydraulics while the vehicle is in the VAB and while it is on the launch pad. It will combine three tools that use three different approaches. The first tool is TEAMS, which is a model-based tool from Qualtech Systems Inc. The second tool is SHINE (Spacecraft Health Inference Engine), which is a rule-based expert system from the Jet Propulsion Laboratory (JPL). The third tool is IMS. SHINE rules will be used to determine the system mode. IMS will have a separate knowledge base for each mode, and will dynamically load the appropriate knowledge base each time SHINE reports a mode change. TEAMS will be used to help operators diagnose and isolate the cause of detected system anomalies. The three tools will be deployed to Hangar AE, where they will be interfaced with live data from the Ares I-X vehicle and from the ground hydraulics. The outputs of the tools will be displayed on a screen in Hangar AE.

Because it is a new vehicle there is no historic Ares I-X sensor data available. However, the first stage of Ares I-X is derived from the Space Shuttle SRB and the Ares I-X first-stage TVC will be very similar to the Shuttle SRB TVC. Also, the ground hydraulics currently used with the Space Shuttle will also be used with Ares I-X. Therefore, historical Shuttle SRB data will be used to train and test IMS. In addition, simple scripts have been developed to simulate faults by replacing some of the historical data with simulated fault data. During development, these simulated faults will be used to test IMS's ability to detect the faults. It is expected that the Ares I-X TVC and ground hydraulics data will be similar to the historical SRB data used for training IMS.

The output of IMS will be the "distance from nominal" scores mentioned earlier – a composite score for the set of parameters as a whole and a separate score for each individual parameter. Additionally, to facilitate out of bounds conditions, IMS will output an alarm when scores exceed a specified threshold.

There are a number of methods to determine the threshold. Theoretically, if there were a comprehensive training data set available, IMS could learn the high and low thresholds of each individual parameter under every operating condition, so any deviation from the IMS model would warrant an alert. However, as a practical measure, the simplest and most often used method in previous deployments has been for the user to specify a threshold value based heuristically upon the statistics of available validation data. For example, twice the standard deviation of the composite score applied to validation data may serve as an alert level, or higher multiples of this value, depending on the skewness and kurtosis of the underlying distribution of the composite score. However, due to the fact that the distribution will invariably change as a function of the data provided, the alert thresholds may vary drastically from one dataset to another. As such, an alternative method can be used which has gained traction and is quickly becoming the standard for assessing performance of classification algorithms within the machine learning community and more recently the aerospace ISHM domain. This method involves the use of ROC (Receiver Operating Characteristic) curve analysis, and the area under the ROC curve (AUC). As such, one of the tasks of the Ares I-X Ground Based Diagnostics (GBD) project is to employ this method for threshold selection.

The ROC curve essentially plots the true positive rate against the false alarm rate for all possible threshold values. It therefore can be used as a design tool in order to select an alert threshold according to pre-established requirements for minimum missed detection and/or false alarm rates. However, in order to compute true positive and false alarm rates, it is necessary to obtain a “ground truth” representation for each monitored example. In this case, an “example” can represent an individual validation flight, or a single time slice. Furthermore, in order to compute true positive and false alarm rates with a reasonable level of accuracy, there is a need to obtain a statistically significant number of labeled examples, both nominal and anomalous. However, the availability of nominally classified examples often surpasses the availability of anomalously classified examples, with the latter often considered to be a rare commodity. In fact, as mentioned previously, it was necessary to write scripts in order to simulate faults in part to make up for the deficit of available anomalously classified examples. These simulations involve the injection of specific system faults or degraded modes of operation. As such, in order to artificially boost the number of anomalously classified examples, classified time points are used as examples in lieu of individual validation flights for construction of the ROC curve, even with the availability of simulated fault data.

The selection of an alert threshold by using the ROC curve is therefore in part based upon an implicit requirement for the availability of a statistically significant number of anomalous examples. As such, this requirement may be recognized as a distinct disadvantage, since it does not exist for the heuristic method described earlier. However, one advantage of using ROC curve analysis is in its inherent robustness against the use of skewed distributions. The distribution here, however, is not the distribution of the IMS score mentioned previously, but the distribution representing the population of nominally and anomalously categorized examples. Furthermore, unlike other alert threshold selection techniques, use of the ROC curve may also be used for optimized selection of IMS parameters (i.e. the number of clusters in a knowledge base). This can be performed by using the area under the ROC curve, which represents overall classification discriminability. Therefore, by maximizing the AUC with respect to the IMS parameters that control the number of clusters, we can ensure that the knowledge base used by IMS in order to perform threshold or alert selection has the best anomaly detection capability possible.

E. Ares I Ground Support Equipment

The IMS tool is under evaluation for continued use after the Ares I-X test flight to monitor vehicle and ground systems for the Constellation program. Initially, prototype IMS applications will be developed to monitor the liquid hydrogen (LH2) ground support equipment (GSE) required to fuel the Ares I launch vehicle. Although the Ares I LH2 GSE will be similar to the Shuttle LH2 GSE, the operating characteristics will differ (for example, tank capacity and fill rates will differ). Still, much of the development can be done in advance based on the Shuttle LH2 GSE, including selecting and weighting parameters as well as integrating with the data and application architecture. Some IMS training can also be done in advance using simulated data. The Constellation Ground Operations program is developing a GSE simulator for Ares I and Orion testing and integration, for Launch Control Center software development, and for operator training. The simulator may not be of high-enough fidelity to capture subtle operational characteristics of the LH2 GSE. Nevertheless, the simulator may be useful for generating nominal data for IMS training and generating fault data for IMS testing.

Higher fidelity training data will become available before the initial flight of Ares I. Prior to the first flight, the vehicle will be repeatedly fueled and de-fueled so that the operators can learn the actual

operational characteristics of the interacting hardware. IMS will be trained on these fueling tests. Thus, even before the first launch, high fidelity training data should be available to enable production of a prototype IMS monitoring application for the LH2 GSE as early as the first launch. It is expected that the operational data from that launch and perhaps the subsequent three to four following launches will provide enough data to capture expected nominal operations well, especially considering that the LH2 GSE has historically been static, that is, it has not had many changes after initial development.

In addition to proving feasibility, one goal of the LH2 GSE prototyping effort is to perform a trade study to determine what types of systems lend themselves well to a data-driven approach. Another goal is to document the effort (time and cost) required for building an IMS application. Because the Constellation vehicles and much of the supporting ground equipment are still under development, archived data is not currently available. Working with existing equipment to determine the characteristics of suitable systems and the effort required to build monitoring applications will allow effective planning for future deployments of IMS and similar data-driven systems. Lessons learned from IMS application to Ares I-X systems will also provide valuable insight for determining effective uses of data-driven techniques in the Constellation program.

F. Ares I Fleet Supportability

The data-driven approach can be applied to fleet supportability tasks. Fleet supportability encompasses the comprehensive performance monitoring and analysis conducted to assure that the Ares I fleet retains its intended design performance, reliability, and safety throughout the program lifecycle. It includes all non-real-time analyses of recorded data, including data generated during manufacturing, transport, mission preparation and prelaunch operations, in-flight, and post mission. It establishes a system that can determine the health condition of reusable components, the expected need for imminent replacement, the need for redesign of reusable or expendable components, or the need for procedure/process redesign.

Fleet supportability is typically associated with large fleets of similar equipment, for example, a fleet of F/A-18 aircraft. The operating characteristics of the fleet are used to develop a failure distribution. This profile can then be used to predict the failure of an individual component on similar instances of the fleet type. Thus, we can infer that a fuel pump may fail in as few as 500 hours or as many as 2000 hours but the majority of pumps fail at 1000 hours. This failure rate curve can be used to extend (or shorten) replacement periods to maintain a desired in-service failure rate. The replacement threshold can be set very low to account for early failures. This can result in increased costs due to much unnecessary replacement or maintenance. Alternatively, the replacement threshold can be set high to mitigate such costs. This can result in some components failing in service – not an acceptable solution for critical components on Ares I.

Another factor against relying on the life usage model described above for Ares I is that it will be difficult to collect enough performance data for statistical significance and accurate inference. The reusable components (the first stage of Ares I; limited iterations) will be in service for approximately two minutes per launch and the expendable components (upper stage and upper stage engine; new components for each launch) will be in service for less than ten minutes. Additionally, the fleet size of the Constellation program will be much smaller than the typical aircraft fleet, with a handful of launches per year.

With the limits imposed by number and duration of operational cycles, alternative approaches for fleet supportability are under evaluation. Ares I will not have the benefit of evaluating the performance of components in thousands of instances. Instead, a data-driven approach like IMS will be applied to look for subtle differences in performance that may indicate impending failure.

Performance-degrading conditions can occur throughout a component's lifetime, from design through launch and reentry to refurbishment. Numerous pre-launch tests verify compliance with expected performance. If a subsystem fails a test, a diagnostic system can isolate the fault to an LRU (line replaceable unit) or perhaps even to a component. An analysis is then performed to determine why that component failed – whether it was due to changes in manufacturing, materials, or quality control, environment during transport, or effects from previous launches, reentry, recovery, and refurbishment. The results of that analysis can then inform required maintenance planning, guide part restocking decisions, or identify needed manufacturing improvements.

Subtle differences in performance may not be noticed if the pre-launch tests pass. Current monitoring techniques – primarily visual comparison of graphs – focus on identifying differences in a single parameter. Because IMS analyzes the interaction between all the parameters simultaneously, it can complement these tests by detecting that the performance on a test is still within limits but is different than on previous tests either on this system or on previous systems of the same type. This may lead to early identification of

under-performing components prior to in-flight failures on subsequent flights. It may also identify the need for revised limits for prelaunch tests. Further, it could assist with both the post-failure analysis as well as degraded-performance analyses by identifying conditions that degrade performance or reduce a component's lifetime. Alternatively, if components are performing well even in the presence of unusual conditions, it may indicate that environmental constraints could be relaxed. Finally, IMS output may be useful as an input to prognostics algorithms if a correlation with future performance can be determined.

IV. Summary and Future Work

Through practical application, it has been demonstrated that data-driven system health monitoring can be useful in a variety of space mission operations settings. Furthermore, data-driven methods can complement other data analysis and diagnosis tools, together providing integrated system health management and fault recovery recommendations. A number of NASA projects incorporating data-driven anomaly detection have been deployed or are under development. The emergent projects will provide proof of concept demonstrations by combining data-driven techniques with model-based and rule-based software tools for fault management. They will also produce recommendations on which types of systems would most benefit from data-driven and integrated data-driven, rule-based, and model-based approaches for fault detection and health management. This information will help guide NASA as it builds operations and support systems for Constellation. Finally, these data-driven concepts also show promise for fleet supportability applications and will be evaluated for that task as the Constellation support plans grow and mature.

For future work, near term plans call for continued development of tools that allow operations personnel to build and maintain their own data-driven monitoring applications without direct support from data mining specialists. This includes utilities to help select and appropriately weight useful system monitoring parameters. These will be combined with tools to detect and remove spurious data from archived data sets used to build monitoring applications. All of these tools will be incorporated into the graphical integrated development environment, allowing users to conveniently proceed from raw archived data to a deployable monitoring knowledge base.

Longer-term activities will address capabilities required for wider application of data-driven and complementary technologies to Constellation and similar programs. One important task to enable the use of data-driven approaches with newly developed hardware systems will be determining how to obtain adequate system characterizations with limited operations data. This will likely involve augmentation with simulation data or synthetic data extrapolated and generalized from available operations data. A related endeavor will address generalizing and normalizing training data collected during operations performed under varying conditions, such as launches in differing weather conditions (e.g., hot, humid summer launches vs. cooler winter or night time launches), and determining how much influence the disparate conditions have on the monitored system characteristics.

The study of combining data-driven technology with other ISHM techniques is still in its infancy and holds promise of further utility, including more sophisticated diagnostic tests and prognostic capabilities. There are also many applications of data-driven approaches yet to be explored for fleet supportability tasks. For instance, multivariate data-driven approaches are naturally complementary to traditional fleet supportability statistical analysis, and may be more effective when limited instances of supported equipment are available. It can also be a valuable technique for sorting through massive amounts of data to focus attention on key pieces of information, both from extended operations and from test and assembly of a new vehicle or system. Additionally, using these techniques to pick out the most divergent data points in operations data can either be used to drive risk-based testing and verification, or in a more traditional sense to simply hand the most interesting periods to design analysts. It may also be possible to classify current excursions from data-derived models to analogous historical operating periods by the degree and pattern of divergence. This could be used to drive procedural and recovery decisions.

References

¹Iverson, D. L., "Data Mining Applications for Space Mission Operations System Health Monitoring", *Proceedings of the SpaceOps 2008 Conference*, ESA, EUMETSAT, AIAA, Heidelberg, Germany, May 2008.

²Iverson, D. L., "Inductive System Health Monitoring," *Proceedings of The 2004 International Conference on Artificial Intelligence (IC-AI04)*, CSREA, Las Vegas, Nevada, June 2004.

³International Space Station Thermal Control System Training Manual, NASA Document ISS TCS TM 21109, January 2004.

⁴Schwabacher, M., Oza, N., and Matthews, B., “Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring,” *Proceedings of the AIAA Infotech@Aerospace Conference*, AIAA, Reston, VA, 2007.

⁵Martin, R. A., Schwabacher, M., Oza, N., and Srivastava, A., “Comparison of Unsupervised Anomaly Detection Methods for Systems Health Management Using Space Shuttle Main Engine Data,” *Proceedings of the JANNAF Propulsion Meeting*, JANNAF, Denver, CO, 2007.

⁶M. Schwabacher and R. Waterman. Pre-Launch Diagnostics for Launch Vehicles. *Proceedings of the IEEE Aerospace Conference*, 2008.

⁷M. Schwabacher, R. Aguilar, and F. Figueroa. Using Decision Trees to Detect and Isolate Simulated Leaks in the J-2X Rocket Engine. *Proceedings of the IEEE Aerospace Conference*, 2009. To appear. http://ti.arc.nasa.gov/m/profile/schwabac/Schwabacher_IEEE_Aero_2009%20revised.pdf