Fault Classification with Gauss-Newton Optimization and Real-Time Simulation

Byoung Uk Kim, Chris Lynn, Neil Kunst, Sonia Vohnout
Ridgetop Group, Inc.
3580 West Ina Road
Tucson, AZ 85741
520-742-3300
tim.kim, chris.lynn, neil.kunst, and
sonia.vohnout@ridgetopgroup.com

Kai Goebel
NASA Ames Research Center.
MS 269-4
Moffett Field, CA 94035
650-604-4204
kai.goebel@nasa.gov

Abstract[1]—
Anomaly diagnostics and fault classification with prognostics is an active research topic, and real-time detection of anomalies and their classification has remained a critical challenge to be overcome. We developed an innovative, model-driven anomaly diagnostic and fault characterization system for electromechanical actuator (EMA) systems to mitigate catastrophic failures. The efficacy of the Model-based Avionic Prognostic Reasoner (MAPR) approach has been proven in real time using test data acquired from a MIL-STD-1553 testbed. Receiver operating characteristic **(**ROC) curves are generated as a result of this study to show the tradeoff between sensitivity and specificity. Results of model optimization and fault classification are also presented. This real-time processing will enable enhancements in flight safety and condition-based maintenance (CBM). Once this system is completely mature, flight safety will be improved by allowing the on-board flight computers to read from the MAPR and update their control envelope based on its evaluations of the hardware health, reducing damage propagation, decreasing maintenance time, and increasing operational safety.

TABLE OF CONTENTS

1. INTRODUCTION..1
2. MODEL-BASED AVIONIC PROGNOSTIC REASONER (MAPR)..2
3. ANOMALY DETECTION..3
4. MODEL OPTIMIZATION..4
5. FAULT CLASSIFICATION..6
6. CONCLUSION..7
REFERENCES..7
BIOGRAPHY..8

1. INTRODUCTION

For space-bound and other mission-critical flight operations, failure in the field is not an option. With millions of dollars invested in satellites, rockets, and spacecraft, the expense of such failures can be costly at best and catastrophic at worst. Vital to successful operation of these craft is the no-fault performance of critical electronics and the ability to constantly monitor the health of these systems.

Ridgetop designed a MIL-STD-1553 bus monitor and a MIL-STD-1553 bus controller to simulate an aircraft data bus, read the environmental (i.e., altitude) and operational (i.e., response of system) data of a system, and determine whether a fault is manifesting; and if true, determine the root cause and symptoms of the fault. Once an anomaly is detected, the Model-based Avionic Prognostic Reasoner (MAPR) solves a user-outlined state-space model, symbolically, using a Gauss-Newton [17] optimization method and the information from the MIL-STD-1553 bus. This algorithm outputs a list of best-fitting parameters to match the command to the actual performance. Then rules are programmed in, based on results from principal component analysis (PCA) [13]. These rules determine both fault mode and the severity of that fault. The rules can distinguish multiple failure modes, such as mechanical jam and MOSFET failure.

The anomaly analysis approach presented in this paper has foundations in model-based analysis. Model-based analysis uses physics knowledge of the system to design models to process and evaluate the current data. These approaches include residual-based [4] [7], multiple models [5], decoupling [15] [6], and hypotheses [1]. In contrast, dta-driven approaches rely mainly on observation of system data, including data mining [14] [10] [2] [8], expert system [11] [3] [9] [12], and other methods. In this paper we present model-based analysis with a proposed health distance and grey-box model optimization algorithm. We also consider the support vector machine (SVM) approach, which has a particular strength on representing boundaries of varying state of health (SoH) measurements, and PCA to see the relationship between multiple variables using eigenvalue decomposition.

The advanced Model-based Avionic Prognostic Reasoner (MAPR) described in this paper features a passive connection to common avionic data buses. Aircraft operators and control staff will have visibility of the health of critical flight control systems and components. Possible applications areas transcend a broad array of commercial segments such as the aerospace and automotive industries.

[1] 978-1-4244-7351-9/11/$26.00 ©2011 IEEE

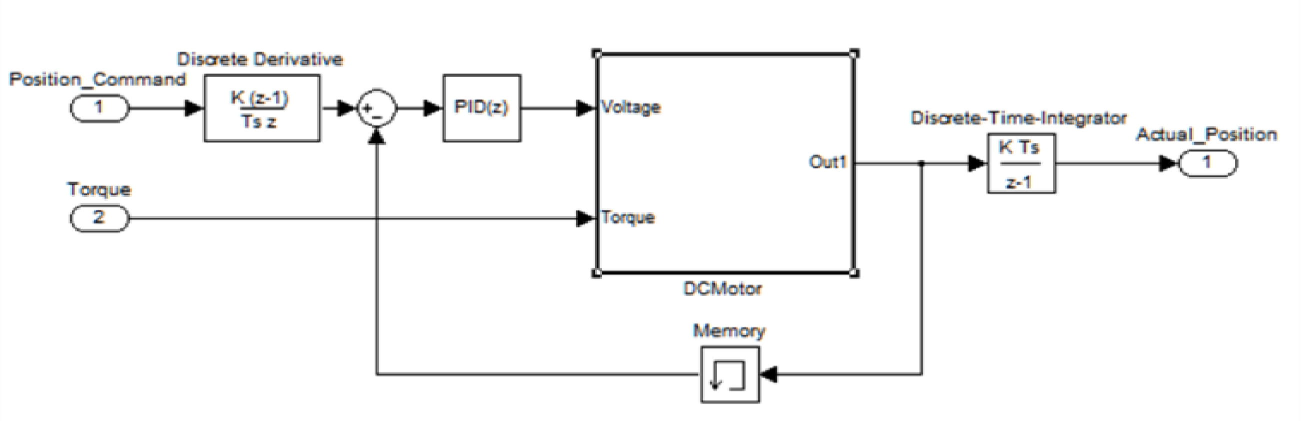


Figure 1: RTWEC motor model

The rest of this paper is organized as follows: In section 2, we explain the Model-based Avionic Prognostic Reasoner (MAPR) with model formulation and dynamic linked libraries. Section 3 explains anomaly detection by comparing the actual and commanded position with performance metrics. Section 4 provides model optimization with defined structure and shows our algorithm control GUI. Principal component visualization of MAPR results are presented in section 5. Section 6, concludes the paper.

2. MODEL-BASED AVIONIC PROGNOSTIC REASONER (MAPR)

Ridgetop developed the Model-based Avionic Prognostic Reasoner (MAPR) that runs on a testbed comprising a MIL-STD-1553 bus monitor and a MIL-STD-1553 bus controller. The controller simulates the aircraft data bus and the monitor, which reads in signals. The monitor has an open architecture that allows deployment and integration of our models.

Ridgetop had previously developed a, model-driven anomaly diagnostic and fault characterization system for electromechanical actuator (EMA) systems to mitigate catastrophic failures: the EMA2000 suitcase testbed [16]. A data file containing the fault-testing results from the EMA2000 was added to the MIL-STD-1553 testbed to simulate the 1553 data from an airplane. These data consist of eight columns of 313 rows of numbers representing the actual and commanded position of the EMA2000 servo motor. While it currently does not gather the dynamic flight data from a 1553 bus, it will result in the development of a portable, demonstrable and flight-test-capable tool that can make a strong case for this technology.

Trapezoids with amplitude were used to be a scaled match of our existing EMA test bed motion profiles. These data are easily logged in a comma-separated volume that is portable to several different modeling environments.

2.1 Model Formulation

The system model had to evolve to lower the effective runtime without sacrificing control of critical parameters. Therefore we built a reconfigurable state-space system model. To make sure that the MAPR system is developed with future usability in mind, the model was given a position to move to and the output was returned to the caller. The model now allows the hardware bus controller to send messages, the bus monitor to receive and log them, software to generate numeric data as a trajectory profile, and for the model to run that command. The state-space model is developed in Real-Time Workshop® Embedded Coder™ (RTWEC), shown in Figure 1.

2.2 Dynamic Link Libraries (DLLs)

Ridgetop's vision in this project was to create highly detailed actuator models in software and then build a dynamic link library (DLL). That DLL was deployed to an embedded system that supports the Windows DLL functions. Creating the DLL required type casting, runtime analysis, and solver configuration. The DLL of the model was then added to the bus monitor (BM) module, enabling data to be received via the 1553 bus. The data are modeled in a point-by-point system architecture; when each point simulation is finished, the plot is updated.

The bus controller (BC) acts as the agent that sends data around the bus in hexadecimal "words." The controller can be configured to send and receive "messages" from 30 remote terminals (RT). The instructions in those messages are ordered by a "subaddress," which allows for certain instructions to be filtered easily by the RT. For the initial setup a few unique words were sents from the bus controller to the remote terminal.

The RT reads in words from the bus and has adjustable filters to allow for the full control of data coming in, and to automate responses. RT is different from the bus monitor (BM) mode because RT devices send responses. In the case of the MAPR, the device must respond with the results of its calculation.

The DLL is set up and controlled in software to check its validity and final runtime. Ridgetop created a simple user interface to control the library with position and torque inputs. However, the feedback controller design has been left out of this simple motor check program. Notice in Figure 2 that the x-axis is in milliseconds. The simulation time has been set to 2 times the maximum theoretical rate of 50 Hz that the 1553 bus can run after adhering to the Nyquist rate.

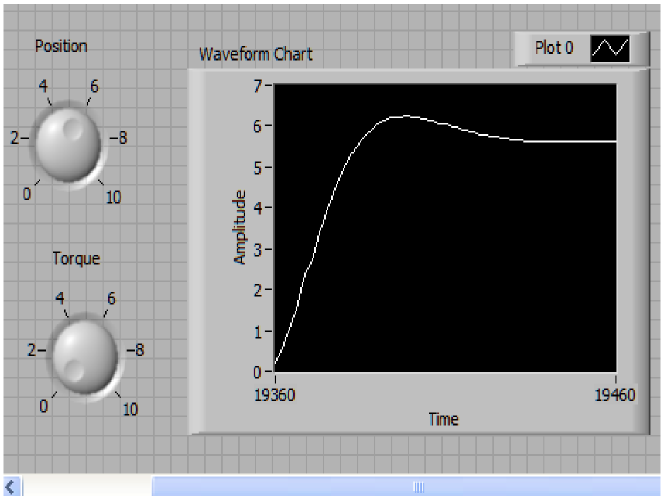


Figure 2: RTWEC DLL result

The support vector machine (SVM) [16] is used to classify different groups of data. As two groups of data points, which represent repeated samples of two parameters, begin to move away from each other, an error is occurring (Figure 3). This SVM is also intended to classify data that are not associated with either group, like an outlier or anomaly. The rules that define the conditions for an outlier define the fault type in the system.

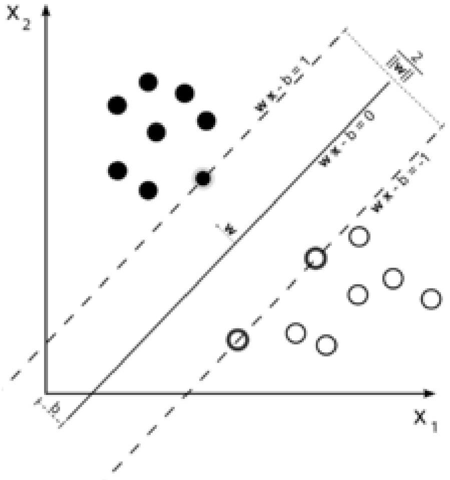


Figure 3: Support vector machine classifying two parameters

3. Anomaly Detection

The first step in the algorithm is to compare the actual (D1) and commanded (D2) position for anomaly detection. We present the current data analysis based on the principle that normal system responses can be tested so that they are measurably different. The difference between a normal signal and an anomalous signal is called the Health Distance™. These “signals” at the time of analysis are not in the time domain.

Anomaly detection is calculated with D2, relative to D1. First, all data are categorized into ω pieces and the sum of all the dot products is equal to that of the dot product over the entire array (equation 1):

$$\langle D_1(\omega), D_2(\omega)\rangle \therefore D_1 \bullet D_2 = \sum_{\omega} D_1(\omega) \bullet D_2(\omega) \tag{1}$$

Now an angle is introduced to represent the difference between the two arrays:

$$\theta(D_1, D_2) = \cos^{-1}\left(\frac{D_1 \bullet D_2}{\|D_1\| * \|D_2\|}\right) \tag{2}$$

$$0 \le \theta \le \pi / 2$$

A returned value of 0 would indicate that the two signals are identical; a score of π/2 indicates that there are no similarities at all. The implementation of this process, in the time domain, results in large fluctuations in the angle of difference between two signals.

This portion of the code also uses the information available to calculate the instantaneous load that the actuator must drive. That result will weigh into the decision on whether to continue analysis, because it will show whether a fault is present or just a large load variation.

3.1 Performance Metrics

The false negative rate is the percentage of abnormal flows incorrectly classified as normal. True negative rate is the percentage of the normal flows that are correctly classified as normal. False positive rate is the percentage of normal flows incorrectly classified as abnormal. When the abnormal event is detected (positive), it is not a real abnormal event (false):

$$\textit{False positive rate} = \frac{\text{False positive}}{\text{False positive + True negative}} \tag{3}$$

The true positive rate showed in experiment results is the percentage of abnormal flows correctly classified as abnormal. When the abnormal event is detected (positive), the abnormal event is a real abnormal event (true). Failure detection situations are classified in Table 1.

$$\textit{True positive rate} = \frac{\text{True positive}}{\text{False negative + True positive}} \tag{4}$$

Table 1: Classification of Failure Detection Situations

		Real	
		Abnormal	**Normal**
Detected	**Abnormal**	TruePositive	FalsePositive
	Normal	FalseNegative	TrueNegative

To show the tradeoff between sensitivity and specificity of our algorithm, we provide a partial ROC curve, Figure 4, for the anomaly detection algorithm.

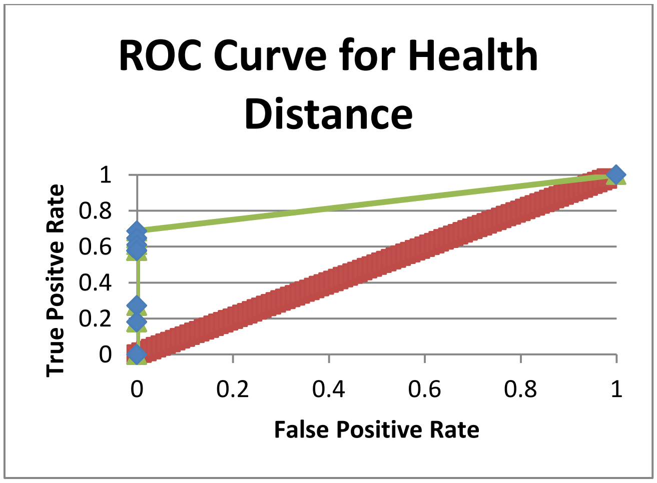


Figure 4: Receiver operating characteristic curve for anomaly detection

The lack of any points between 0 and 1 on the ROC curve is due to 0 false positives at thresholds below 99% matching and not enough resolution between 99 and 100. The one point in the upper right is when a 100% threshold is applied, and no signal would pass as healthy in this case. It is encouraging to see that the false positive rate stays at 0 for a number of true positive values. At this point no judgment can be made regarding performance over the 99%.

4. Model Optimization

Grey-box Gauss-Newton algorithms allow for the optimization of a model with defined structure, while imposing strict conditions on the variables that can be adjusted for a minimization of the error between the inputs and outputs of that model. An understanding of any system's dynamics can be represented using ordinary differential equations (ODE) with unknown parameters. Grey-box model equations specify the mathematical structure of the model explicitly, including couplings between parameters and known parameter values. Grey-box modeling is useful when the relationships between variables, constraints on model behavior, or explicit equations representing system dynamics are known.

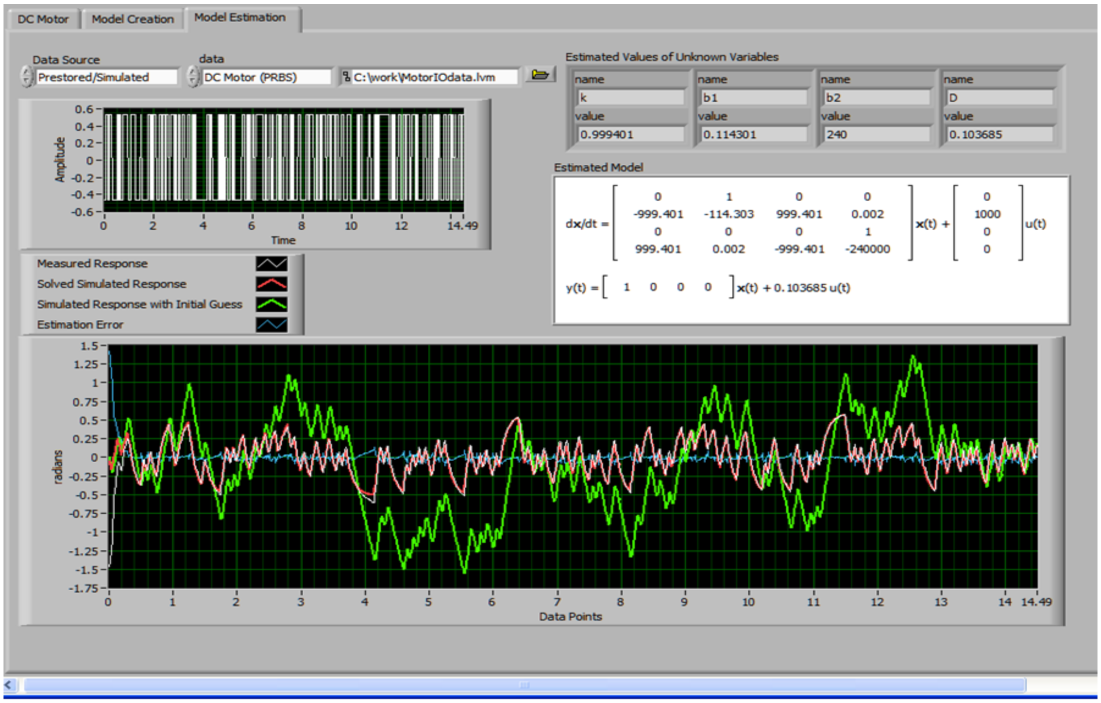
A grey-box model consists of a combination of a mechanistic (first principle) model and an empirical (black-box) model. The form is based on the ordinary differential equation:

$$\dot{x} = f_{FP}(x,u,f_{EM}(x,u)) \tag{5}$$

which contains first principle equations f_{FP} which describe the interaction of the model states x, inputs u and the outputs of an empirical model f_{EM}. Such models are usually derived from conservation laws and balance equations but are, without exception, reduced to lumped models of low order. In our approach, the general partial differential equation (PDE) including empirical term:

$$\frac{\partial T}{\partial t} = A(T) + B(u) + F(T,u) \tag{6}$$

is considered. Here $T(..,t)$ denotes the state variable at position x in some spatial geometry Ω and at time t, $u(x,t)$ denotes the input. For all t, $T(..,t)$ is assumed to belong to a Hilbert space H; A is a linear operator $A: D(A) \to \mathrm{H}$ where $D(A) \subset \mathrm{H}$ is the domain of A; B denotes the input operator and F represents nonlinear terms and model mismatch. The system is separated in two parts:



Data Source: Prestored/Simulated

data: DC Motor (PRBS)

C:\work\MotorIOdata.lvm

Estimated Values of Unknown Variables

name	k	b1	b2	D
value	0.999401	0.114301	240	0.103685

Estimated Model

$$d\mathbf{x}/dt = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -999.401 & -114.303 & 999.401 & 0.002 \\ 0 & 0 & 0 & 1 \\ 999.401 & 0.002 & -999.401 & -240000 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1000 \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + 0.103685\, u(t)$$

Figure 5: Grey-box algorithm control GUI

$$\frac{\partial T}{\partial t} = A(T) + B(u) + q \qquad (7)$$

$$q = F(T,u)$$

where the nonlinear function $F(T,u)$ is viewed as the (known or unknown) empirical part of the model.

A dynamic motor model can be represented by the system shown in Figure 6.

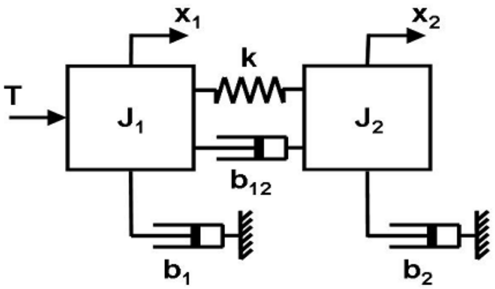


Figure 6: Motion dynamics representation for a motor

To set up the grey-box model, we derive the equations for the motion of the system shown in the following:

1. Rearrange and collect terms
2. Normalize
3. Add additional states
4. Write in State-Space Form ==>

$$\begin{bmatrix} x1' \\ x1'' \\ x2' \\ x2'' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{k}{J1} & \dfrac{-(b1+b12)}{J1} & \dfrac{k}{J1} & \dfrac{b12}{J1} \\ 0 & 0 & 0 & 1 \\ \dfrac{k}{J2} & \dfrac{b12}{J2} & -\dfrac{k}{J2} & \dfrac{-(b2+b12)}{J2} \end{bmatrix} * \begin{bmatrix} x1 \\ x1' \\ x2 \\ x2' \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{J1} \\ 0 \\ 0 \end{bmatrix} *T$$

$$\begin{bmatrix} PositionActuator \\ PositionLoad \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} x1 \\ x1' \\ x2 \\ x2' \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} *T$$

Once the system has been formatted as above, the actual system hardware parameters can be filled in to define the known parameters. Then Ridgetop's algorithm candidate can calculate the parameters that fit real data.

Assuming that the first step of the algorithm process determines that a fault exists, two signals (the actual and commanded position of an actuator) are written to a file where they can be accessed. The file with the input and output data is specified by the user via the grey-box algorithm control interface shown in Figure 5.

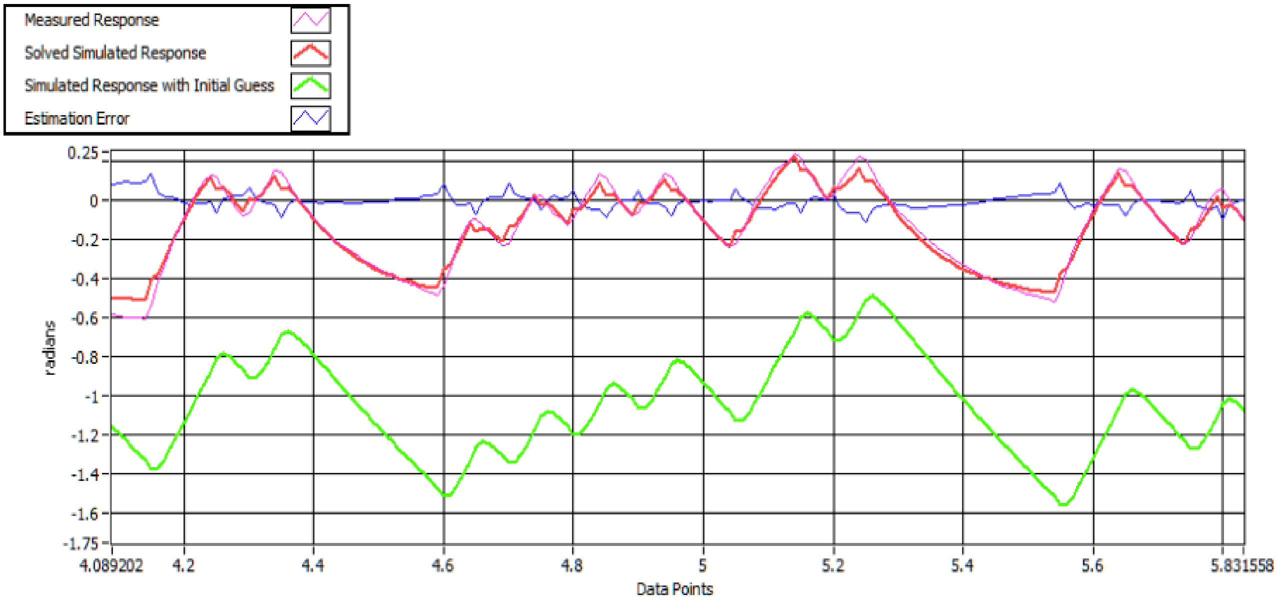


Figure 7: Close-up of grey-box optimization results

Figure 7 shows the response of the "initial guess" in green. The initial guess is rather far from the desired response and the red line is the final simulated response after the parameter estimation has found a suitable solution. The recorded "actual" response is the pink line and the *following error* is in blue. The plot shows that the grey-box algorithm was able to match the desired response.

The resulting parameters of the grey-box method and parameter value solver appear in the user interface. Each cell of the matrix is turned into an equation, each with one unknown, and a simple parameter change solver routine is used to extract the value of the unknowns.

5. Fault Classification

The fault classification was completed by writing the parameters of the grey-box solver to a file then using software to plot those values over time. Figure 8 shows a repetition in the D-matrix (purple) value, which is showing negative energy escaping the system, so an external force is being applied other than the command signal.

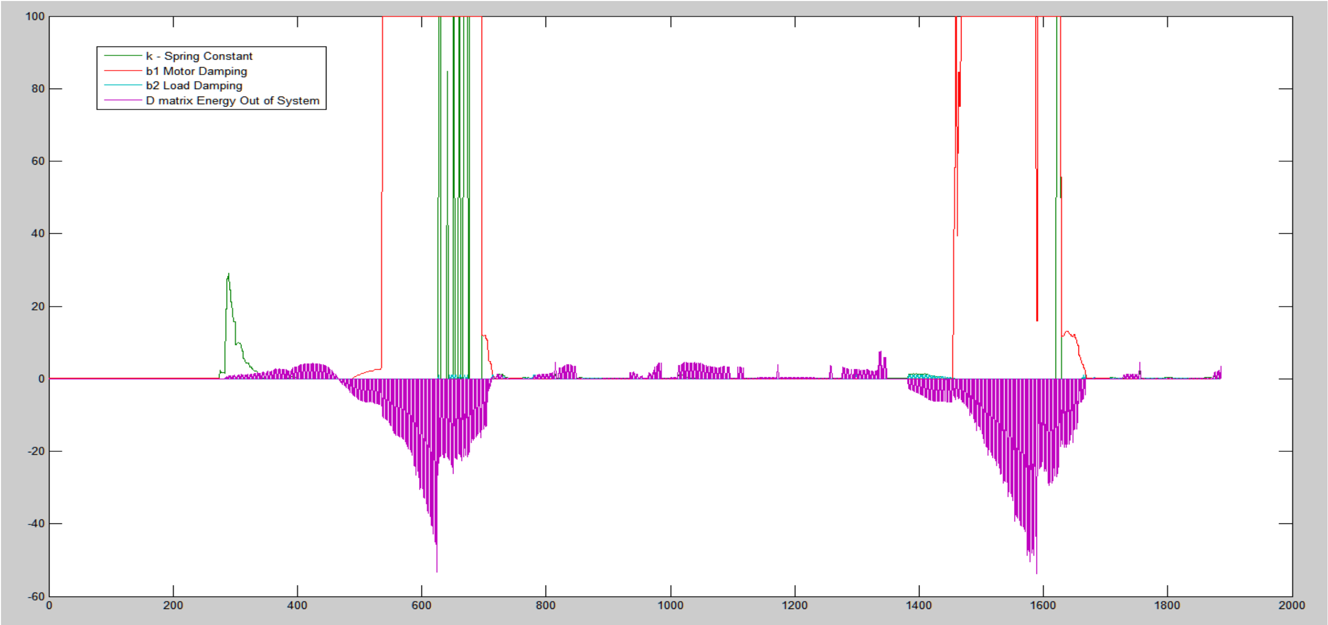


Figure 8: Graphical representation of MAPR values through a repeating fault

As the fault repeats we can examine the values of the system and describe the relationship between all the parameters when a fault is present. For example, Figure 9 shows the relationship between k, b1 and b2 of the model. Notice the cluster in the upper right of the plot; the assumption is that it corresponds to a particular fault mode.

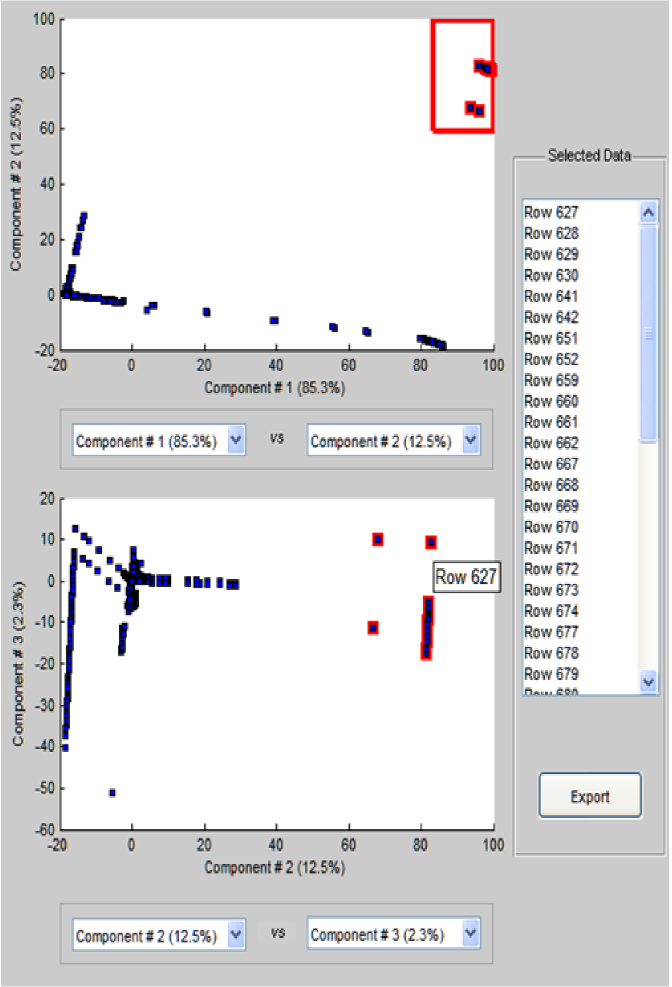


Figure 9: Principal component visualization of MAPR results

The PCA approach is an excellent research instrument as it allows us to see the relationship between multiple variables. New rules for a fault mode must still be programmed in, but the conditions can be determined by running failure data through the MAPR and plotting the results in the PCA approach. We also added a fault table to track all the faults detected during the operation of the MAPR. In future development, we plan to combine this fault table with the "Estimated Values of Unknown Variables." This data will be input to Ridgetop's Sentinel Network™ platform for MAPR to track these values for an extended period of time.

6. CONCLUSION

We have presented that MAPR shows the efficacy of our approach in real time using test data acquired from a MIL-STD-1553 testbed. The result from ROC shows a very low false positive rate and high true positive rate. In our approach, the grey-box model allows us to optimize a model with defined structure, while imposing strict conditions on the variables. Fault classifications with the grey-box solver have explained in principal component visualization of MAPR. We expect the advanced MAPR described in this paper to become a very important embedded prognostic reasoner with a passive connection to common avionic data buses. The MAPR real-time processing will allow for several critical evolutions in flight safety and provide IVHM and CBM support strategies. Flight safety can be improved by allowing the on-board flight computers to read from the MAPR and update their control envelope based on its evaluations, reducing damage propagation and increasing operational safety. We will continue research into aging and providing degradation data sets to lend new insights into failure modes. Ridgetop's Sentinel Network software is a ground-based application that provides an easy-to-understand fusion of parameters and required maintenance, so maintenance personnel can have an extra angle of insight into the systems for which they are they are responsible.

ACKNOWLEDGMENT

This work was in part funded by NASA Phase 1 SBIR NNX10CC03P "Physical Modeling for Anomaly Diagnostics and Prognostics."

REFERENCES

[1] M. Basseville and I. Nikiforov. Fault isolation for diagnosis: nuisance rejection and multiple hypotheses testing, 15th IFAC World Congress, Barcelona, IFAC, July 2002.

[2] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and knowledge discovery, Vol. 2. Number 2, 1998.

[3] W. Becraft and P. Lee. An integrated neural network/expert system approach for fault diagnosis. Computers and Chemical Engineering 17 (10), pp.1000-1014, 1993.

[4] A.S. Willsky. A survey of design methods for failure detection in dynamic systems. Automation (12), pp.601-611, 1976

[5] Y. M. Zhang and X. R. Li. Detection and diagnosis of sensor and actuator failures using IMM estimator. AES-34(4), pp.1293-1312, Oct, 1998.

[6] N. Viswanadham and R. Srichander. Fault detection using unknown-input observers. Control Theory and Advanced Technology 3, pp. 91-101, 1997.

[7] J. Chen and R.J. Patton. Robust Model-based Fault Diagnosis for Dynamic Systems. Boston, MA, Kluwer,

1999.

[8] B. U. Kim and S. Hariri, "Anomaly-based Fault Detection System in Distributed Systems." Paper presented at the 5th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications (IEEE SERA), Korea, August 2007.

[9] S. H. Rich and V. Venkatasubramanian. Model-based reasoning in diagnostic expert systems for chemical process plants. Computers and Chemical Engineering 11, pp.111-122, 1995.

[10] B.Scholkopf, R.C. Williamson, A.J. Smola, J. Shawe Taylor, J. Platt. Support vector method for novelty detection. Neural Information Processing Systems, pp 582-588, 2000.

[11] D. Chung and M. Modarras. A method of fault diagnosis: presentation of a deep knowledge system. AIChE National Meeting, New York, NY, November, 1987.

[12] B. U. Kim and S. Hariri, "Abnormality-based Fault Detection in Distributed System", International Journal of Computer and Information Sci. 8:3 2007.

[13] J Yang, J-Y Yang, D Zhang, and J-F Lu, "Feature Fusion: Parallel Strategy *vs* Serial Strategy," Pattern Recognition, Vol 36, pp 1369-1381, 2003.

[14] I.H.Witten and E.Frank. Data Mining: Practical Machine Learning Tools and Technique with Java Implementations, Morgan Kaufman Publishers, 2000.

[15] J. Chen, R.J. Patton and H. Zhang. Design of unknown input observers and robust fault detection filters, International Journal of Control, vol. 63, no.1, pp. 85-105, 1995.

[16] B. U. Kim, C. Lynn, N. Kunst, T. Dudgeon. Pattern Analysis in Real Time with Smart Power Sensor, IEEE Aerospace Conference (IEEE AIAA 2010) , Big Sky, Montana, USA, March 2010.

[17] D. Foresee and M. Hagan. Gauss-Newton Approximation to Bayesian Learning, International Joint Conference on Neural Networks, 1997.

BIOGRAPHY

Byoung Uk Kim, Ph.D. *is a Senior R&D Engineer and a project lead for reliability analysis tool development at Ridgetop Group. The field of interest for his doctoral program was fault detection and root cause analysis systems, electronic prognostics, data mining and data analysis, and self-healing algorithms with autonomic computing. His collegiate repertoire also consists of numerous published papers in reliability analysis and autonomic configuration. Dr. Kim worked on a key NASA reliability/prognostics project in 2006 for Ridgetop. He has contributed to the development of innovative solutions that are currently deployed in the NASA ADAPT program at the Ames Research Center.*

Chris Lynn *is an Electrical Engineer at Ridgetop Group. His focus is in determining reliability of critical systems and predicting their failures. Chris graduated from the University of Arizona where he studied device physics and state-space modeling of physical systems.*

Neil Kunst *is an Engineering Project Manager at Ridgetop Group. He earned his BSEE from the University of Arizona, where he was a member of the Tau Beta Pi National Honor Society. Neil received the Silver Bowl award and awards for outstanding achievement in Physics. He previously worked for Hamilton Test Systems, Intelligent Instrumentation, Inc., Mosaic Design Labs, Inc., Environmental Systems Products, Inc., Dataforth Corp., and SMSC. He also owned and operated his own firm, Palmtree Software, before joining Ridgetop. Mr. Kunst has more than 20 years of experience in product engineering, systems engineering, test engineering, logic design, software development, project management, and consulting.*

Sonia Vohnout *is a Systems Engineer at Ridgetop Group. She earned her MS in systems engineering from the University of Arizona. With a diverse background and experience, Sonia is well-suited to manage Ridgetop's commercialization efforts from its many government-funded projects. Sonia joined Ridgetop after successfully building an electronic subassembly business in Mexico, working as a Systems Engineer at IBM and handling overseas installations of software with Modular Mining Systems (now part of Komatsu). During her career, she has held executive management and senior technical positions. In addition, Sonia has co-founded several companies. Sonia is a board member of the Society for Machinery Failure Prevention Technology (MFPT), an interdisciplinary technical organization strongly oriented toward practical applications. Sonia also founded the Prognostic and Health Management (PHM) Professionals LinkedIn Group, a fast-growing group whose objectives are to: Discuss PHM related topics, network with others in the PHM community, and increase awareness of PHM. Areas of interest include electronic prognostics and system health management.*

Kai Goebel, Ph.D. *Kai Goebel, Ph.D. is a senior scientist at NASA Ames Research Center where he leads the Prognostics Center of Excellence. His research is concerned with developing methods for systems health management, in particular for remaining life estimation, uncertainty management, and decision making. He is the associate editor of the International Journal of Prognostics and Health Management. He has published more than 150 papers on Prognostics Health Management and has received 13 patents in this area.*