Infotech@Aerospace 2012
19 - 21 June 2012, Garden Grove, California

AIAA 2012-2421

A Model-based Avionic Prognostic Reasoner (MAPR)

Sonia Vohnout,[1] Byoung Uk Kim,[2] Neil Kunst,[3] Bill Gleeson,[4] and Robert Wagoner[5]
Ridgetop Group, Inc. Tucson, AZ USA 85741

and

Edward Balaban[6] and Kai Goebel[7]
NASA Ames Research Center, Moffett Field, CA 94035

The Model-based Avionic Prognostic Reasoner (MAPR) presented in this paper is an innovative solution for non-intrusively monitoring the state of health (SoH) and predicting the remaining useful life (RUL) of electronic and electromechanical assets by accessing and processing data obtained from a standard avionics data bus. To support Integrated Vehicle Health Monitoring (IVHM) initiatives, the solution being described here has been designed to be as non-intrusive as possible. An innovative, model-driven anomaly diagnostic and fault characterization system for electromechanical actuator (EMA) systems was developed to mitigate potentially catastrophic faults. EMA systems are used in a wide variety of aircraft applications to control critical components such as control surfaces, landing gear and thrust vector control. Failure in any one of these systems can compromise passenger safety, as well as mission success. A MIL-STD-1553 bus interface and monitor were designed to extract environmental (e.g., altitude, air speed, air density) and operational (i.e., response of system to a commanded change) data of a representative EMA system and to determine whether an anomaly is detected, and the corresponding severity. The MIL-STD-1553 bus was chosen as the test bed to develop this approach, due to its large installed base and availability of compatible development tools. Advanced and unique reasoning methodologies are applied to the extracted data sets to provide anomaly detection and fault classification on various fault modes and eventually yield SoH and RUL. In this paper we describe a data monitoring unit that will, in real time, identify, isolate, and characterize faults and establish their severity so that major performance problems can be alleviated. When built, this system will consist of a laptop with a Peripheral Component Interconnect (PCI) card slot that can accept multiple interfaces to the MAPR software package. The MAPR package will be designed to be adaptable for a large number of different platforms, for portability and for maximum input data type flexibility. This paper describes a ground-based prototype of the technology to show the efficacy of the method.

I. Introduction and Background

THE Next Generation Air Transportation System (NextGen) aims at making travel more efficient and speedy while maintaining or increasing safety. NASA's Aviation Safety Program (AvSP)[1] supports the NextGen by providing and supporting research on existing safety challenges such as increase in air traffic, introduction of new vehicle concepts, continued operation of legacy vehicles, increased reliance on automation, and increased operating complexity. Assurance of flight-critical systems, vehicle health assurance and prognostic algorithm design are among the top 10 technical challenges that AvSP supports.

[1] Director, Advanced Diagnostics & Prognostics, 3580 W. Ina Rd., Tucson, AZ, USA, 85741.
[2] Principal Research Engineer, Advanced Diagnostics & Prognostics, 3580 W. Ina Rd., Tucson AZ, USA, 85741.
[3] Principal Systems Engineer, Advanced Diagnostics & Prognostics, 3580 W. Ina Rd., Tucson AZ, USA, 85741.
[4] Senior Electrical Engineer, Advanced Diagnostics & Prognostics, 3580 W. Ina Rd., Tucson AZ, USA, 85741.
[5] Senior Software Engineer, Advanced Diagnostics & Prognostics, 3580 W. Ina Rd., Tucson, AZ, USA, 85741.
[6] Senior Scientist, Prognostics Center of Excellence, NASA Ames Research Center, Moffett Field, CA 94035.
[7] Research Scientist, Prognostics Center of Excellence, NASA Ames Research Center, Moffett Field, CA 94035.

The ability to accurately and precisely predict the remaining useful life (RUL) of critical aircraft components and subsystems enhances aircraft safety by enabling decision making and failure mitigation. In addition, it can improve operational efficiency by facilitating condition-based maintenance (CBM) and reducing unscheduled maintenance. In pursuit of the goals outlined by AvSP, this paper presents a Model-based Avionic Prognostic Reasoner (MAPR), an innovative solution for non-intrusively monitoring the state of health (SoH) and RUL of electronic and electromechanical assets by accessing existing data obtained from a standard avionics data bus.

As the aviation industry evolves toward next-generation fly-by-wire vehicles, hydraulic actuators are being replaced with their electromechanical actuator (EMA) counterparts. By eliminating fluid leakage problems while reducing weight and enhancing vehicle control, the feasibility of EMAs in avionic applications has been established. However, due to the inherent nature of electronic components and systems to degrade and eventually fail, improved diagnostic and prognostic methods are required to maintain the all-electric aircraft at safe levels. An advanced and unique reasoning methodology in anomaly diagnostic and fault characterization system for EMA systems was developed to mitigate potentially catastrophic faults. EMA systems are used in a wide variety of aircraft applications to control critical components such as control surfaces, landing gear and thrust vector control. Failure in any one of these systems can compromise passenger safety as well as mission success.

A MIL-STD-1553 bus interface and monitor were designed to extract environmental (e.g., altitude, air speed, air density) and operational (i.e., response of system to a commanded change) data of a representative EMA system and to determine whether an anomaly is detected, and the corresponding severity. The MIL-STD-1553 bus was chosen as the test bed to develop this approach due to its large installed base, and availability of compatible development tools. Advanced and unique mathematical algorithms are applied to the extracted data sets to yield SoH and RUL.

MAPR also has the ability to redefine the physical property model variables based on anomalous events that reveal systems degradation. With the fault severity cataloged in a database on the analysis platform, the detected fault can be isolated and tracked over time to support off-board analysis and calculation of the RUL of the device under test (DUT).

Accurate RUL predictions and fault diagnostics for critical systems of next-generation aircraft are vital for proactive health management. The ability to predict impending failure before it occurs supports mitigation of the fault and taking corrective actions. In this paper we describe a data monitoring unit that will, in real time, identify, isolate, and characterize faults and establish their severity so that major performance problems can be alleviated. When built, this system will consist of a laptop with a Peripheral Component Interconnect (PCI) card slot that can accept multiple interfaces to the MAPR software package. The MAPR package will be designed to be adaptable for a large number of different platforms, for portability, and for maximum input data type flexibility.

In previous research,[4,5] the authors 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. This paper presents the continuation of that research, where the authors leverage Ridgetop's EMA prognostics methodology to further develop the MAPR and describe a ground-based prototype of the technology to show the efficacy of this data analysis and fault classification method.

II. A Model-driven Anomaly Diagnostic and Fault Characterization System

The fundamental premise of our real-time MAPR is that an avionic system can be represented as a grey-box model with parameters that correspond to physical characteristics of the modeled system. For example, when applied to an EMA system, coefficients of resistance, inductance, and friction characterize the brushless DC (BLDC) motor that actuates the control surface. Since degradation of the system alters these parameters, system identification techniques such as Extended Kalman Filtering can be used to estimate system parameters and measure deviation from a healthy system. Presumably, parameter deviations can be analyzed and grouped to identify the damaged subsystem or component. Ideally, parameter drift can be trended to support prognostic measurements such as SoH and RUL.

During early research, a proof-of-concept version of the MAPR was introduced that processed relatively short motion profiles in pseudo-real-time using LabVIEW modeling techniques to successfully detect and classify anomalous EMA behavior.[2]

A. System Architecture

Figure 4 illustrates the system-level design of our integrated real-time MAPR development environment. The proposed test network architecture is ideally suited for both hardware-in-the-loop (HIL) and model-in-the-loop (MIL) testing, where the fault-enabled EMA testbed is replaced with a high-fidelity MATLAB/Simulink model

correlated with the testbed, dramatically accelerating technology readiness and commercial introduction. Potentially, the testbed and model could co-simulate identical MIL-STD-1553 position commands and fault modes simultaneously while the MAPR observes their individual responses.

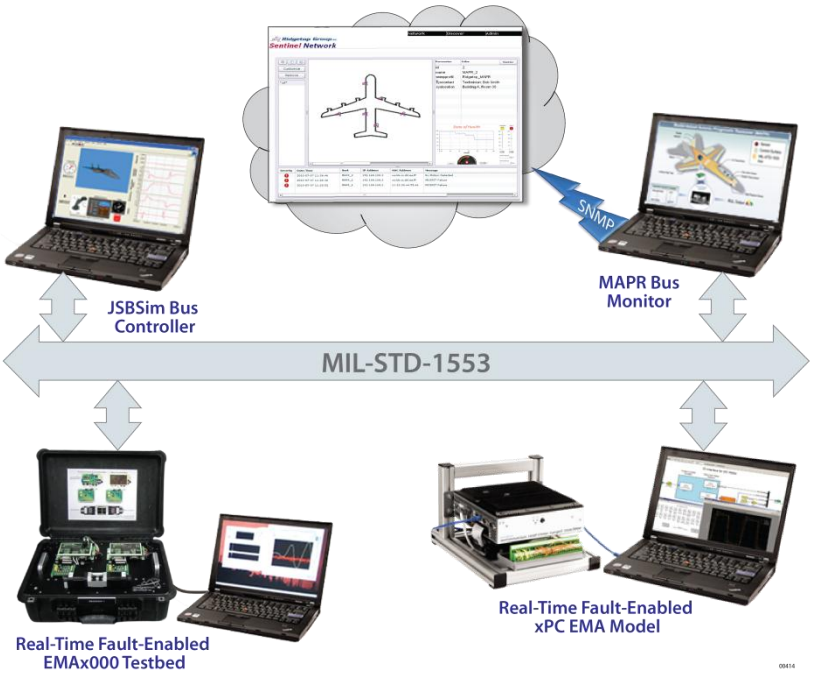


Figure 1: Integrated real-time MAPR development environment

The JSBSim tool was selected to produce realistic flight data for an advanced EMA testbed and modeling environment. JSBSim is an open-source flight dynamics model (FDM) that compiles and runs under many operating systems.[3] An application programming interface (API) was developed to transfer JSBSim control surface commands and simulated environmental conditions experienced by the aircraft to a local bus controller module. The bus controller then fed the aircraft flight data onto the physical MIL-STD-1553 data bus in real time, where it is simultaneously consumed by the EMA testbed and/or model and monitored by the MAPR sensor.

Once deployed, anomalous EMA events are detected by the MAPR and dispatched to a software analysis platform for further analysis and decision making. Ideally, the MAPR prototype would be capable of producing prognostics health data, such as actuator SoH and RUL measurements, in addition to detected anomaly events.

B. Real-time Testbed Architecture

Figure 2 illustrates the architecture of the real-time testbed being developed for MAPR anomaly detection and fault-to-failure progression (FFP) signature characterization. As shown in the graphic, a National Instruments single-board Rio (NI-sbRio) platform will provide a single MIL-STD-1553 remote terminal (RT) with one subaddress (SA) for control surface position command data and another for the associated position feedback sensor data. The NI-sbRio provides a common input/output (I/O) hardware interface for both HIL and MIL testing.

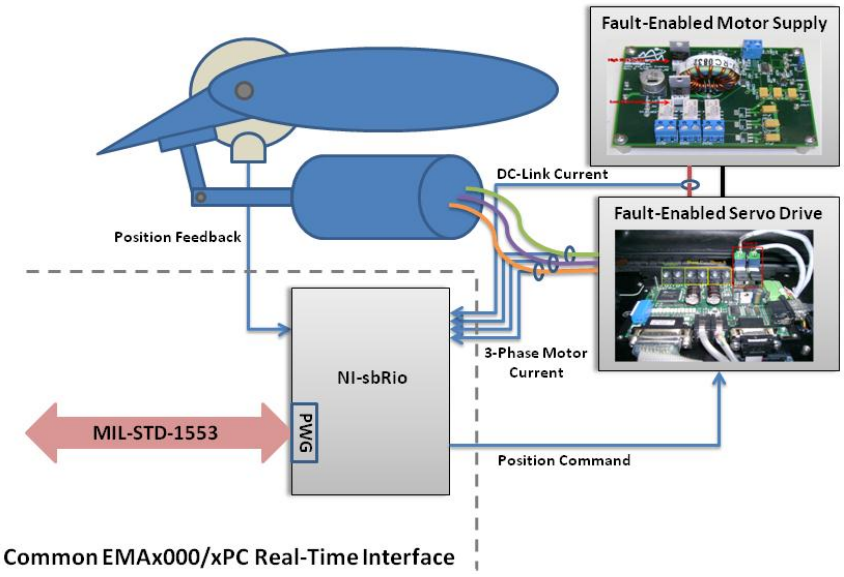


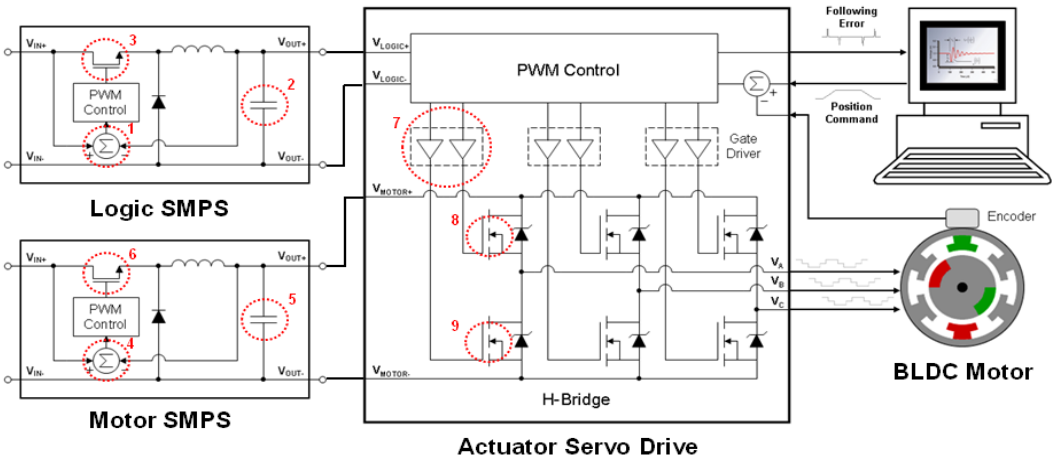
Figure 2: Real-time testbed architecture

Aside from retrofitting our state-of-the-art EMA2000 prognostic-enabled testbed with a scaled-down unmanned aerial vehicle (UAV) actuator and control surface, as depicted in Figure 2, a key element in the proposed architecture is the addition of external sensors to acquire motor DC-Link and phase current signals, in addition to the control surface position feedback, in real time. The sensor data acquisition, which is synchronized in time, is retrieved by the MIL-STD-1553 bus controller and is thus available to the MAPR for continuous, real-time analysis. Furthermore, MIL-STD-1553 RT subaddressing, along with sbRio I/O expansion, supports extensibility of the multivariate data. For example, thermocouples and accelerometers can easily be added to the testbed to measure metal-oxide semiconductor field-effect transistor (MOSFET) device temperature and motor vibration, respectively, for improved anomaly detection and FDI resolution.

The architectural diagram provided in Figure 3 identifies the power stages that comprise a typical servo drive (SD) subsystem, along with the electronic components most prone to aging and ultimately failure. A critical ingredient in the recipe of realizing an effective MAPR sensor capable of producing prognostic health data is a controlled aging methodology. Ridgetop plans to continue its productive collaboration with the NASA Ames Research Center (ARC) to improve component aging process control and apply the ARC's aging results to its proven artificial aging techniques.

Prior work conducted at the ARC revealed that applying thermal cycling overstress to MOSFET devices leads to thermomechanical stresses on the microelectronics due to mismatch of thermal expansion coefficients between different elements in the packaged device. Thermal runaway, latchup, and delayed or inhibited MOSFET switch turn-on were the prevalent failure modes observed during the experiment. By eliminating the traditional heat sink from the device and controlling device self-heating through various power cycle regimes, device output signals provide insight into the level of degradation during the aging process, as well as a precursor to failure for prognostic algorithm development.[7]

It may not be feasible to use natural aging data for the purposes of developing prognostic technologies. Although all electronic devices are prone to failure, it could take thousands of service hours, or many years, for the mission-critical applications of these devices to fail. Accelerated aging enables characterization of "run-to-failure" data in a manageable timeframe. The ultimate goal of our collaborative effort is to leverage the ARC accelerated MOSFET aging methodology to control the degradation of the MOSFET devices used in our MAPR testbed. Ideally, the MOSFET aging could be controlled to reasonable levels (e.g., degradation steps of 10-20% from 100% to 0%). The aged devices could be inserted into high- and/or low-side MOSFET socket(s) to characterize multivariate FFP signatures for singular, as well as compound, damage modes and produce effective diagnostic and prognostic MAPR algorithms.



Component Fault Candidates:

Logic SMPS	Motor SMPS	H-Bridge
1. Feedback amplifier	4. Feedback amplifier	7. Gate driver IC
2. Output filter capacitor	5. Output filter capacitor	8. High-side MOSFET
3. MOSFET switch	6. MOSFET switch	9. Low-side MOSFET

Figure 3: Fault-enabled servo drive power stage topology

C. Real-Time Model Architecture

Figure 4 illustrates the initial application of the real-time modeling methodology to the integrated MAPR development environment; we call it the Real-time Modeling Factory (RMF). Note the common I/O interface provided by the NI-sbRio for both HIL and MIL testing.

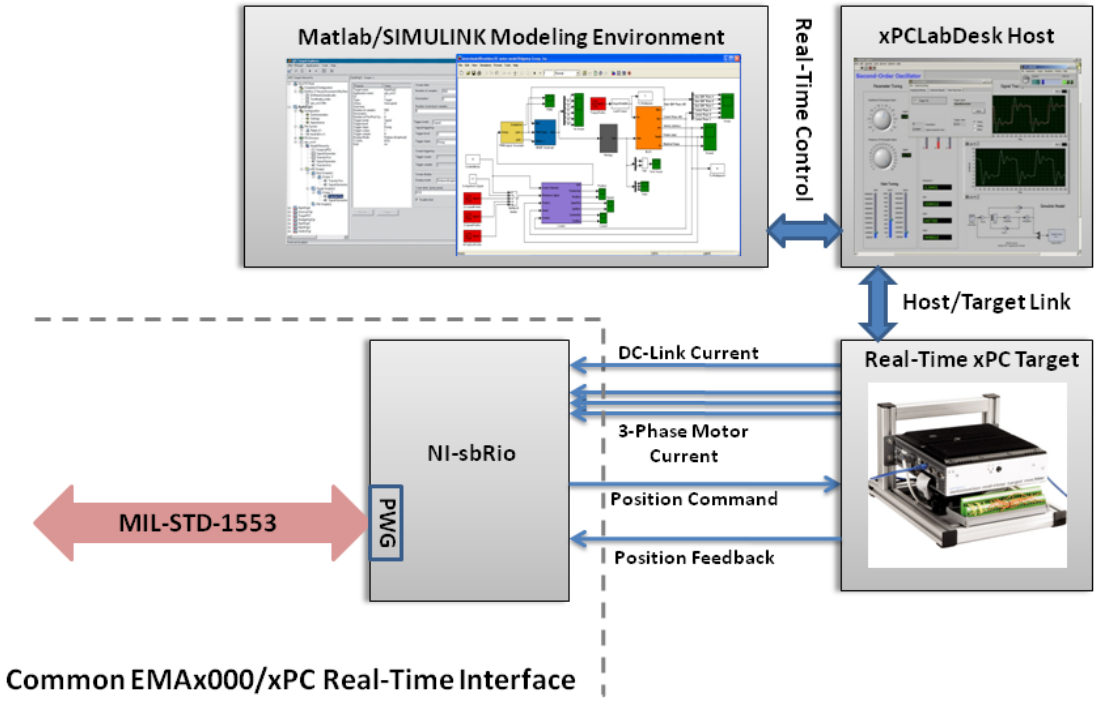


Figure 4: Real-time modeling architecture

The centerpiece of our real-time modeling architecture is the MathWorks xPC Target. Our real-time modeling flow begins in the MATLAB/Simulink environment, where the state-space EMA model is adapted for real-time execution on the xPC target. The Simulink Coder tool synthesizes the model into a C/C++ language representation, which is automatically compiled into object code for download from the host computer to the real-time target.

The xPC Target host software includes flexible APIs to programmatically control the real-time model on the target system. Parameters can be tuned, faults injected, and signals monitored in real time via the API. A third-party LabVIEW-based tool, xPCLabDesk, is being evaluated for RMF host control functionality. If suitable for the RMF, xPCLabDesk would couple the powerful simulation environment offered by MATLAB/Simulink with LabVIEW's rich graphical user interface (GUI) development environment to yield a compelling MAPR demonstration.

Along with multi-core processor and FPGA options, the xPC target offers a rich set of I/O blocks to enable HIL and MIL testing. The RMF will use xPC target I/O to interface the real-time model with the NI-sbRio, which provides a common HIL/MIL gateway to the MIL-STD-1553 data bus.

A critical architecture design process associated with targeting a complex model, like our high-fidelity EMA system, to a real-time xPC target is illustrated in Figure 5. That is, the model subsystems must be analyzed and partitioned based on simulation time step requirements, with the most demanding subsystems targeted at the FPGA module(s) and less demanding subsystems targeted at the processor core(s). Depending on model requirements, multi-core xPC target systems can be configured with multiple FPGA modules.

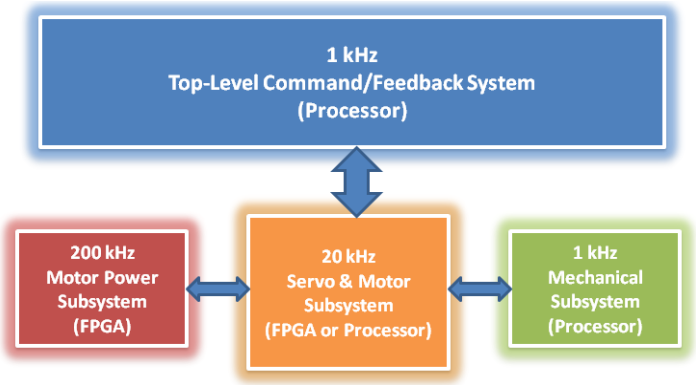


Figure 5: Multi-rate EMA model architecture

III. A Model-based Avionic Prognostic Reasoner (MAPR)

The MAPR runs on a testbed comprising a MIL-STD-1553 bus monitor and a MIL-STD-1553 bus controller (see Figure 1). 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.

The ultimate goal of the MAPR is to non-intrusively detect anomalous behavior and predict the RUL of electronic and electromechanical assets. This starts with a model that will be used extensively to create datasets, starting with the nominal response. Fault classification will identify MOSFET failure, capacitance degradation, etc. In this paper, we focus on the implementation of anomaly detection and fault classification requirements.

Finally, prognostics will be used for predicting an RUL value based on a state-of-health value derived from detected and classified faults.

A. Modeling and Anomaly Detection

The relationship between a classic dynamic motor model and the output data is illustrated in Figure 6. The classic motor model consists of a system of differential equations and model parameters including inductance, friction, inertia, torque constant, resistance, etc., among others, which are part of any generic motor system. The Simulink model provides the ability to manipulate model parameters (for simplicity referred to as a, b, c, d,…) and produce a response for position, velocity, current, and voltage.

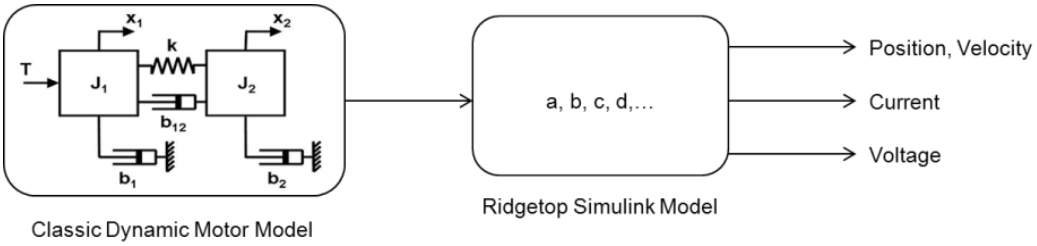


Figure 6: Model parameters and data output

The parameters in the classic dynamic motor model have been mapped to the Simulink model. Hence, changes to model parameters in the Simulink model correspond to changes in the classic motor model. The effects of changes to the model parameters in the Simulink model with respect to the response are being investigated, while keeping in mind that those model parameters are directly related to the classic dynamic motor model parameters.

B. Model Development

An important component of the MAPR development environment is real-time modeling of mission-critical avionic systems like the EMA. Toward that end the RMF concept was introduced to build a strong foundation for robust prognostics technology development. Figure 4 illustrated the initial application of the RMF methodology to the integrated MAPR development environment. As our original BLDC motor model evolved to support the RMF, two primary engineering objectives have emerged: integration with a mechanical subsystem for control surface actuation, and transformation of the power subsystem for real-time modeling.

Mechanical Model

Modeling of all the physical testbed components was successfully completed using the SolidWorks 3D CAD system, as shown in Figure 7. The rotary encoder wheel on the control surface hinge point of the Simulink model represents the high-resolution position sensor installed in the physical testbed .

The main body and control surface pieces are made from aluminum sheeting and assembled with machine screws, providing a lightweight and strong testbed. The control surface has one hardened steel shaft connected to the main body with one end bearing on each side for stability and low axial friction.

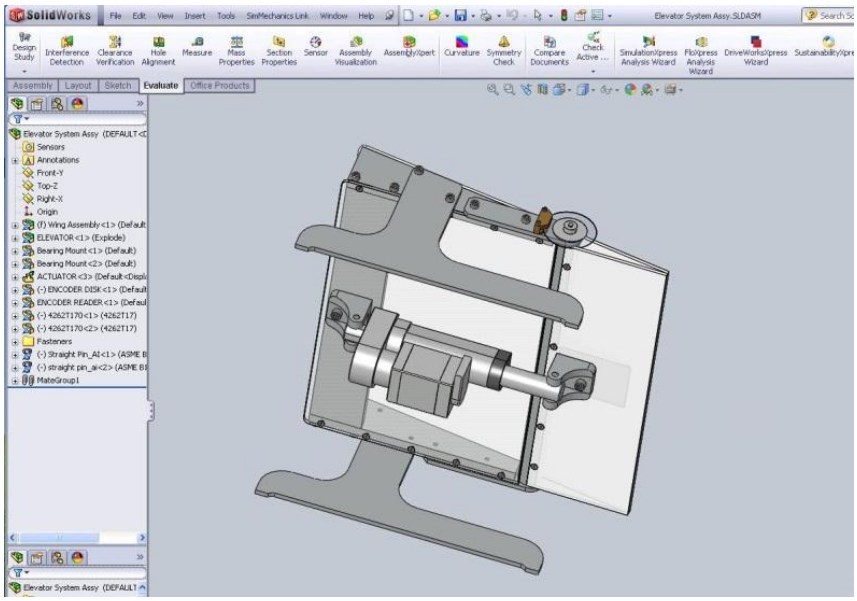


Figure 7: Completed 3D CAD assembly of testbed

Conversion from the SolidWorks 3D CAD assembly to the Simulink environment was successful with all mechanical properties included, as shown in Figure 8. Each component in the testbed can now be included with additional mechanical simulation blocks for use with the modeling environment in exercising the model and gathering test data.

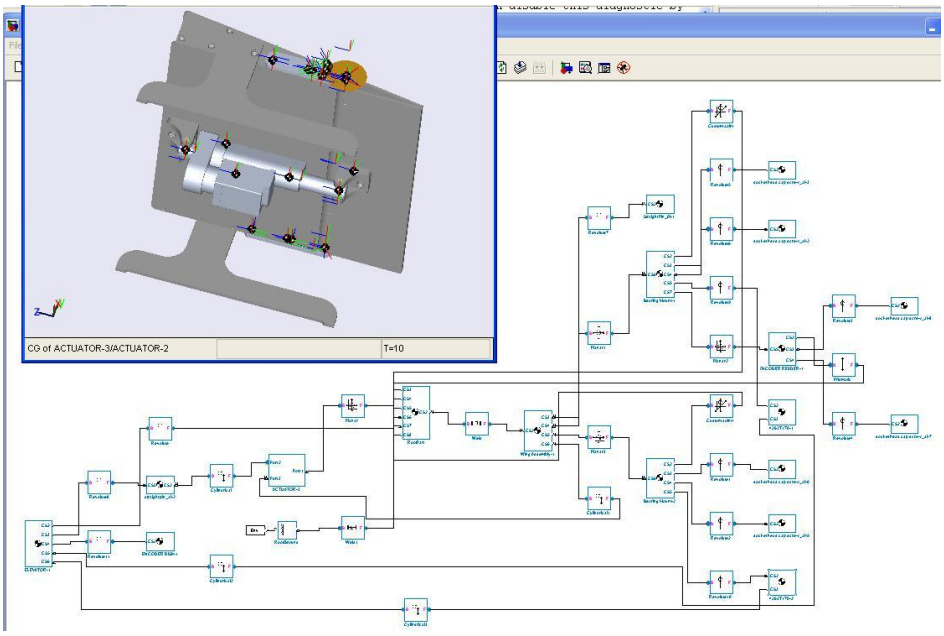


Figure 8: Converted 3D CAD to the Simulink environment

Power Subsystem Model

Model subsystems must be analyzed and partitioned based on simulation time step requirements, with the most demanding subsystems targeted at the FPGA module(s) and less demanding subsystems targeted at the processor core(s). Depending on model requirements, multi-core xPC target systems can be configured with multiple FPGA modules.

Preliminary analysis of our EMA model topology revealed the need for an FPGA for the BLDC motor power subsystem. Aside from converting all continuous and floating point blocks to discrete and fixed-point, respectively, the primary caveat associated with porting a Simulink model, like our BLDC motor, to a real-time FPGA target is that not all toolkits (e.g., SimPowerSystems) are supported. Fortunately, Simulink provides tools to help automate the conversion process. However, before using the conversion tools, we first had to replace the SimPowerSystems-based subsystem, circled in the center of our BLDC motor model presented in Figure 9, with an equivalent state-space representation.

Development of the BLDC motor model has increased the capability for fault injection to accommodate degradation to any parameter. In order to efficiently control and generate fault-seeded simulations, we created a simulation environment, shown in Figure 10, which programmatically collects input parameters, and records simulation outputs. The simulation environment was created as a data generation tool, demonstrating the ability of the BLDC model to operate under a variety of fault conditions. The model currently supports multiple fault modes: two terminal failure modes and five modes of variable degradation, outlined in Table 1.

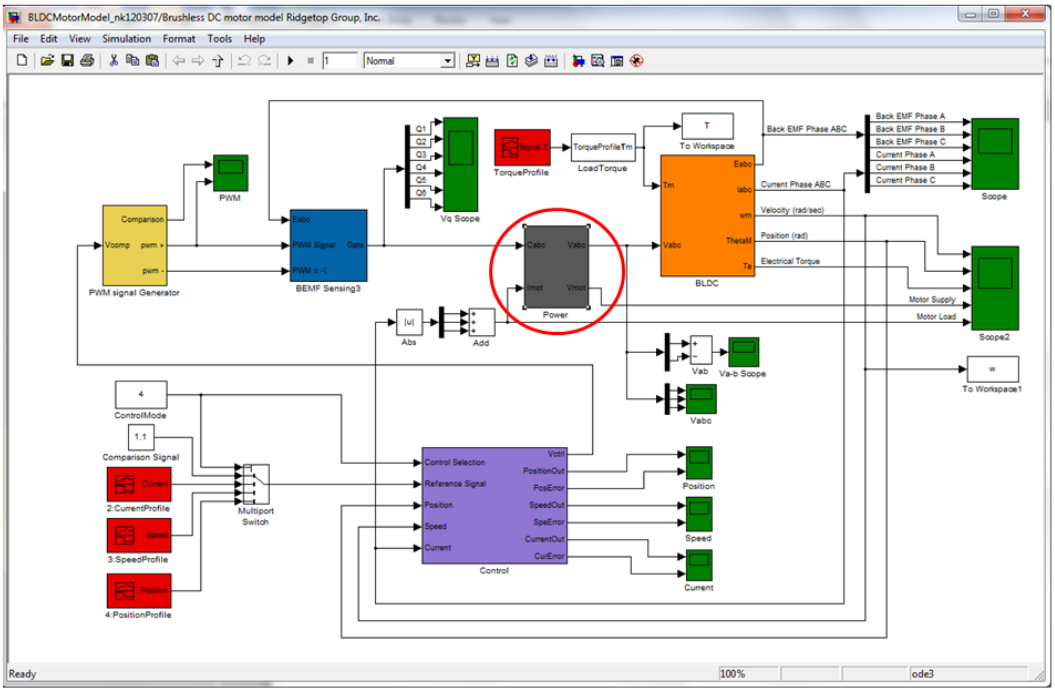


Figure 9: BLDC motor model

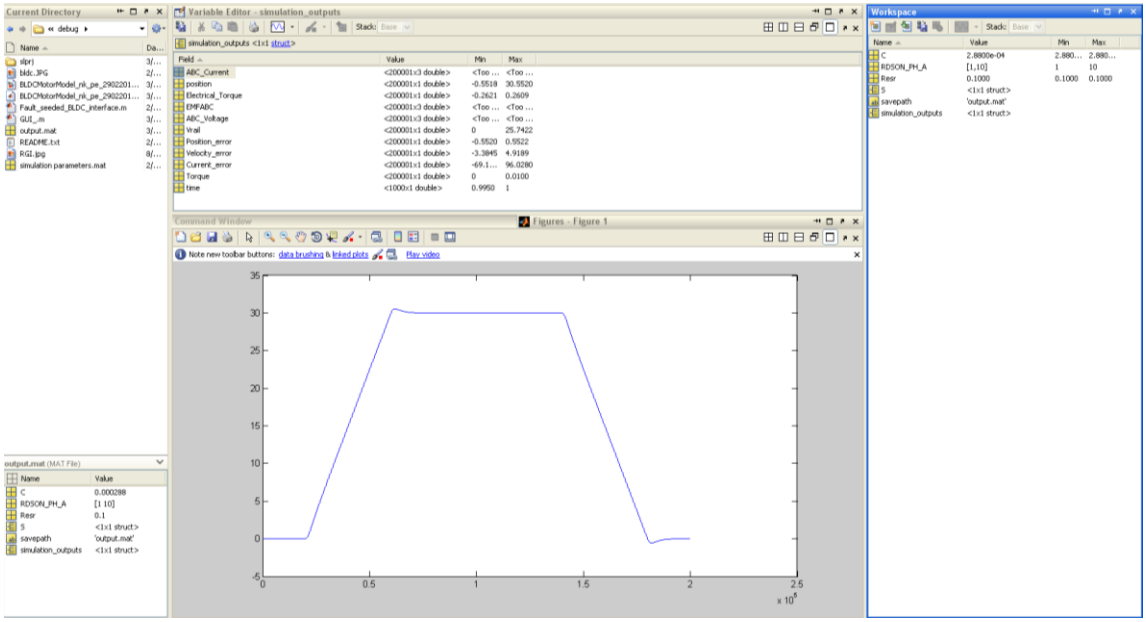


Figure 10: MATLAB simulation environment for Ridgetop's BLDC Motor Model

Table 1: Fault-Enabled Model Components

System Block	Subsystem Block	Model Parameter	Nominal Value	Min/Max	System Reaction
H-Bridge	SMPS	Output Filter Capacitor	288 µF	0-288	1) Inverter Input Instability
		Output Filter ESR	0.05 ohms	0.05-1.0	2) Ripple Noise
		MOSFET High-Side Ron	0.2 ohms	0.2-2.0	3) Output Instability
		MOSFET Low-Side Ron	0.2 ohms	0.2-2.0	4) Output Instability
	Inverter	Phase A Inverter High-Side On Resistance	0.036 ohms	0.036-0.36	5) Position Tracking
		Multi-Phase MOSFET Short	-	On/Off	6) Position Tracking
		Multi-Phase MOSFET Open	-	On/Off	7) Position Tracking

The following observations (as numbered in the System Reaction column in Table 1) were made from simulations performed on the model:

1. Reductions to the SMPS output filter capacitor dramatically increase the startup time of the SMPS 24 Vdc output. This also limits the amount of impulse current that is available to the inverter subsystem.
2. An increase in SMPS output filter equivalent series resistance (ESR) creates a larger ripple voltage on the 24 Vdc SMPS output.
3. Increases in High-Side MOSFET on resistance create an increased voltage, while reducing current output capability. These effects cause the SMPS output to become unstable.
4. Similarly for the Low-Side MOSFET, increases in on resistance cause signal instability.
5. Inverter High-Side MOSFET on resistance changes effect the inverter's ability to output current, which directly effects the motor's position.
6. A terminal failure, such as a short or open, to the Inverter's High or Low-side MOSFETs would remove the inverter's ability to move current through a phase of the motor winding. A MOSFET short would cause a significant current draw on the system, and cause position instability.
7. Similarly to 6, a MOSFET open would not allow current to move through the motor winding, and thus cause position instability.

Shown in Figure 11 is the simulation GUI. The Fault-Seeded BLDC Model Interface manipulates parameters in the simulation environment, and computes the system outputs of a Simulink model file.

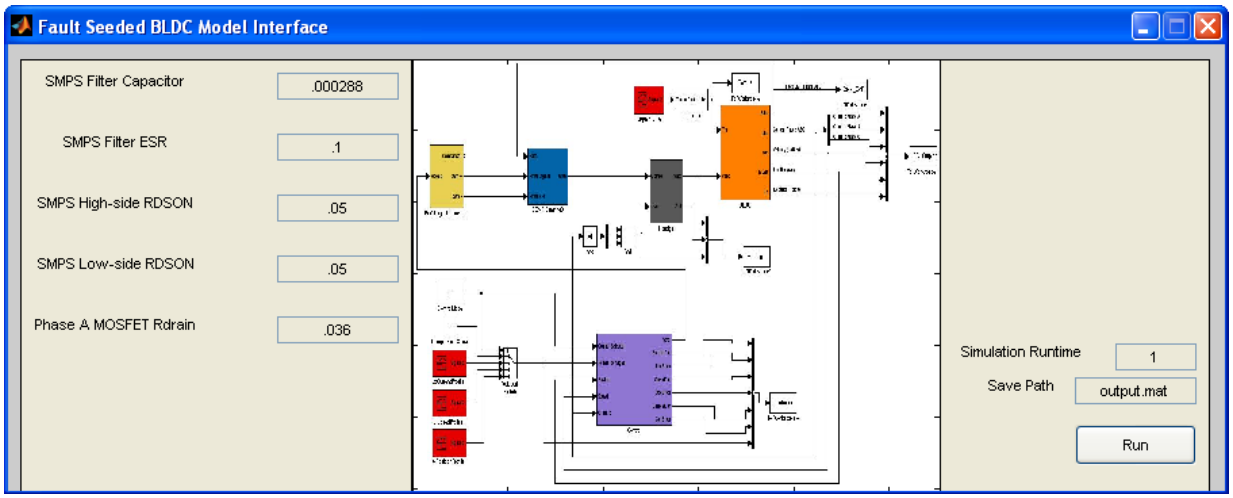


Figure 11: Fault-Seeded BLDC Model Interface

The methodology and tools developed for fault-seeded generation are a key component to successfully analyzing system-level data for prognostic analysis. With the Faulted-Seeded BLDC Model Interface GUI, generating fault-seeded simulations can be done efficiently and programmatically. By leveraging this method, data generation for analysis is expedited and a compound fault dictionary can be generated.

C. Data Analysis

Analysis began with seven data sets produced from the Simulink model. This included one nominal data set, three data sets for low, medium, and high capacitance faults, and three data sets for low, medium, and high on-resistance faults. We explored various approaches to analyzing the data including: auto-correlation technique, non-linear regression, computation of intermediate values, k-means clustering, and gradient descent. Below we describe in more detail the gradient descent approach taken in this research work.

Gradient Descent

During the initial evaluation of the data sets produced from the model with known seeded faults, it was hypothesized that an objective function could be used to identify the seeded fault. The objective function would need to incorporate both the nominal data set from the model produced under ideal conditions as well as the data set produced by a perturbation in one of the primary model parameters used for fault seeding. Numerically, this optimization would tend to a minima or maxima corresponding to the identification of the seeded fault. The initial hypothesis was further restricted to the point that a certain objective function measuring "a distance" between the nominal and fault-seeded data were minimized (i.e., tend to zero); this would identify the seeded fault qualitatively as low, medium, or high. With the objective function in place, various methods can be used to find the minima.

Gradient descent[6] is a method for finding a minimum for an n dimensional function f. The method is based on an initial guess, gradient, direction, step, and stopping condition. Recall that the gradient is defined as the partial derivative of the function f with respect to each independent variable in each of the unit vectors in n space. The magnitude of the gradient is the square root of the sum of the square of the gradient in each of the unit vectors in n space. This approach is computationally more efficient than, for example, Newton's method as it does not rely on the second derivative, which can increase the computational complexity of the algorithm from $O(n)$ to $O(n^2)$. There are various methods in which to determine the initial guess, step, and stopping condition.

Our approach to gradient descent has been developed and is described in this section. This algorithm also seeks to find the true minima and was founded on the concept walking down a hill to locate the valley. To find the true valley you start by looking around and identifying the direction of steepest descent (gradient). Taking one step (select step size) in that direction you compare your current position with the previous position (obtain result). If the new position is better, take a longer step in the same direction. If the new position is worse, take a half step back and compare your current position with the previous two positions. This description maps well to the algorithm shown in Figure 12.

```
Decide initial guess
While stopping condition is not met,
   Calculate the gradient
   Select a step size
   Obtain the result
   If result is better
      Compute a larger step in same direction
      Obtain result
   Else
      Compute a half step back
      Compare new result with previous two results
   End If
End While
```

Figure 12: Gradient Descent Analysis Methodology Procedure

One of the novelties of this approach is that the stopping condition is directly related to the size of the step. The step size will ultimately get smaller and smaller as you approach the true minima. This is in contrast to the traditional approach, which relies upon a stopping condition being the magnitude of the gradient less than some epsilon, $\|\nabla f\| < \varepsilon$. The results produced by this Multivariate Gradient Descent with Variable Stride Length are just as good if not better than the traditional approach.

The gradient descent approach has proven extremely useful as it has seeded the need to incorporate the data analysis while actively perturbing the model parameters. This methodology provides the basis for the first algorithm we have developed to detect and classify faults for output filter capacitance and on-resistance, described in this section.

The objective function to be minimized using our gradient descent was formulated in the following manner: Several intermediate values were computed. This includes the mean *DC*, frequency *f*, bandwidth *B*, and power *P*. These intermediate values are calculated for each of the phase currents and position producing 16 intermediate values for each of the 32 time windows. This is repeated for the nominal data set and all fault seeded data sets. Next, a difference is calculated between the first degradation level data set and the nominal data. This 32x16 matrix is the input to Principal Component Analysis (PCA), producing coefficients and variances for those coefficients identifying the components that contribute the most to the composition of the data. A total of 16 coefficients are produced from PCA and these are labeled a_i in the following equation set representing the computation of the new values NV_i. Each new value is the sum of the product of the PCA coefficients a_i with the corresponding *DC*, *f*, *B*, and *P* values.

$$NV_1 = a_1DC_{P1} + a_2f_{P1} + a_3B_{P1} + a_4P_{P1} + \ldots + a_{13}DC_{Ic1} + a_{14}f_{Ic1} + a_{15}B_{Ic1} + a_{16}P_{Ic1}$$

$$NV_2 = a_1DC_{P2} + a_2f_{P2} + a_3B_{P2} + a_4P_{P2} + \ldots + a_{13}DC_{Ic2} + a_{14}f_{Ic2} + a_{15}B_{Ic2} + a_{16}P_{Ic2}$$

.

.

.

$$NV_{32} = a_1DC_{P32} + a_2f_{P32} + a_3B_{P32} + a_4P_{P32} + \ldots + a_{13}DC_{Ic32} + a_{14}f_{Ic32} + a_{15}B_{Ic32} + a_{16}P_{Ic32}$$

For each level of degradation in the fault, the 32 new values are calculated. It is then possible to compare the values in each cell of the *NV* vector over increasing levels of degradation. This methodology spawned the idea of a good times matrix, which is described in more detail in the next section (D), to help produce a fault curve for each of the fault types.

D. Fault Classification Methodology and Results

Fault classification curves for output filter capacitance, on-resistance, and ESR have been created. These are shown in Figure 13, Figure 14, and Figure 15. The curves allow for the identification of the seeded fault as well as the value of the seeded fault to within an acceptable error. Each of the fault curves shows the seeded fault on the x axis and a computed distance from nominal on the y axis. The definition of the computed distance from nominal will

be explained in a later section. This method for fault classification will prove to be an excellent predictor for various fault modes injected into the model.

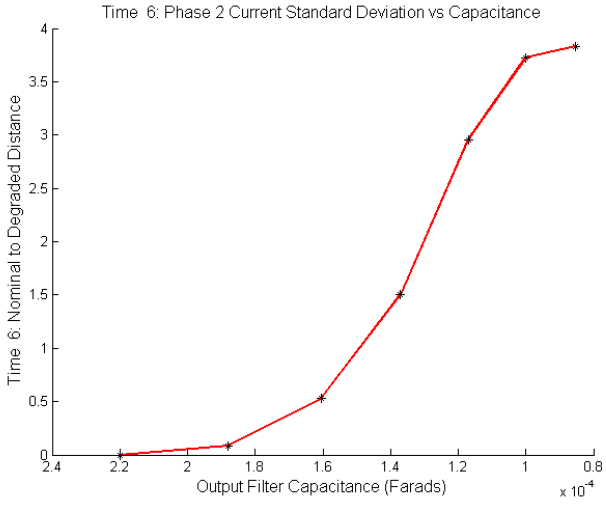


Figure 13: Output Filter Capacitance Fault Curve

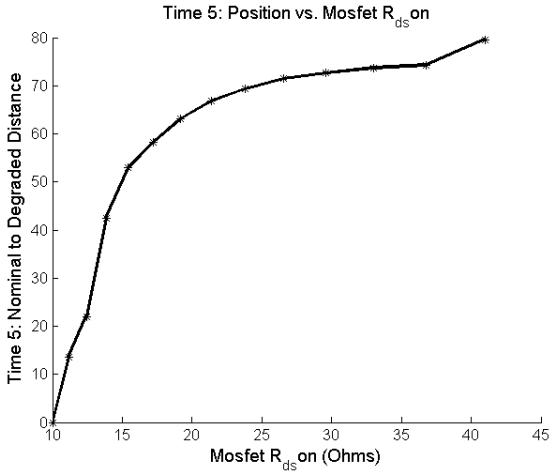


Figure 14: On-resistance Fault Curve

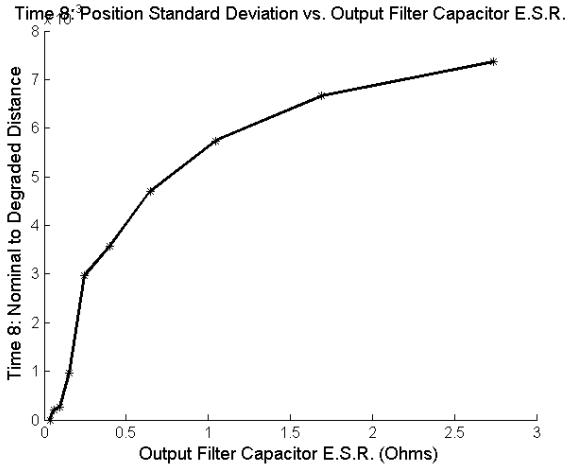


Figure 15: ESR Fault Curve

Each of the curves shows a relatively smooth shape. Some of the edges in the plots can be attributed to results from the model for fault conditions far away from nominal that are close to being outside the acceptable range for the model. However, it is possible to identify an equation of fit with a relatively small error term for the given fault curves. In addition, interpolation between the computed data points will also yield good results with small error, thus making it possible to guess with low error not only the fault, but the magnitude of the fault.

The curves are formulated in the following manner. Beginning with the raw data, the motion profile is divided into 32 time windows, 8 windows centered at each of corners of the response. Each time window is composed of 4096 sample values. This is depicted in Figure 16 where the black vertical lines are used to show the eight time windows around corner 1, blue vertical lines are used to show the eight time windows around corner 2, green vertical lines are used to show the eight time windows around corner 3, and red vertical lines are used to show the eight time windows around corner 4. There is no overlap between the time windows in the first and second corners as well as between the third and fourth corners.

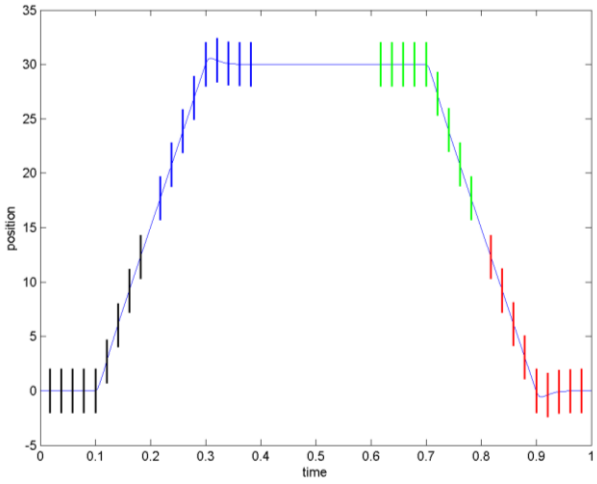


Figure 16: Nominal response with eight 4K windows (4 left, 4 right) around each highlighted corner

For each of the time windows, the following intermediate values are computed.

- DC or mean
- Standard deviation
- Cube
- Frequency

- Power
- Left Half Power
- Right Half Power
- Bandwidth

The mean is calculated in the standard way, $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$. Similarly, standard deviation is calculated as $s = \sqrt{\sigma^2} = \sqrt{\sum_{i=1}^{n}(x_i - \mu)^2}$. The cube is calculated as σ^3. Frequency is calculated using a Ridgetop-developed approach that is a hybrid of the classical discrete Fourier transform (DFT) calculation. Starting with DFT the peak is identified along with the computed point to the left and right of the peak. These two points that enclose the peak are used to identify a more accurate frequency and corresponding power. The left half power and right half power are computed from the known frequency and power. Bandwidth is the difference between the right half and left half power.

The calculation of the intermediate values for each of the time windows produces a 32x32 square matrix *M*. This matrix is produced for the nominal data and then produced for each seeded fault. Each of the matrices *M* is used to produce a single point in the fault curves shown in Figure 12, Figure 13, and Figure 14. The rows of *M* correspond to the time window that is row 1 is the first time window and row 32 is the last time window. The columns correspond to the phase currents and position. DC or mean make up the first three columns, where the DC or mean is calculated for each phase current. The next three columns correspond to the standard deviation for each phase current. This is repeated for the remaining intermediate values producing 24 columns. The final eight columns correspond to the calculation of the eight intermediate values for position.

Once the intermediate values have been calculated, the difference to the nominal matric M_{nom} is determined. This is the square root of the difference squared element by element of the fault-seeded matrix *M* and the nominal matrix M_{nom}; call it new matrix *N*. This difference is determined for each of the fault seeded simulations. These new matrices *N* are then ordered based on an increasing level of degradation.

Next, a "good times" matrix *G* is computed. The good times matrix is initialized to a 32 by 32 ones matrix. The assumption is that initially all intermediate values are good. Recall that the matrices *N* are ordered for increasing degradation. For each matrix *N* of increasing degradation, element $e_{i,j}$ is compared to the previous $e_{i,j}$ for a value that is greater than or equal to the previous value. If for levels of increasing degradation for a particular seeded fault, the intermediate value $e_{i,j}$ stays the same or increases, then element $e_{i,j}$ of the good times matrix *G* keeps the value of 1. If at any point the comparison yields a smaller value, then element $e_{i,j}$ of the good times matrix *G* goes to zero. This typically produces a sparse good times matrix *G* for each of the fault types.

The good times matrix is then used to produce a final values matrix *V*. The dimensions of this matrix are produced from the good times matrix *G*. The number of rows correspond to the number of ones in the good times matrix *G*. The number of columns corresponds to the number of fault-seeded matrices *N* for a particular fault type, plus an additional column used as a reference value.

For each 1 in the good times matrix $G_{i,j}$, a row in the final values matrix *V* is determined from the corresponding elements of the degraded matrices *N*. For example, assume that $G_{1,1}$ in the good times matrix is a 1, then the first row of the final value matrix contains the $N_{1,1}$ elements for all N matrices. The last value in the row is the reference value and is calculated as 1000 times the row *i* plus the column *j*. Again, the number of rows will correspond to the number of elements equal to 1 in the good times matrix *G*. Figure 17 is a plot of all the possible fault curves for on-resistance. Again, each curve is produced from each row in the final values matrix *V*. While many methods can be used to select the fault curve that will be used as a basis for fault classification, the chosen method is relatively simple, and it is to look for the maximum difference value between the fault-seeded data and the nominal at the highest degradation level.

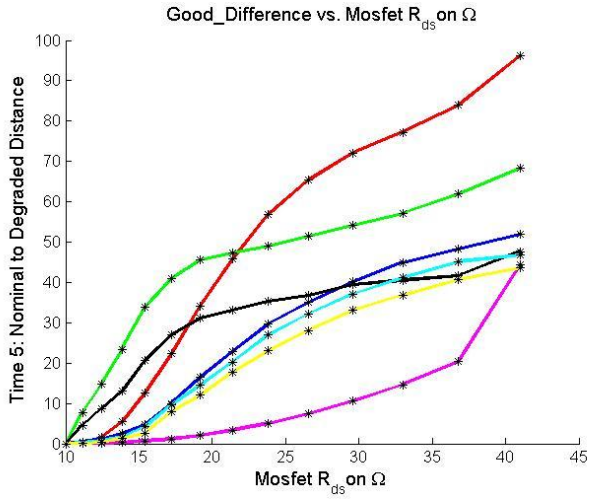


Figure 17: Plot of all possible fault curves for on-resistance fault type

To evaluate the effectiveness of the Ridgetop Classification Algorithm, 52 tests were completed with 4 signatures each. This produced a total of 208 testing runs. Each run contained a single anomaly, either power supply output filter capacitance, ESR of power supply output filter capacitance, increases to RDS_on of the MOSFET, or friction of the three-phase motor. The results from the testing are shown in Figure 18.

Of the 208 testing runs, 52 anomalies and 155 no anomalies were correctly identified. We achieved 99.5% accuracy and only one false alarm was found. The false alarm is shown as the point with the circle around it in Figure 18.

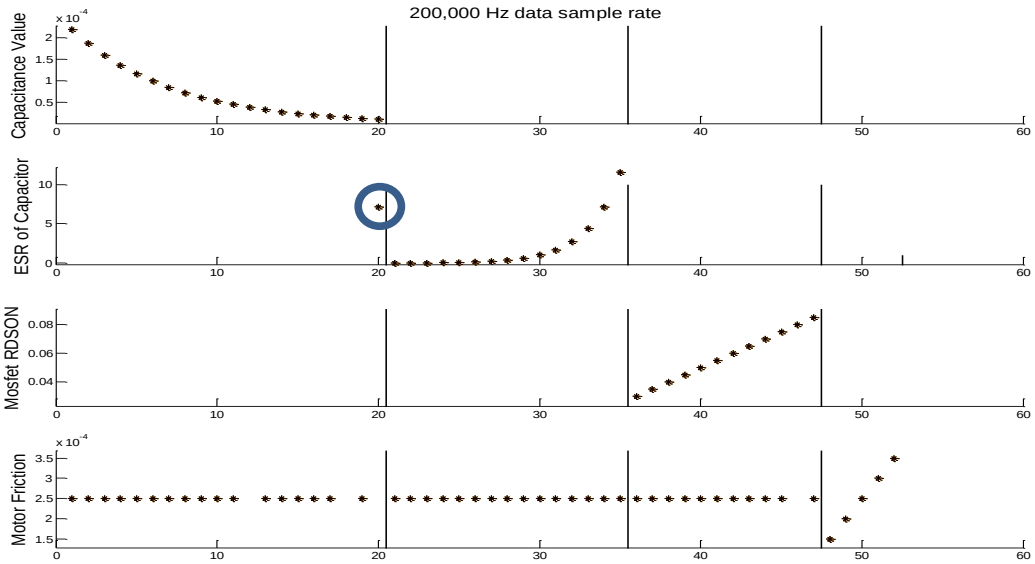


Figure 18: Results of fault classification

IV. Conclusions and Future Work

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.

Scalability, extensibility, and testability are key elements of the MAPR design. Our innovative approach effectively decouples the passive prognostic reasoner from the target flight control system (e.g., actuator) and supports multiple avionic data bus interfaces, such as MIL-STD-1553 and ARINC 429, simplifying adoption, validation, integration, and support. Potentially, a single MAPR could monitor multiple flight control systems, optimizing overall sensor costs.

Thus far, the anomaly detection and fault classification algorithms have been developed and validated solely with data provided from traditional modeling techniques and limited electromechanical fault modes. That said, preliminary results from traditional modeling have revealed an extremely high degree of effectiveness. As our algorithm methodology evolves with an expanding dictionary of actuator fault signatures, a comprehensive range of compound, as well as singular, damage modes will be supported. The FFP signatures will provide input into a prognostics analysis engine that will fortify the detection of anomalous actuator behavior with an accurate prediction of remaining service life.

Achieving our goal of advanced diagnostics and prognostics (ADP) of mission-critical avionic systems is contingent upon creating an integrated ADP development environment, complete with real-time command of control surface actuation and sensor feedback. Ridgetop's patent-pending MAPR abstracts the information provided by the low-level physical sensors to provide a high-level virtual sensor that reports anomalous system behavior and overall system health in real time to interested end-users, including operations and maintenance, air traffic control, and pilots.

A physical testbed that comprises a scaled version of the entire control surface actuation system commanded from a representative MIL-STD-1553 network is under construction at Ridgetop's Power Prognostics Laboratory. In parallel to physical testbed construction, our integrated EMA system model will be ported to a real-time modeling environment to eliminate the simulation latency of traditional modeling techniques. The real-time model will be correlated with the physical testbed to provide the computing kernel of our virtual MAPR sensor. Ultimately, the MAPR sensor can be deployed on-board as a node on the aircraft data bus to provide real-time alerts of anomalous actuator behavior or off-board as part of a ground-based IVHM system that can monitor the playback of historical flight data to provide vehicle- or fleet-wide analysis of system health.

References

[1]NASA's Aviation Safety Program. www.aeronautics.nasa.gov/programs_avsafe.htm.

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

[3]Main project website on SourceForge. http://jsbsim.sourceforge.net/

[4]N. Kunst, S. Vohnout, C. Lynn, and B.U. Kim. A Prognostics Approach for Electronic Damage Propagation and Analysis in Electromechanical Actuator Systems. MFPT, 2010.

[5]B.U. Kim, C. Lynn, N. Kunst, and S. Vohnout. Fault Classification with Gauss-Newton Optimization and Real-Time Simulation. IEEE Aerospace, 2011.

[6]Jan A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing, 2005.

[7]J. Celaya, A. Saxena, S. Saha, and K. Goebel. Prognostics of Power MOSFETs Under Thermal Stress Accelerated Aging Using Data-Driven and Model-Based Methodologies. PHM Society, 2011.