Benchmarking the Vehicle Integrated Prognostic Reasoner

Xenofon Koutsoukos[1], Gautam Biswas[1], Dinkar A. Mylaraswamy[2], George D. Hadden[2], Daniel Mack[1], Doug Hamilton[3]

[1] *Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, 37235, USA*
Xenofon.Koutsoukos@vanderbilt.edu
Gautam.Biswas@vanderbilt.edu
Daniel.L.Mack@vanderbilt.edu

[2] *Honeywell, 1985 Douglas Drive North, Golden Valley, MN 55422, USA*
dinkar.a.mylaraswamy@honeywell.com
George.d.hadden@honeywell.com

[3] *Lockheed Martin Space Company, Denver, CO 80201, USA*
(Work done while at Honeywell)
douglas.w.hamilton@lmco.com

ABSTRACT

This paper outlines a benchmarking approach for evaluating the diagnostic and prognostic capabilities of the Vehicle Integrated Prognostic Reasoner (VIPR), a vehicle-level reasoner and an architecture which aims to detect, diagnose, and predict adverse events during the flight of an aircraft. A number of diagnostic and prognostic metrics exist, but these standards are defined for well-circumscribed algorithms that apply to small subsystems. For layered reasoners, such as VIPR, the overall performance cannot be evaluated by metrics solely directed toward timely detection and accuracy of estimation of the faults in individual components. Among other factors, the overall vehicle reasoner performance is governed by the effectiveness of the communication schemes between the different monitors and reasoners in the architecture, and the ability to propagate and fuse relevant information to make accurate, consistent, and timely predictions at different levels of the reasoner hierarchy. To address these issues, we outline an extended set of diagnostic and prognostics metrics that can be used to evaluate the performance of layered architecture, and we discuss a software architecture as well as an evaluation plan for benchmarking VIPR.*

1. INTRODUCTION

This paper outlines a benchmarking approach for evaluating the diagnostic and prognostic capabilities of the Vehicle Integrated Prognostic Reasoner (VIPR), a vehicle-level reasoner and an architecture which aims to detect, diagnose, and predict adverse events during the flight of an aircraft. All these functions are aimed at meeting the goal of automated mitigation and increasing aviation safety. Faults can arise in one or more aircraft subsystems; their effects in one subsystem may propagate to other subsystems, and faults may interact. VIPR must be able to handle these interactions and provide an accurate diagnostic and prognostic state for the aircraft. VIPR is a vehicle level reasoner at a prototype stage that relies on diagnostic and prognostic monitors available at the aircraft subsystem and LRU level. VIPR performs system-level reasoning by systematically decomposing the problem via a layered architecture.

A number of diagnostic and prognostic metrics have been reported in the literature (Byington et al., 2003; Kurtoglu et al., 2008; Saxena et al., 2008; Leao et al., 2008; Kurtoglu et al., 2009), but these standards are defined for well-circumscribed algorithms that apply to small subsystems. On the other hand, VIPR is designed to be a system-level reasoner that encompasses multiple levels of a large, complex system, such as aircraft and spacecraft. The wide variety of reasoners employed in such systems span from individual Line Replaceable Unit (LRU) health managers to Area Health Managers (AHM) and the Vehicle Health Manager (VHM). The different health managers are organized hierarchically, and operate in a coordinated manner to derive diagnostic and prognostic inferences from symptoms and conditions reported by a set of Diagnostic and Prognostic Monitors (DMs and PMs). A brief description of the layered VIPR architecture is presented in Section 2.

Existing metrics for evaluating fault detection, fault isolation, and prognostics schemes are directly applicable to the diagnostic monitors (DMs), prognostic monitors (PMs), and the LRU health managers (HMs). For layered reasoners, such as VIPR, the overall performance cannot be evaluated by metrics solely directed toward timely detection and accuracy of estimation of the faults in individual components. Among other factors, the overall vehicle reasoner performance is governed by the effectiveness of the communication schemes between the different monitors and reasoners in the architecture, and the ability to propagate and fuse relevant information to make accurate, consistent, and timely predictions at different levels of the reasoner hierarchy. To address these issues, we outline an extended set of diagnostic and prognostics metrics in this report. These metrics can be broadly categorized as evaluation measures for: (1) diagnostic coverage, (2) prognostic coverage, (3) accuracy of inferences, (4) latency in making inferences, and (5) sensitivity to different fault and degradation conditions.

Our overall approach will involve a systematic study of the effectiveness of the VIPR system using a simulation testbed designed to generate off nominal events corresponding to a number of different fault scenarios. The evaluation studies will involve systematic generation of degradation and fault data, realistic analysis using the set of monitors and reasoners in the VIPR architecture, and a methodology to compute the values for the set of chosen metrics using the performance data collected from the software testbed. The objective of benchmarking VIPR is to assess how it can increase aviation safety. In order to achieve that, we plan to use the Boeing 787 Central Maintenance Computing Function (Christensen, 2010) as a baseline and will evaluate VIPR's performance using the diagnostics and prognostics metrics described in Section 4.

The rest of this paper outlines the VIPR architecture, the simulation testbed for benchmarking studies, the metrics chosen for diagnostic and prognostics reasoner evaluation, the Boeing 787 Central Maintenance Computing Function to be used as a baseline for comparison, and an outline of our evaluation plan.

2. VIPR ARCHITECTURE AND FUNCTIONALITY

The primary function of Honeywell's Vehicle Integrated Prognostic Reasoner (VIPR) is to detect faults and failures at the aircraft level, enable isolation of these faults, and estimate remaining useful life of components in the system. A simplified functional view of VIPR is shown in Figure 1. VIPR is organized into a hierarchical architecture. At each level or layer of the hierarchy, the VIPR processing blocks maintain relationships with other blocks at the same level. Only a subset of messages is allowed from one level to another to satisfy bandwidth and power constraints. At the lower level LRU HMs receive measurements from the sensors and they perform Diagnostic & Prognostic (D&P) monitoring tasks to compute indicators. The next level is organized in multiple Area HMs which follow the principal spatial and temporal decomposition of the aircraft functionality and behavior. The main task of an Area HM is to perform D&P reasoning using the indicators provided by the LRU HMs. Finally, a Vehicle HM is responsible for collecting the data from all Area HMs and solving any ambiguities with the assistance, if necessary, of off-vehicle health management services. In the following, we first describe the candidate algorithms that can be used in VIPR and then discuss the information flow between the various levels of the VIPR.

2.1 LRU Health Management

At the LRU level, the objective is twofold: (1) discover information that can be used for D&P reasoning in the raw and noisy measurements by performing feature extraction and (2) compress the data so that they can be efficiently transmitted and used by the higher levels for more integrated analyses (e.g., reason about the effects of fault propagation between subsystems). Both tasks are accomplished by a suite of D&P monitors that can be classified into two categories: (1) simple D&P monitors and (2) advanced D&P monitors.

Simple D&P monitors test if a sensor measurement or measurement rate exceeds a threshold. All major subsystems in an aircraft have Built-In Tests (BIT) that perform such operations and present the simplest form of feature extraction, generating binary health indicators. Mathematically, these tests are based on well-defined detectors such as likelihood ratio test, z-test, and t-test. In addition to such algorithms, *advanced D&P monitors* are used for discovering and extracting information from multivariable measurement sets. A representative algorithm for such a monitor is Principal Component Analysis (PCA) which transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. After using multivariate signal processing

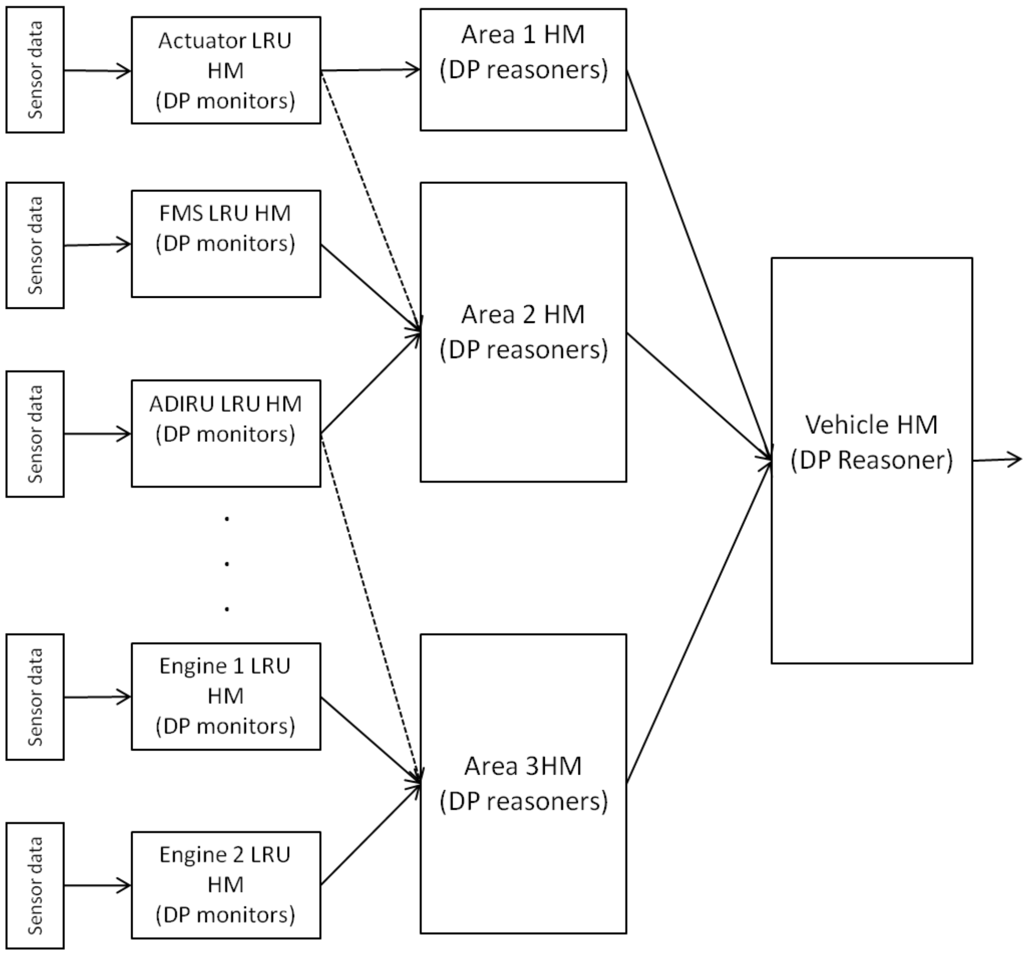

Figure 1: Functional View of VIPR

algorithms, advanced D&P monitors can use simple classification and trending algorithms to encapsulate information to D&P indicators that are forwarded to the Area HMs. Candidate algorithms include bin classifiers, nearest-neighbor classifiers, and discriminant analysis. The computed indicators include: (1) condition indicators that can describe, for example, the engine compressor efficiency and spectral energy content from a vibration signal, (2) health indicators that capture, for example, inlet filter, compressor rub, or foreign object damage (FOD) incidents, and (3) prognostic indicators that show, for example, the evolution of engine health for the next 100 hours of a specific mission.

2.2 Area Health Management

Area HMs are responsible for D&P reasoning for aircraft subsystems that include multiple LRUs. They are organized by taking into consideration the spatial and temporal boundaries of fault manifestation and propagation in order to minimize the communication between aircraft subsystems. Since perfect containment of a fault in one area cannot be guaranteed, Area HMs are capable of querying remote LRUs if necessary. Candidate algorithms at this level include decision trees, discrete event system diagnosers, failure propagation graphs, neural networks, fuzzy classifiers, and Bayesian networks. Heterogeneous reasoners are needed to deal with binary, discrete, and continuous indicators provided by the LRUs as well as with event-driven and time-driven dynamics of the underlying aircraft components.

2.3 Vehicle Health Management

The Vehicle HM is responsible for reasoning across spatial and temporal boundaries of the various areas and possibly uses off-vehicle health management services. The main objective of the Vehicle HM is to resolve ambiguities that may arise from the Area HMs, initiate additional D&P tests, and provide warnings. The D&P reasoning technologies are similar to those in the Area HMs but special care must be taken to deal with multiple temporal scales.

2.4 Reference Models

Managing and evaluating the operation of VIPR requires a database of the health management components that are available at the LRU, Area, and Vehicle levels. Every component is associated with a reference model (implemented as an XML file) that captures the interface of the component and information about the internal functionality if available. At the very least, it defines the input output relations for the associated component.

2.5 Information Flow

Next, we briefly discuss typical information flows in VIPR addressing (1) what is communicated and (2) how the communication is organized. The basic information flow starts from the sensors that communicate raw measurements to the LRU HMs. The LRU HMs compute D&P indicators using simple or advanced D&P monitors and sent them to the Area HMs. The D&P reasoners in the Area HMs generate fault candidates that are sent to the Vehicle HM. At the vehicle level reasoners generate detections and predictions of failure modes and advisories.

In addition to communicating the output of the health management components at this level, VIPR will consider an enhanced information flow that complements the component results with metadata that provide valuable information related to how these results have been computed. Examples of such information include the number and type of sensors used for extracting specific features and the mode of the aircraft at the time an adverse event was triggered. If health management components at the LRU level expose information in terms of the internal algorithms used for computing the results, it can be included also in the communication and utilized by the reasoners. The metadata communicated will instantiate the attributes of the reference model of the corresponding component in the VIPR architecture constructing an accurate runtime representation of the VIPR configuration.

Sensors and LRU monitors typically operate periodically and store relevant information in dedicated non-volatile memory. The information flow then can follow two paradigms. First, low level components can forward important messages to higher levels upon detection, for example, of adverse events. Second, high level components can actively query low level components for specific information that will be used to disambiguate fault candidates or improve fault prediction. In addition, VIPR supports active sensor tests that are invoked on demand.

2.6 Evaluation

Evaluation of a vehicle-level health management architecture, such as VIPR, must assess how it can increase aviation safety. This goal is directly linked to the following measures:

1. Diagnostic coverage,
2. Prognostic coverage,
3. Accuracy,
4. Latency, and
5. Sensitivity.

Benchmarking VIPR performance will be done by comparing these measures for VIPR-generated diagnostics and prognostics with existing state-of-the-art vehicle health management systems. Given VIPR's hierarchical architecture, the benchmarking process must consist of two steps:

1. Quantify the effectiveness of each VIPR in terms of the above metrics, and
2. Determine the accumulated inaccuracies as information is passed up the architecture.

The following section presents well-defined metrics that can be used to evaluate VIPR performance. In addition to such measures, it is important to evaluate the efficiency and scalability of VIPR in terms of the computational resources needed as well as to quantify the trade-offs between performance and resource usage. The enhanced information flow and the active querying described above, for example, can improve performance and increase safety but they require increased computation and communication capabilities.

2.7 Offline Design

VIPR contains various components such as advanced D&P monitors and reasoners that are designed based on data-driven and model-based methods. A secondary but still important aspect to be evaluated is the efficiency of the offline methods for constructing such components. For example, D&P monitors and reasoners will be constructed using: (1) data-driven methods for training neural network and Bayesian network classifiers, (2) model-based approaches for constructing decision tree and failure propagation graph diagnosers, and (3) integrated data-driven and model-based approaches. In addition, the VIPR functionality is the result of the integration of multiple health management components that is achieved through the use of a reference model implemented as database of XML files. A useful evaluation metric is the complexity and the amount of time required to build the reference model.

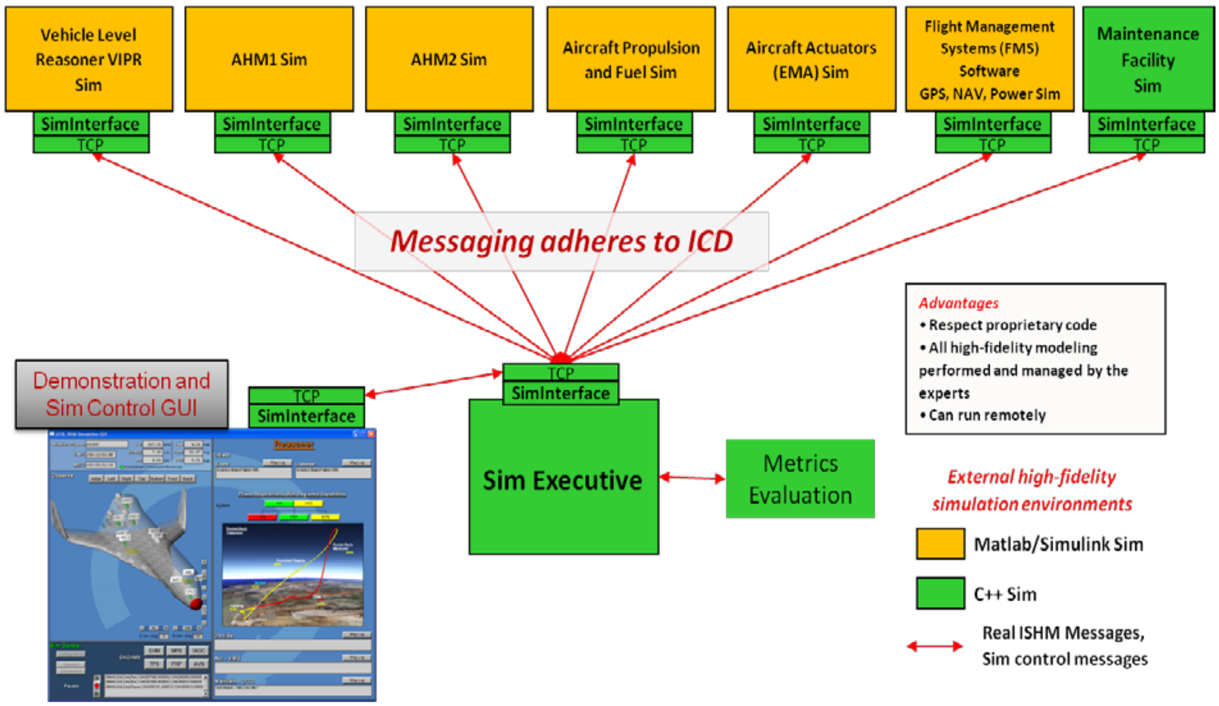

Figure 2: Software architecture for benchmarking

3. SOFTWARE ARCHITECTURE FOR BENCHMARKING VIPR

This section presents the recommended software architecture for benchmarking VIPR and explains how the evaluation methodology will ensure that (1) the comparison with the existing state-of-the art will be performed, (2) a set of representative test scenarios is used, and (3) the data sources reflect realistic conditions.

The performance evaluation and operational assessment will be demonstrated using the SMARTlab modeling and simulation environment. SMARTlab (a division of Honeywell devoted to simulation-based design) will enable performing scenario-based demonstration and evaluation of the integrated detection, diagnostic and prognostic capabilities. In addition, SMARTlab will support the benchmarking of VIPR at multiple levels starting from data file playback within the simulation environment, high-fidelity simulations, and hardware-in-the-loop simulations.

The software architecture that will be used for benchmarking VIPR is presented in Figure 2. The physical system to be simulated consists of various aircraft subsystems. The simulation models of the subsystems capture the dynamics of the subsystems as well as the functionality of the corresponding LRUs. Additional functionality includes the simple and advanced monitors used in the VIPR system and the ability to inject faults for running a variety of test scenarios. The area reasoners and the vehicle-level reasoners will be represented as external high-fidelity simulators. All the models are orchestrated by the Sim Executive via a TCP scheme. A benchmarking software facility responsible for evaluating the diagnostic and prognostic metrics is connected directly to the Sim Executive and has access to all the messages exchanged by the high-fidelity simulators. A graphical user interface for visualizing the results and controlling the simulation is also linked to the Sim Executive.

The testing procedure is scenario-based using the scenarios that cover a spectrum of Adverse Event Types listed in Table 2 of the NASA-IVHM Technical Plan (NASA, 2009). The Simulation will focus on performing the following functions:

- Demonstrate diagnostic and prognostic reasoning necessary for the identified scenarios,
- Send and receive messages necessary to simulate the scenarios,
- Log and capture metrics involved in each scenario,
- Visualize the sequence of events comprising each scenario, and
- Inject fault on-the-fly

A significant factor in evaluating health management software is to use realistic data sources that represent accurately the failure modes and faulty behavior.

4. RECOMMENDED METRICS

We describe how the VIPR performance can be evaluated in terms of (1) its ability to detect and diagnose faults and (2) its ability to predict faults.

4.1 Detection and Diagnosis Metrics

Diagnostic Coverage

The diagnostic coverage characterizes the types of faults that can be detected by a health management system and it is one of the most critical measures that affect aviation safety. Characterization of the coverage should include a list of faults which is as exhaustive as possible. Further, it should include fault attributes such as persistent or intermittent, abrupt or incipient, and fault magnitude. Determining the diagnostic coverage of a vehicle-level health management architecture requires prohibitively high time and cost. Instead, the recommended approach in VIPR is to compare with state-of-the–art vehicle health management systems and evaluate how VIPR improves coverage by performing testing using a representative set of scenarios. Diagnostic coverage improvement can be assessed in the following ways:

1. Identify test scenarios with faults that could not be detected and/or isolated with existing approaches and demonstrate VIPR's effectiveness for these scenarios.

2. Identify test scenarios with faults that can be successfully detected and isolated with existing approaches and demonstrate the improvement in terms of accuracy, latency, and sensitivity.

Accuracy

In order to evaluate the accuracy of VIPR, we will focus on the representative scenarios and define measures based on statistical methods that utilize a comprehensive appropriately designed set of tests. The test selection will ensure unbiased estimation of the metrics in terms of aircraft operating condition, fault magnitude, sensor noise, and other uncertainties. Given a set of tests, accuracy can be characterized using the following metrics:

1. *Detection false positive rate:* The probability of rejecting the null hypothesis when it is actually true (no fault) among all tests performed (also known as type I errors).

2. *Detection false negative rate:* The probability of failing to reject the null hypothesis when it is actually not true (there is a fault) among all tests performed (also known as type II errors)

3. *Isolation misclassification rate:* A well known method is based on the construction of a confusion matrix (Button and Chicatelli, 2005) that summarizes diagnostic results produced by a reasoner over a number of test/use cases. The results can then be summarized using a number of measures. For example, (Biswas et al., 1997) developed two measures for single fault diagnosis: (i) *accuracy,* which measures how close the diagnosis comes to generating the actual fault candidate; and (ii) Resolution, which refers to the number of candidates generated, i.e., an algorithm that generates a large number of candidates has *poor resolution*, whereas an algorithm that generates only one candidate has *high resolution.* For multiple fault situations the Kappa Coefficient (Button and Chicatelli, 2005) is used to measure the ability of an algorithm to discriminate between multiple fault candidates.

Latency

Latency measures will be computed also using statistical methods based on a set of tests and statistical analysis to compute average and worst case values as well as the variance.

1. *Time to detect:* Time interval between fault injection and detection.

2. *Time to isolate*: Time interval between fault injection and isolation

Sensitivity

Test design and selection must ensure that the estimation of the above metrics is unbiased in terms of system uncertainty. Further, VIPR effectiveness must be evaluated in terms of various system dynamic parameters and environmental disturbances such as load, wind, etc.

4.2 Prognosis Metrics

The objective of the prognostics VIPR reasoner is the detection of a failure precursor followed by the prediction of remaining useful life (RUL). RUL estimation is based on statistical methods for predicting the time of failure and a confidence interval associated with the prediction.

Prognostic Coverage

Our approach for evaluating the prognostic coverage of VIPR will be similar to the diagnostic coverage. Specifically, we will

1. Identify test scenarios with faults that could not be predicted with existing approaches and

demonstrate VIPR's effectiveness for these scenarios.

2. Identify test scenarios with faults that can be successfully predicted with existing approaches and demonstrate the improvement in terms of accuracy, precision, and sensitivity.

Accuracy

In order to evaluate the prognostic accuracy of VIPR, we will focus on representative scenarios and utilize metrics for evaluating the performance of prognostic techniques (Saxena et al., 2008). Specifically, given a set of tests, prognostic accuracy can be characterized using the following metrics:

1. *Error = predicted RUL – actual RUL.* The error compares the predicted value to the actual RUL value. Several variations of the basic notion of the error can be used to formally define metrics that average the predictions for multiple units over a prediction horizon. Examples include the mean squared error (MSE), mean absolute error (MAE), and root mean squared percentage error (RMSPE).

2. *Average bias.* The average bias is a measure of how close the RUL estimate is to the actual failure time and it is computed by averaging the prediction errors over the prediction horizon.

3. *Timeliness*. Timeliness metrics exponentially weigh RUL prediction errors by penalizing late predictions more than early predictions. They can be extended to assess false positives and false negatives by specifying acceptable ranges for prediction.

Precision

Precision measures estimate the size of the confidence interval associated with the RUL prediction. The metrics are defined based on the variance of the predicted results for a set of experiments. Typical mathematical definitions assume a normal distribution of the error and estimate the size of the confidence interval. Special care can be taken for cases when the error plots do not resemble a normal distribution by using appropriate dispersion metrics (Saxena et al., 2008).

Sensitivity

Sensitivity metrics measure how sensitive the prognostic capabilities are in terms of system inputs and environmental disturbances. Assuming a nominal value for an input/disturbance, sensitivity can be assessed against any performance metrics by evaluating how the performance changes as a function of the distance between inputs/disturbances.

5. COMPUTATIONAL METRICS

The metrics described in the previous section can directly evaluate how VIPR can increase aviation safety. However, metrics that measure the computational performance of VIPR are also needed to demonstrate the requirements in terms of computational resources for implementing and deploying VIPR. In order to benchmark the computational performance of VIPR, we will perform (1) offline analysis of the hierarchical architecture to compute a vector of complexity metrics and (2) profiling of the run-time execution of VIPR to measure the performance of the VIPR implementation.

5.1 Offline Complexity Analysis

Complexity analysis of algorithms is used in theoretical computer science to predict the resources that the algorithm requires in terms of running time, memory, and communication bandwidth. Typically, worst- or average-case estimates as a function of the size of the algorithm input are derived. We will analyze the complexity of the D&P monitors and reasoners and ensure that there are no unrealistic assumptions required for implementing VIPR.

Additional metrics that can be used to assess the complexity of VIPR are the number of software components, the number of links between these software components, the number of inputs and outputs that need to be communication between components, and the size of the code.

5.2 Online Profiling

In addition to offline analysis, it is important to perform online profiling for measuring the VIPR computational performance. This is an important task especially because of the timeliness requirements of health management software. Profiling measures not only the computational performance of the software implementation but the combined performance of the hardware-software deployment. Since it is important to measure the computational performance before the system is actually deployed in an aircraft, the recommended approach is to fix the hardware configuration and compare VIPR's performance against existing health management systems. Such an approach can identify if and when additional computational resources are needed. Metrics that need to be measured include (1) CPU execution times, (2) CPU utilization, (3) network delays, (4) bandwidth utilization, and (5) amount of memory used.

The utility of onboard health management services typically increases with the frequency of the execution, especially for components that are executed periodically with real-time input data streams. A useful metric of the computational performance is how fast VIPR can be executed while keeping up with the generation of the real-time data sources.

6. BASELINE: BOEING CENTRAL MAINTENANCE COMPUTING FUNCTION SUMMARY

The objective of benchmarking VIPR is to assess how it can increase aviation safety. In order to achieve that, we plan to evaluate VIPR's performance against the Boeing 787 Central Maintenance Computing Function (CMCF) (Christensen, 2010). We will follow the scenario-based testing procedure outlined in Section 3 and we will evaluate the diagnostic and prognostics metrics presented earlier in order to quantify the improvement due to VIPR's capabilities. It should be noted that some of the CMCF functionality (e.g. the ground operations) will be left out since VIPR does not offer this capability and comparison would not be useful. Comparison will be performed by identifying improved capabilities and coverage of VIPR against CMCF and evaluating the improvement in terms of the D&P metrics against scenarios documented in CMCF.

6.1 System Overview

CMCF software is installed in the Common Core System (CCS) of every 787 aircraft. CMCF functionality is configured via three separately loadable databases: Option Selection Software (OSS), a Loadable Diagnostic Information (LDI) database, and a field-loadable CMCF Airline Modifiable Information (AMI) file. A baseline version of each of these databases is provided by Boeing with the aircraft. Aircraft operators may re-configure certain functions/behaviors of CMCF by modifying and re-loading the AMI file using the Boeing-provided Ground Based Software Tool (GBST).

The CMCF is composed entirely of software that executes in the CCS. Four software components are required for CMCF to function. Each is separately loadable and may be updated independently:

1. Central Maintenance Computing Function (CMCF) Partition - one instance residing on one GPM.

2. Option Selection Software (OSS) database. The OSS is a separately loadable database that is accessed by the CMCF partition.

3. Loadable Diagnostic Information (LDI) database. The LDI is a separately loadable database that is accessed by the CMCF partition.

4. Airline Modifiable Information (AMI) database. The AMI is a separately loadable database that is accessed by the CMCF partition.

6.2 System Functions

The CMCF is responsible for the management of health information on the airplane. This includes receipt and processing of fault reports, display and storage of maintenance messages, and support for other maintenance activities, such as control and display of initiated tests, control and display of onboard engine balancing activities, and retrieval of LRU fault history data. Additionally, the CMCF is responsible for retrieval, display, and reporting of airplane system configuration information. The CMCF gathers configuration information that is reported by the airplane member systems, and formats the data for display on a maintenance terminal.

7. EVALUATION PLAN

Our objective is to build a testbed simulation that will allow us to evaluate first the software and eventually the hardware of the VIPR system. Our testbed will provide us the ability to evaluate VIPR against the metrics listed above. The VIPR testbed will implement and evaluate the basic "plumbing" of the system. Inter- and intra-layer message passing will be implemented according to the message passing protocols defined earlier in the program. This will provide the ability to evaluate quantities such as timing, network delays, bandwidth utilization, and so on. We also plan to add hardware in the loop simulation and bring the software to a higher TRL. This will allow us to evaluate the VIPR software on actual flight hardware and to evaluate the system with real aircraft subsystems, e.g. flight surface actuators. Measurements of injected faults, signal path timing, bandwidth, real time behavior, etc. will be more accurate and informative.

ACKNOWLEDGMENT

This work is supported by NASA under contract number NNL09AD44T.

REFERENCES

Byington, C.S., Watson, M., Kalgren, P., & Safa-Bakhsh, R. (2003). Metrics Evaluation and Tool Development for Health and Usage Monitoring System Technology, *Third International Conference on Health and Usage Monitoring - HUMS2003*, Melbourne, Australia, G.F Forsyth (Ed.), pp. 27-35.

Kurtoglu, T., Mengshoel, O, & Poll, S. (2008). A Framework for Systematic Benchmarking of Monitoring and Diagnostic Systems, *International Conference on Prognostics and Health Management Conference*, Denver, CO.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for

Evaluating Performance of Prognostic Techniques, in *International Conference on Prognostics and Health Management (PHM08)*, Denver CO.

Leao, B.P., Yoneyama, T., Rocha, G.C., & Fitzgibbon, K.T. (2008). Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit. *Proceedings of the International Conference on Prognostics and Health Management (PHM08)*, Denver, CO.

Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., Kuhn, L., de Kleer, J., van Gemund, A., & Feldman, A. (2010). Towards a Framework for Evaluating and Comparing Diagnosis Algorithms. *Proceedings of the 20th International Workshop on Principles of Diagnosis*, Stockholm, Sweden.

Christensen, M. (2010). *Boeing 787 Central Maintenance Computing Function Summary*, Technical Report, Honeywell Labs.

NASA (2009). Integrated Vehicle Health Management Technical Plan, Version 2.03, National Aeronautics and Space Administration.

Button, R.M. & Chicatelli, A. (2005). Electrical Power System Health Management, *Proc. 1st International Forum on Integrated System Health Engineering and Management in Aerospace*, Napa, CA.

Biswas, G., Kapadia, R., & Xu, X.W. (1997). Combined Qualitative–Quantitative Steady-State Diagnosis of Continuous-Valued Systems, *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 27(2), pp. 167-185.