Autonomic Integrated Prognostics Health Management Systems: Concepts and Designs

Tim Kim[1], Chris Lynn[2], Neil Kunst[3], and Sonia Vohnout[4]

[1,2,3,4] *Ridgetop Group Inc., 3580 West Ina Road, Tucson, Arizona, 85741, USA*

tim.kim@ridgetopgroup.com
chris.lynn@ridgetopgroup.com
neil.kunst@ridgetopgroup.com
sonia.vohnout@ridgetopgroup.com

ABSTRACT

The current state of the art in health management systems does not fully support the detection, collection, and remediation of real-time faults in mission critical platforms such as satellites. This has led researchers to consider alternative health management paradigms and techniques that are based on strategies used by biological systems to deal with complexity, dynamism, heterogeneity, and uncertainty. In the proposed autonomic health management paradigms, data on multiple characteristics will be sequentially collected from various components and/or subsystems from different levels of the system. The resulting data set will create a multi-dimensional data-rich environment, consisting of a *temporal* time-series degradation signal, *spatial* multivariate characteristics measurements, and hierarchical multi-level system layout information. Ridgetop's autonomic health management paradigm explained in this paper utilizes this collected sensor data, which handles uncertainties and anomalies, and realizes systems and subsystems capable of managing themselves with minimal human involvement.*

1. INTRODUCTION

Fault detection, analysis, and recovery coupled with effective monitoring in mission critical system environments, is a challenging research problem due to growth in scale and complexity of applications, the continuous changes in resource configuration, and the variety of services being offered and deployed. These capabilities maximize system effectiveness in the presence of anomalies and are defined as health management; health management technologies have been considered critical for detection and prediction of impending system faults, initiating fault mitigation, and providing valuable information to facilitate proactive logistics planning and fleet-operation decision processes (Kim, 2008).

Despite past advances in health management within government and commercial sectors, numerous difficulties persist in real-world scenarios. To illustrate, in satellite systems it is difficult to carry-out inspections, even in ideal circumstances, largely due to their complexity and physical density. After deployment, satellites often exhibit problems from connection fatigue, pixel degradation, contamination, stuck focus motors or actuator mechanisms, or frozen components.

Radiation damage from solar events creates additional challenges in maintaining system operational readiness. Various system environments, particularly those employed in safety-critical environments should be enabled to perform correctly despite fault occurrences.

The significance of these issues has grown to such an extent that the need for autonomic health management systems with self-healing appears essential for successful and cost-effective system performance.

The potential impact of self-healing systems has led researchers to consider alternative health management paradigms and techniques that are based on strategies used by biological systems to deal with complexity, dynamism, heterogeneity, and uncertainty.

After extensive studies regarding health management technologies and issues conducted by numerous researchers, such as Malin and Oliver, Bird, and Millar, an autonomic health management concept and design was developed. The researchers used the human autonomic nervous system as inspiration, since it handles uncertainties and anomalies and realizes systems and subsystems capable of self-management with little to no human involvement.

In this paper, Ridgetop first outlines a concise explanation of the autonomic nervous system and uses it as an analogous approach to developing the autonomic health management concept and design. Ridgetop then illustrates smart sensor works in progress to explain fault monitoring and analysis which is one of component in our concepts.

2. MOTIVATION

Contemporary science points to the human nervous system as the most intricate model of autonomic behavior today. The human nervous system acts as the dominant regulator within the body, supervising internal and external changes, integrating sensory inputs, and channeling appropriate responses. Together with the endocrine system, the nervous system sustains homeostasis, an interdependent scheme of meticulous checks and balances. Homeostasis evaluates even the slightest environmental flux, including random disturbances. The nervous system reacts to changes with a series of modifications equal in size and opposite in direction to the disturbance source.

The goal of these modifications is to maintain internal balances essential to the system's well-being. Homeostasis is pervasive in the human body. For example, note the delicate requirements of glucose limitations in the blood: concentrations must be kept below 0.06% and above 0.18%, either side of that range spells disaster. To counteract disaster, if the blood-glucose concentration falls below 0.07%, adrenal glands secrete adrenaline, causing the liver to convert glycogen into glucose, which then passes into the blood and counteracts the blood-glucose concentration drop. If the blood-glucose concentration rises excessively, the secretion of insulin by the pancreas is increased, causing the liver to intervene and remove excess glucose from the blood. Excess glucose removal also occurs through muscles and skin. If the blood-glucose concentration exceeds 0.18 %, the kidneys excrete excess glucose into the urine. This example of homeostasis provides a simplified illustration of the diversity of vital checks and balances within the human body.

Analogous regulation systems exist for other human body mechanisms, such as systolic blood pressure, structural integrity of the *medulla oblongata*, and severe pressure of heat on the skin. These systems are vital for the organism survival (in this case the human body). However, the vital nature of these mechanisms is not necessarily uniform across the board. Certain mechanisms are more closely linked to survival than others. In addition, these and other mechanisms that are closely linked to each other, reflect changes linearly. In other words, marked changes that occur in one mechanism eventually appear in another. These linked systems are considered essential variables in studying the design and structure of the human brain, (Ashby, 1960).

Autonomic computing concepts, initiated by IBM (Kephart et al., 2003), encompass self-governing computer systems that manage rapid growth complexities with a minimum of human interaction. The term autonomic derives from human biology: the autonomic nervous system self-monitors without any conscious effort, (Kim and Hariri, 2007). Similarly, autonomic health management capabilities anticipate system requirements and resolve issues with nominal human interface. This allows system professionals to focus on critical tasks.

3. AUTONOMIC HEALTH MANAGEMENT SYSTEM CONCEPTUAL ARCHITECTURE

Ridgetop's approach is based on an autonomic computing paradigm that requires initial continuous monitoring and analysis of the system state, followed by planning and execution of appropriate actions, if it is determined that the system deviates appreciably from expected normal behaviors, as shown in Figure 1.

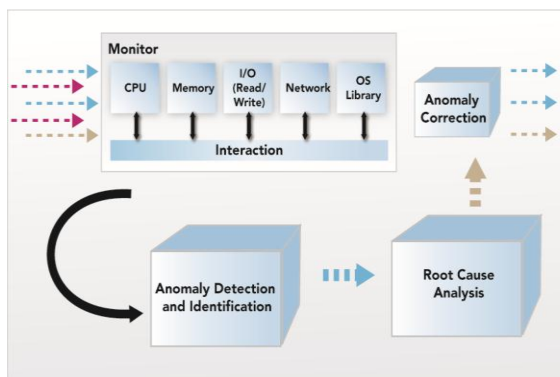

Figure 1. Conceptual Architecture Flow.

By monitoring each subsystem, Ridgetop collects measurement attributes from each subsystem's operations such as SMPS output filter capacitor, feedback amplifier, PWM controller, etc. Data analysis can reveal anomalous behavior potentially triggered by failures. Once a fault is detected, the next step is, if needed, to identify the source of faults and the appropriate fault recovery strategy to bring the system back into a fault-free state.

Another important function in our concept is the control, as shown in Figure 2, which manages unexpected events as faults occur at runtime. A fault bears the potential of to slow down the entire system and affect overall performance of the application.

These functionalities have been executed by the following engines:

- Monitoring Engine (ME)
  Monitors the state of the system environment through various sensors.

- Fault Analysis Engine (FAE)
  Analyzes data for fault detection.

- Fault Identification Engine (FIE)
  Identifies fault sources and the appropriate fault recovery strategy.

- Execution Engine (EE)
  Plans overall execution strategies to optimize behaviors and operations (self-healing) of system environments.

- Knowledge Engine (KE)
  Provides support for decisions on the appropriate rule based on a set of rules that improves functionalities and performance.

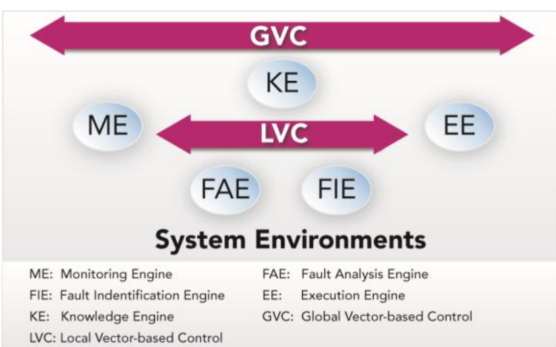

Figure 2. Autonomic Conceptual View, Control.

It realizes autonomic control and management objectives with aid of two closed loop control subsystems.

3.1 Local Vector-based Control

The local, or fine vector-based, control loop will locally handle behavior of individual and local system elements on which the components execute. This can be viewed as adding self-managing capabilities to conventional components/elements. This loop will control local algorithms, resource allocation strategies, distribution and load balancing strategies, etc. Note that this loop will only handle *known* environment states; the mapping of environment states to behaviors is encapsulated in its *knowledge engine* (*KE*).

For example, when the load on the system resources exceeds the acceptable threshold value, the fine loop control will balance the load by either controlling the local resources or by reducing the size of the computational loads. This will work only if local resources can handle the computational requirements.

However, the fine loop control is blind to the overall behavior and cannot achieve the desired overall objectives unaided. Therefore, the fine loop control unassisted can lead to sub-optimal behavior.

3.2 Global Vector-based Control

A time will occur when one of the system's essential variables will exceed its limits, triggering a global vector-based control loop subsystem. The global control loop will then manage behavior of the overall application and define knowledge that will drive local adaptations.

This control loop can manage unknown environmental states using three cardinals (fault-tolerance, configuration, and performance) for monitoring and analysis of the high-performance, mission-critical establishment. These cardinals are analogous to essential variables described in Ashby's ultra-stable system model (1960) of the autonomic nervous system.

This control loop acts toward changing existing behavior of a superior-performance, mission-critical set-up so that it can adapt to environmental changes.

For example, in the previous load-balancing set-up, existing behavior, as directed by the local loop, preserved the local load within prescribed limits. This action carried out blindly can corrupt overall system performance.

Ultimately, this change in the cardinal associated with overall performance, triggers the global control loop. The global control loop then selects an alternate behavioral pattern from a pool of patterns. The execution engine (EE) determines the appropriate plan of actions using its KE. The EE then executes the new plan within the critical environment in order to adapt its behavior to new conditions.

The most noteworthy feature of the autonomic system is the integrated approach of its controller. The controller unit manages, in an integrated manner, fault performance and configuration of systems and their applications.

In the classical paradigm, each of these properties has been isolated and treated separately. These practices have contributed extensively to the control and management challenges of large-scale interacting and dynamic computing systems and services.

The next section details Ridgetop's approach to implement an autonomic health management system based on the conceptual architecture..

4 AUTONOMIC HEALTH MANAGEMENT SYSTEM FRAME WORK

The importance of an autonomic health management system framework is exemplified in its ability to automatically detect and recover from a wide range of faults.

The framework is designed to achieve fault-tolerant services for any components such as subsystems, system, applications, hardware resources and services. It achieves anomaly detection by tracing the interaction among components during runtime, identifying the source of faults and then planning the recovery actions without any user interference, as shown in Figure 3.

Our framework consists of several core modules, such as the Self-Healing Engine (SHE), Application Fault Management Editor (AFME), System Health Manager (SHM), and Component Fault Manager (CFM).

The *Application Fault Management Editor* (AFME) allows users to specify the health requirements associated with each component or system involved in the health management system. SHE receives the requirements from the AFME and builds an appropriate *Fault Management Strategy* (FMS).

The composition of FMS by SHE is policy driven. Policies are a set of predicates that define a course of action.

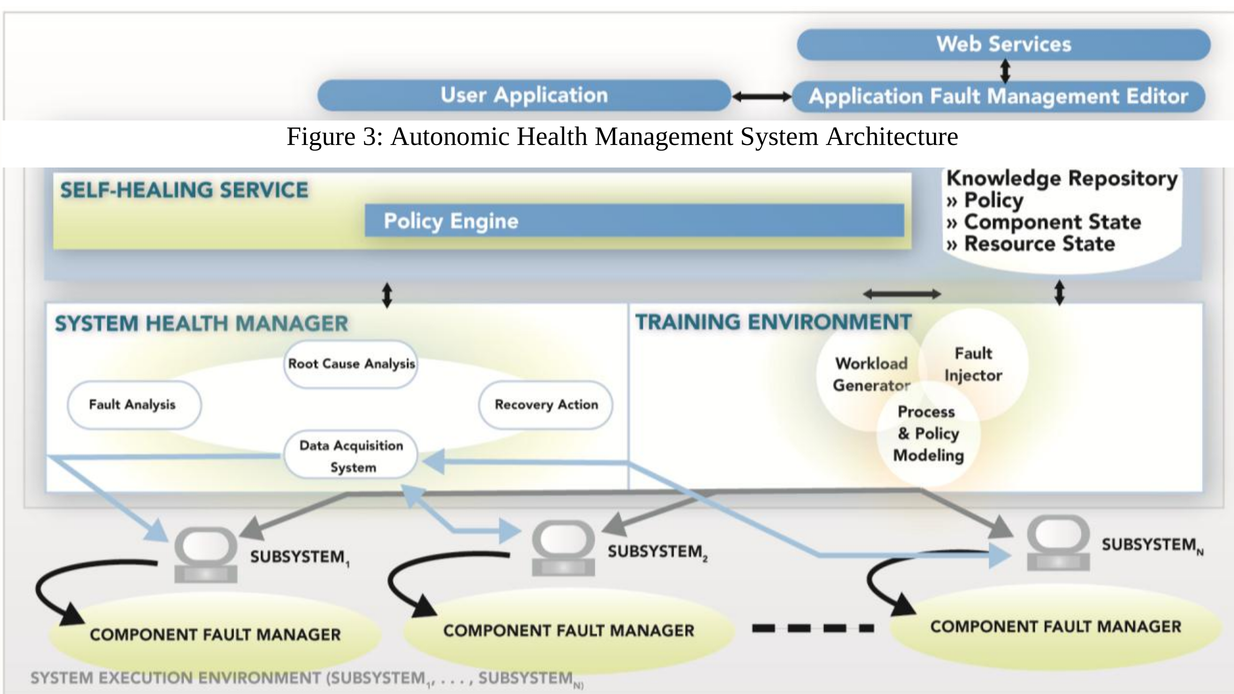

Figure 3: Autonomic Health Management System Architecture

For each health management requirement, the knowledge repository defines the appropriate health management strategy to maintain such requirements.

Through this process, the SHE can identify the appropriate resource configurations to run the selected FMS in runtime. Once the environment is properly configured, the SHM which is responsible for executing and maintaining the FMS, gets an instance of the FMS

4.1 System Health Manager

The System Health Manager (SHM) is responsible for several activities, such as monitoring, anomaly analysis, root-cause analysis, and recovery.

Runtime behavior interactions of components and services are stored in a knowledge repository with features capturing spatial and temporal operations for each component.

Once we obtain behavior information, we can analyze the component behavior to detect any anomalous events that might have been triggered by faults.

To increase the accurate modeling of anomaly behavior, AFM also has a fault injector and a workload generator that are used during the training phase.

If the behavior of application shows abnormal states, SHM identifies the source of faults by failure mode effect analysis (FMEA) with fault tree.

Once a fault is detected, the next step is to identify the appropriate fault recovery strategy defined in the FMS to bring the system back into a fault-free state.

The activities such as monitoring, anomaly analysis, root-cause analysis, and recovery exist in a hierarchical fashion so that failure of any one level does not lead to failure of the entire system

4.2 Component Fault Manager

The first step for fault detection is to identify a set of measurement attributes that can be used to define the normal behaviors of these components as well as their interactions with other components within the distributed system.

For example, when a user runs a QuickTime application in computing environments, one can observe certain well-defined CPU, memory, and I/O behaviors.

These operations will vary significantly when the application experiences unexpected failures that lead to application termination.

One can observe that the application, although consuming CPU and memory resources, does not interact normally with its I/O components.

CFM resides in each component and traces all sets of measurement attributes for applications and nodes. Once it is collected, it sends monitored data to a knowledge repository with features.

In addition, the CFM, which is a subordinate to the SHM, focuses on the execution and maintenance of the FMS associated with the component running the CFM module.

During runtime phase, SHM and CFM maintain operations of each resource and component according to the policies specified in the FMS

5 AN ILLUSTRATIVE EXAMPLE: SMART POWER SENSOR

In this section, Ridgetop introduces smart sensor technology, which can be implemented as either a single-board or system-on-chip (SoC) solution.

Ridgetop's smart sensor technology is based on the extraction of eigenvalues from data collected by CFM with wide applicability to EPS, EMA, and power drive stages to show initial testing results for fault detection.

5.1 Sensor Hardware

The top panel of the prototype IEEE 1451- enabled smart power sensor is presented in Figure 4.

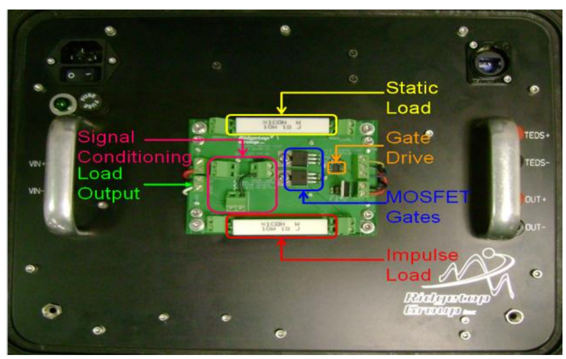

Figure 4. Top Panel View of RS1000-1 Smart Power Sensor.

The close-up view highlights a key advancement in Ridgetop's power supply load control technique. The improved load board enables dynamic control of the two load resistors fundamental to our RingDown prognostics methodology: static and impulse.

In legacy testbeds, like the RD1000-1 deployed in the ARC Advanced Diagnostics and Prognostics Testbed (ADAPT), the static load is always enabled and sized to draw approximately 25% of the target power supply's power.

To minimize heat dissipation in the static load resistor, allow for reductions in power rating and ultimately size, the smart power sensor provides switching control of the static load, as well as the impulse load.

The static load resistor is enabled first for 4 milliseconds (ms). Midway through the static load period, or 2 ms after static load is enabled, the impulse load is enabled for 10 microseconds (µs).

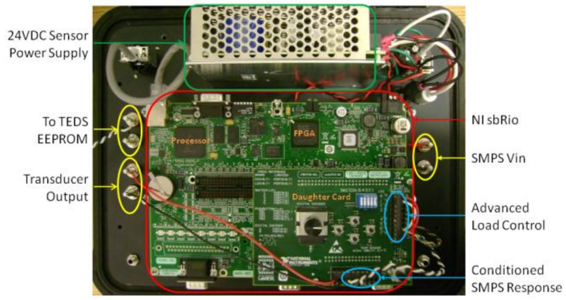

Figure 5. Bottom Panel View of RS1000-1 Smart Power Sensor.

As shown on the right side of Figure 5, the load board's digital control and conditioned power supply signals are routed to the embedded sbRio mounted to the bottom side of the prototype sensor.

The sbRio's embedded 400 MHz MPC5200 Freescale processor provides the computing horsepower needed for the signal processing and data analysis tasks of the smart sensor; while the Spartan 3 Xilinx FPGA is programmed to manage the data acquisition functionality.

The analog and digital I/O signals are routed through the NI DAQ daughter board to provide load control of the target power supply, RingDown waveform capture, and transducer SoH/RUL output. For this first prototype, the TEDS functionality is provided by a Maxim 1-Wire DS2433 EEPROM mounted to the bottom of the sensor case.

5.2 Experimental Process and Results

The singular assumption made in the algorithm development process for RingDown technology is that a power supply's response to a changing load will adjust with its health.

First, it is essential for an understanding of what a normal or healthy response is. Data collected from the test setup is shown in Figure 7 and illustrates a healthy response of the power supply regulating to 5 V.

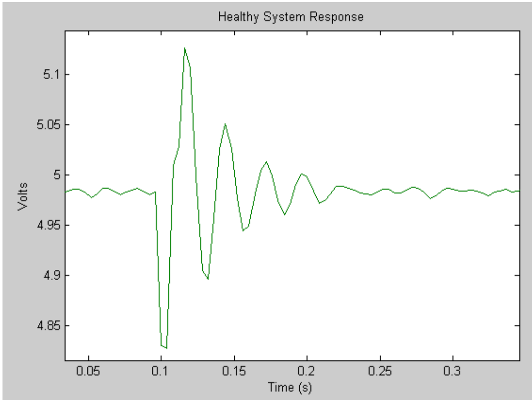

Figure 7. Healthy System Response in Power Supply Regulating to 5 V.

This response is ideal since the time in which the device is out of regulation lasts approximately 1/10 of a second and the fluctuation is only 3/10 of a volt.

Less healthy systems stay out of regulation longer and their voltages fluctuate comparable to variation, individual faults, and advanced stages of compound faults, which do not require complex algorithms to diagnose.

However, a self-imposed requirement of accurately predicting RUL at all times requires the

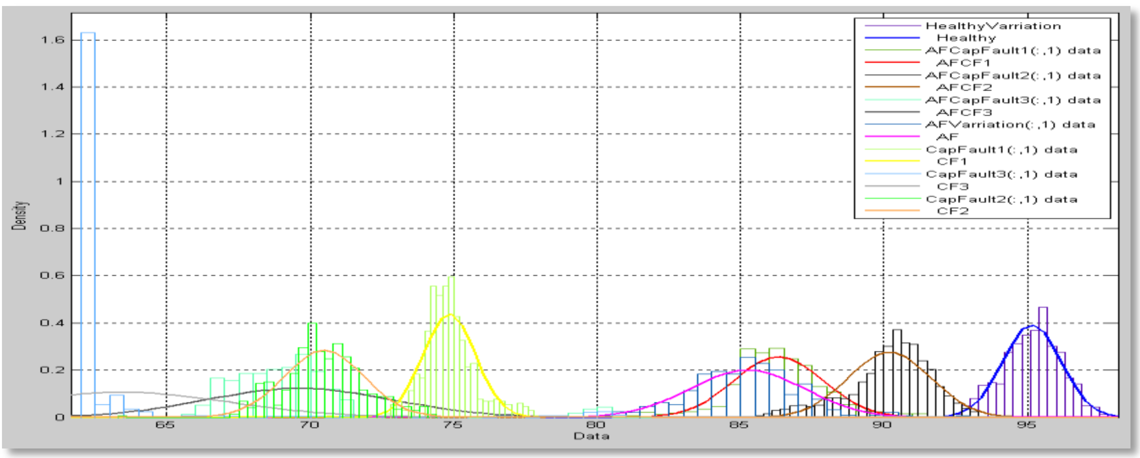

Figure 6. PDF of SMPS Health with Multiple Fault Cases.

implementation of these complex algorithms.

Difficulty in detecting degrading health of a power supply arises when compound failure mechanisms manifest simultaneously, since the characteristics of one failure mode may compensate for the characteristics of another.

In our experiments, we consider various fault scenarios including compound faults by considering amplifier degradation fault and capacitor degradation fault simultaneously.

Figure 6 is a probability density function (PDF) plot of Ridgetop's fault detection algorithm output from SHM. The x-axis represents the computed output SoH value across multiple induced fault conditions. The expected result was that the SoH value would decrease as more severe faults were injected. The theory, also proven by this plot, is that while health decreases, variation in the computed SoH increases. This asserts that as SMPS health decreases, system dependability breaks down.

The blue curve shown in Figure 6 represents the healthy system response which has a standard deviation of 1.0187%; the brown curve represents an early fault case, but already the standard deviation is 1.4454 %.

The increase occurs in all except the first level cap fault, which still rates 75% healthy. These results show the critical distinction between normal and abnormal status.

6 CONCLUSION

We have presented autonomic health management paradigms to detect the fault with effective sensor monitoring in mission critical environments.

Autonomic health management concepts represent most of its principles from autonomic human nervous system that has proved its effectiveness to handle uncertainty and anomalies in complex and heterogeneous environments.

We strongly believe the proposed autonomic health management will enable us to make advance in the way. However, since autonomic concept is an emerging technology in the health management field, numerous challenges persists that require resolutions and improvements by the research community.

In closing, Ridgetop presents the following queries to address integral issues facing this evolving technology:

- How can we dynamically configure heterogeneous resources at runtime?
- How can we confirm pointing systems and subsystems?
- Do we need a concept for autonomic components to deal with autonomic properties, such as self-configuration and self-healing?
- If required, how can we integrate existing components?
- How can we convert existing components to have autonomic properties?
- How can we improve monitoring functions to a decision-making level?
- How can we create dynamic composition to add, delete, and change the algorithm at runtime?

REFERENCES

J. O. Kephart and D.M. Chess. (2003) The Vision of Autonomic Computing, IEEE Computer, 36(1), pages 41-50

W. Ross Ashby, Design for a brain (Second Eidtion Revised 1960), published by Chapman & Hall Ltd, London.

Bird, Jeff et al. (2007), "A Framework of Prognosis and Health Management- A Multidisciplinary Approach", Proceedings of ASME Turbo Expo 2007

Paris, Deidre E.; Michael D.Watson, Luis D. Trevino (2005), "A Framework for Integration of IVHM Technologies for Intelligent Integration for Vehicle Management," IEEE

Malin, Jane T. and Oliver, Patrick 1. (2007), "Making Technology Ready: Integrated Systems Health Management", AIAA Infotech at Aerospace Conference

Keller, Kirby J. et al. (2007), "Health Management Engineering Environment and Open Integration Platform", IEEEAC

Clark, Greg 1. et al. (2007), "Multi-platform Airplane Health Management",IEEAC

Wilmering, Timothy 1. and Ofsthun, Stanley C. (2006), "Towards a Generalized Framework for Health Management Application Integration", IEEE

Millar, Richard C. (2007), "A Systems Engineering Approach to PHM for Military Aircraft Propulsion Systems", IEEEAC

Kim, Byoung U. (2008), "Anomaly-based Self-Healing Framework in Distributed Systems." Diss. University of Arizona

Kim, Byoung U, et al. (2008), "Anomaly-based Fault Detection in Pervasive Computing Systems." Paper presented at the Proceedings of the ACM International Conference on Pervasive Services (ACM ICPS)

Kim, Byoung U, and S. Hariri. (2007), "Anomaly-based Fault Detection System in Distributed Systems." the 5th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (IEEE SERA)

Tim Kim, Ph.D. *is a senior R&D engineer at Ridgetop Group Inc., where he has contributed to ground-breaking technological improvements in electronic prognostics. His current research involves integrating Ridgetop's sensor array technology with reasoning engines and developing incorporated self-healing algorithms.*

Chris Lynn *is an electrical engineer at Ridgetop Group Inc. His expertise is in computer modeling and determining the reliability of critical systems and predicting their failures. His BSEE program at the University of Arizona entailed advanced studies in device physics and state-space modeling.*

Neil Kunst *is an engineering project manager at Ridgetop Group Inc. with more than 20 years of experience in product engineering, systems engineering, test engineering, logic design, software development, project management, and consulting. He earned his BSEE from the University of Arizona, where he was a member of the Tau Beta Pi National Honor Society. Mr. Kunst received the Silver Bowl award and awards for outstanding achievements in physics.*

Sonia Vohnout *earned her MSEE from the University of Arizona. Ms. Vohnout manages Ridgetop's commercialization efforts for its many government-funded projects.*