JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 12, No. 3, March 2015

Prognostic Health-Management System Development for Electromechanical Actuators

Edward Balaban*
NASA Ames Research Center, Moffett Field, California 94035
Abhinav Saxena†
SGT, Inc., NASA Ames Research Center, Moffett Field, California 94035
Sriram Narasimhan‡
University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, California 94035
Indranil Roychoudhury§
SGT, Inc., NASA Ames Research Center, Moffett Field, California 94035
Michael Koopmans¶
Tesla Motors, Palo Alto, California 94304
Carl Ott**
U.S. Army Aviation Development Directorate, NASA Ames Research Center, Moffett Field, California 94035
and
Kai Goebel††
NASA Ames Research Center, Moffett Field, California 94035

DOI: 10.2514/1.I010171

Electromechanical actuators have been gaining increased acceptance as safety-critical actuation devices in the next generation of aircraft and spacecraft. The aerospace manufacturers are not ready, however, to completely embrace electromechanical actuators for all applications due to apprehension with regard to some of the more critical fault modes. This work aims to help address these concerns by developing and testing a prognostic health-management system that diagnoses electromechanical actuator faults and employs prognostic algorithms to track fault progression and predict the actuator's remaining useful life. The diagnostic algorithm is implemented using a combined model-based and data-driven reasoner. The prognostic algorithm, implemented using Gaussian process regression, estimates the remaining life of the faulted component. The paper also covers the selection of fault modes for coverage and methods developed for fault injection. Validation experiments were conducted in both laboratory and flight conditions using a flyable electromechanical actuator test stand. The stand allows test actuators to be subjected to realistic environmental and operating conditions while providing the capability to safely inject and monitor propagation of various fault modes. The paper covers both diagnostic and prognostic run-to-failure experiments, conducted in laboratory and flight conditions for several types of faults. The experiments demonstrated robust fault diagnosis on the selected set of component and sensor faults and high-accuracy predictions of failure time in prognostic scenarios.

I. Introduction

IN RECENT years, the designers of new aerospace vehicles have been moving away from the more traditional hydraulic actuators and toward fly-by-wire technologies. Several types of actuation mechanisms are being used in such fly-by-wire designs, with electromechanical actuators (EMA) being one of them. As actuators are typically among the most safety-critical components of an aerospace system, an undetected or unmanaged actuator failure can lead to serious consequences [1]. Although actuators have been studied extensively from a functional point of view, studies from a health-management point of view have been rather limited, largely due to the unavailability of fault data (either from fielded applications or laboratory experiments with seeded faults). In setting up actuator fault-injection experiments, however, a researcher is faced with a number of challenges:

1) The first challenge is realism. To gain confidence in electromechanical actuator health-management algorithms, phenomena inherent to the flight environment (such as vibration, acoustic noise, electromagnetic interference, and acceleration loads) need to be reproduced in the test setup. Fault seeding techniques need to be chosen with realism in mind as well, especially for faults that would normally occur as a function of long-lasting operational conditions (fatigue-induced faults, for example).

Presented as Paper 2011-1518 at the Infotech@Aerospace 2011, Saint Louis, MO, 29–31 March 2011; received 17 August 2013; revision received 3 June 2014; accepted for publication 10 November 2014; published online 24 March 2015.

*Computer Engineer, Intelligent Systems Division, Discovery and System Health Area, MS 269-4; edward.balaban@nasa.gov.
†Research Scientist, Intelligent Systems Division, Discovery and System Health Area, MS 269-4; abhinav.saxena@nasa.gov.
‡Project Scientist, Intelligent Systems Division, Discovery and System Health Area, MS 269-4; sriram.narasimhan@nasa.gov.
§Research Scientist, Intelligent Systems Division, Discovery and System Health Area, MS 269-4; indranil.roychoudhury@nasa.gov.
¶Test Engineer, 3500 Deer Creek Road; michael.t.koopmans@gmail.com.
**Associate Director for Design & Test/Experimental Test Pilot, Aviation Development Directorate–Aeroflightdynamics, AMRDEC, MS 219; carl.r.ott.mil@mail.mil.
††Area Lead, Intelligent Systems Division, Discovery and System Health Area, MS 269-4; kai.goebel@nasa.gov.

2) The second challenge is cost. As is often the case with experimental data collection, there is the need to strike a balance between realism and cost. Quality actuators, especially those developed for use in flight applications, can cost tens of thousands of dollars each. Yet, the number of test articles needs to be large enough to discover salient data features and ensure statistical significance of the conclusions. Creating a realistic environment via a flight-test program or an appropriate laboratory setup could become expensive as well. Cost constraints may also impose limits on the realism of fault injection.

3) The third challenge is time. Development of a detectable fault and its transition to failure can take years for most EMA faults modes under common operating conditions. Such a length of time is rarely available for any development program. Thus, accelerated aging methods need to be developed and their correlation to naturally occurring faults needs to be established.

4) The fourth challenge is instrumentation. When instrumenting test articles, limitations on sensor suits and data acquisition (DAQ) systems imposed on fieldable solutions have to be considered. For example, although having high-precision piston position sensors on an actuator would help with detecting and characterizing fault models such as backlash, it would not be practical for most fielded applications.

In this paper, we describe our efforts in addressing the aforementioned challenges. To increase the realism of fault-injection experiments (while keeping experimental costs low), a hardware testbed was constructed for use in both laboratory and in-flight experiments. The most common and important EMA fault modes were examined and fault-injection methods for several of them (of both abrupt and continuous degradation types) were developed. The fault-injection methods used allowed execution of repeatable experiments with variable motion profiles, load profiles, and fault magnitude values. A prototype diagnostic system, combining techniques from both data-driven and model-based areas, was developed to monitor for and diagnose several fault types common to EMA. A prototype prognostic system for remaining useful life estimation was developed as well, based on the principles of Gaussian Process Regression. The prognostic system was demonstrated on a cascading fault scenario, where a fault in one component (jam in the ball-screw mechanism) leads to a fault in another component (heat build up and eventual windings insulation failure in the electric motor). The paper is organized as follows. Section II, outlining prior and concurrent efforts, is based on a literature review conducted in the early stages of the project. Section III gives a high-level overview of the research approach. Sections IV and V present the details of the diagnostic and prognostic systems, respectively. The testbed used in the experiments is described in Sec. VI. Laboratory diagnostic experiments are covered in Sec. VII, while prognostic experiments are the subject of Sec. VIII. A description of the flight experiments conducted to date follows in Sec. IX. The paper concludes with a summary of the work accomplished and suggestions for future research directions.

II. Related Work

This section reviews related efforts in EMA health management. The section is organized in three parts, with the first one covering EMA performance assessment methods, the second focusing more on fault-injection experiments, and the third illustrating diagnostic/prognostic algorithm development work. The partition is not strict, as some of the efforts performed work within more than one area. In those cases, the primary objective is emphasized.

A. EMA Performance Assessment

An EMA performance testing program was conducted at NASA Dryden Research Center in 2000 [2]. The Electrically Powered Actuation Design (EPAD) program was a joint effort between the Air Force, Navy, and NASA and had the goal of testing EMAs in-flight, driving a primary control surface of a high-performance aircraft (F-18 jet). The test program was conducted over the full flight envelope of the aircraft and EMA performance was compared to the standard hydraulic actuator in the opposite wing. The program demonstrated that EMA performance, according to the key metrics, was close to that of the hydraulic actuator it replaced.

B. Fault-Injection Experiments

A team at Lockheed Martin developed accelerated aging methods for EMA and conducted run-to-failure experiments to identify precursors to failure [3]. In these experiments, the actuator lubricant was seeded with an abrasive contaminant in order to induce accelerated wear. The goal was to achieve test article failure in 24 h or less, providing acceptable test times while ensuring that the precursor signatures are representative of normal wearout (the term "precursors" in this study refers to the set of features that are particularly informative in predicting onset of a failure). Failure precursors were identified, although no one failure precursor was consistently reliable in all of the experiments. Other challenges were also identified (such as wearout of nonseeded components affecting test results or cascading effects of support-bearing wear on the electric motor).

Spall, ball-screw return-channel jam, and backlash injection experiments [4] were conducted at Moog Corporation (with some of the data used in the initial phase of the work described in this paper). An EMA test stand was constructed that used Moog 883-023 actuators as test articles. Data were also collected on nominal test articles, both on new test articles and those that underwent repeated disassembly and reassembly in order to confirm repeatability of results after actuator refurbishing.

Astronics AES developed a test stand for nondestructive EMA testing [5]. Actuator overload, binding, jam, and broken linkage faults were simulated by using an external "dead" weight to simulate drag or overload by bolting the actuator linkage to the support table to simulate jam and by disconnecting the linkage to simulate broken linkage. No extra sensors were added to the actuator, with all the fault-detection features extracted from the current measurements in the power distribution system. Although the fault-injection methods were relatively simple, interesting findings were presented on fault detection with features derived from current signatures only.

C. Diagnostic and Prognostic Algorithm Development

Impact Technologies developed an EMA test stand, where a variety of faults were injected and data were collected in a laboratory environment. Fault modes were chosen based on an extensive failure modes, effects, and criticality analysis (FMECA) study and included motor winding shorts, rotor shaft eccentricity, and encoder malfunctions [6]. A fault-detection system was developed and validated using the collected data.

A joint team from the Georgia Institute of Technology, the Northrop Grumman Corporation, and the U.S. Air Force Research Laboratory pursued development of a prognostics and health management (PHM) system for EMAs using Kalman and particle filters [7]. Another activity conducted under the program was the development of reconfigurable control algorithms [8]. The algorithms took advantage of online real-time remaining useful life (RUL) estimates of a failing component and reconfigured the available control authority by trading off system performance with control activity.

A linear quadratic optimal control was developed at The Boeing Company to extend the remaining life of the motor bearing and brushless direct current motor windings in an EMA [9]. A model was developed to correlate motor operating temperature and the winding lifetime, and various control strategies were compared using a simulation environment. In another collaborative effort between The Boeing Company, the U.S. Air Force Research Laboratory, and Smiths Aerospace in 2006, an aircraft electrical power systems prognostics and health-management system was developed to address faults in electric actuation, fuel pumps and valves, and arc fault protection [10]. Actuator jam faults due to gear, bearing, and

ball-screw malfunction were studied. Further, winding shorts were simulated at four discrete levels by switching connections in single motor phase windings. Ball bearing degradation was tracked as an increase in freeplay in the EMA motion. The conclusion from the study, however, was that the indicators found had limited sensitivity to be able to distinguish between various EMA fault modes and more investigation was required to determine the appropriate sensor suite and signal processing methods.

Work conducted by Ridgetop Technologies focused primarily on modeling, simulating, and developing health-management methods for power electronics components of EMA systems [11]. The most likely fault modes in power components, such as metal oxide semiconductor field-effect transistors (MOSFET) and insulated-gate bipolar transistors (IGBT), were selected and modeled. Ridgetop Technologies also created a series of hardware testbeds that allowed fault injection by switching current flow from a healthy component to a degraded one.

III. Approach

The overall goal of this work was to create, to the extent possible, an example process of developing an EMA health-management system, starting with fault modes analysis and ending with prognostic assessment of the actuator health. The step subsequent to prognostic health assessment, involving decision-making based on system health information, is the subject of ongoing research, with some of the preliminary results described in [12,13].

A. Failure Modes, Effects, and Criticality Analysis Studies

As the first step, representative failure modes, effects, and criticality analysis (FMECA) documents for several EMA types were obtained in order to compile a prioritized list of fault modes. This list contains faults that are deemed to be the most likely and consequential. The faults on the list were classified into four approximate categories: sensor, motor, mechanical/structural, and electrical/electronic. Examples of sensor faults include bias, drift, scaling, and dropout. Motors are prone to developing winding shorts, rotor shaft eccentricities, and magnet delaminations. Mechanical/structural faults examined include return-channel jam, spalling, and backlash. Electrical and electronic components can suffer dielectric breakdowns, insulation deterioration, or intermittent contact, among other problems. For a more detailed discussion on the subject, please refer to [14].

B. Modeling

Given the prioritized list of EMA faults, the modeling effort was initiated. A physics-based modeling approach was chosen to provide an accurate representation of fault progression. The physics-based approach was complemented by data-driven machine learning techniques, where appropriate. The modeling efforts ranged from creation of high-level models of EMA operation [14,15] to the more detailed studies of bearing balls motion inside ball-screw raceways and return channels, as well as the effects of lubricants on the dynamics of that motion. Although not all of these models ended up being selected for use with our health-management algorithms, all of them provided crucial insights into the specifics of EMA operation under various conditions.

C. Diagnostic and Prognostic System Development

Development of health-management (diagnostic and prognostic) algorithms followed. First, a neural network-based diagnostic system was developed, described in detail in [16]. The system was capable of diagnosing such fault modes as ball-screw return-channel jams, spalls, and sensor faults of varying magnitudes. For the next version of the diagnostic system, a hybrid (combined model-based/data-driven) approach was adopted [17]. This was complemented with a prognostic system, which covers a subset of faults using Gaussian process regression (GPR) approach (described in Sec. V).

D. Experimental Validation

To verify performance of the algorithms in a relevant environment, several EMA testbeds were used. One of them, previously mentioned in Sec. II, was constructed by Moog Corporation. Data collected on this test stand were used in verifying the performance of the neural network diagnostic classifier. Although laboratory test stands make it possible to simulate some of the desired flight conditions on the ground, testing equipment and algorithms in the presence of vibrations, noise, acceleration loads, electromagnetic interference (EMI), and temperature changes inherent to flight is invaluable. These considerations led to the development of the flyable electromechanical actuator (FLEA) testbed, which is the testbed used in this work.

The FLEA was designed as a self-contained lightweight test fixture containing three actuators: one nominal, one injected with faults, and the third providing dynamic load (Fig. 1). The load is switched in flight from the healthy to the faulty test actuator, thus allowing collection of both baseline and offnominal sensor data under the same conditions. The testbed flies on a host aircraft, mimicking one of its control surface actuators. Position and load information for the active test actuator is derived from the corresponding real-time values of the target actuator on the host aircraft. Data collected on the stand are routed to the prognostic health-management system that monitors actuators for faults and, if a fault is detected, predicts the effects on actuator performance and its remaining useful life. Although the main purpose of the FLEA is to perform data

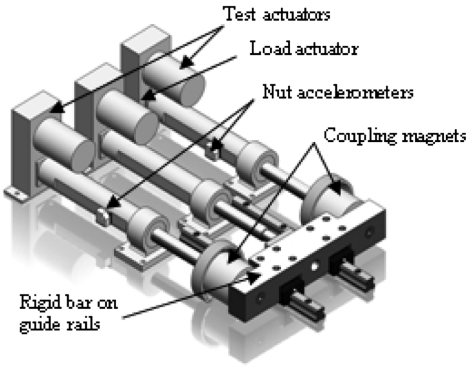


Fig. 1 FLEA actuator coupling system.

collection in real-time flight conditions, experiments using the testbed can (and have been) performed in laboratory conditions. Laboratory experiments are useful for verifying and validating performance using well-defined scenarios. Flight and laboratory experiments involving the FLEA are described in detail in Secs. VII–IX. Careful consideration was given to the design of the experiments, especially to developing fault-injection techniques that are as realistic as possible, both in terms of their magnitude and, for gradually growing faults, in terms of time progression from fault to failure. These techniques are covered in Sec. VI.

In addition to demonstrating diagnostic and prognostic capabilities, an important aspect of the flight experiments was to collect suitable data for further development of PHM algorithms between the flights. These data are required to have certain properties to make them suitable for the purpose. For instance, prognostic algorithms require availability of run-to-failure data. It can be challenging to obtain continuous run-to-failure data during a limited number of relatively short-duration flights, as some of the faults types take longer than that to grow and reach failure levels. Furthermore, slow fault growth characteristics lead to a very large amount of data (multiple sensor readings combined with high sampling frequencies) with a relatively small incremental information gain from the prognostics point of view. To alleviate this problem, several approaches have been tested. With the one currently used on the testbed, snapshots of high-sampling-rate data are taken during a flight at predetermined intervals. This way, under the assumption of slow fault growth, the loss of important trend information is contained to acceptable levels. It must be noted that this approach is not used for the diagnostic system, which requires continuous data at a lower sampling rate (and, possibly, a smaller overall set of measurements). Once a fault is detected and confirmed, a trigger is used to activate the prognostic algorithm and a higher-sampling-frequency data collection for relevant channels is initiated.

IV. Diagnostic System

Diagnostic approaches can be broadly divided into two types: model-based [18] and data-driven [19]. Model-based schemes rely on a system model built from a priori knowledge about the system, whereas data-driven schemes do not require such models but instead require large sets of exemplar failure data, which are often not available. Some of the sensors available in the EMA, such as current and voltage sensor outputs, can be modeled using physics-based differential equations and can be used for model-based diagnosis of faults in EMA. The modeling of accelerometers, however, is outside the scope of this work, and hence a data-driven feature-based diagnosis approach is better suited to leverage accelerometer information for faults disambiguation. Our approach to EMA diagnostics synergistically combines model-based and data-driven diagnosis techniques in order to improve upon either approach implemented individually. First, we discuss the model-based and feature-driven methods implemented, and then we describe how these methods are combined into the diagnosis engine for the FLEA.

A. Model-Based Fault Detection

We use the TRANSCEND diagnosis approach [20] for the model-based diagnoser (Fig. 2). In TRANSCEND, the observer module takes into account the control inputs sent to the system and the measurement readings obtained by the sensors in order to track the system dynamics and estimate the unobservable system states. The observer can, for example, be implemented as a particle filter [20], which generates estimates using a system of first-order differential state-space equations.

For fault detection, TRANSCEND uses a statistical Z-test [21] on each sensor output to determine whether the deviation of a sensor output from its nominal expected value is statistically significant, taking into account sensor noise and other uncertainties. Once a significant deviation is detected in any one measurement, the symbol generation module is initiated, and every measurement residual ($r(t) = y(t) - \hat{y}(t)$, where $y(t)$ is an observed measurement and $\hat{y}(t)$ is the measurement estimate calculated based on the state estimates obtained from the observer) is converted into qualitative $+$, $-$, and 0 symbols, based on whether or not the observed measurement is above, below, or at its expected nominal value, respectively.

The detection of a fault triggers the qualitative fault isolation module to determine the fault hypotheses, i.e., all possible system parameters and their direction of change that could explain the observed measurement deviation from nominal. Initially, based on the first observed measurement deviation, a set of fault candidates is generated [20]. Then, for each fault candidate, we systematically determine a fault signature for each measurement [20]. A fault signature is an ordered set of two 0, $+$, or $-$ symbols (one for magnitude and the other for slope), which represent how each measurement residual would deviate if that fault was the only fault in the system (in this work, we restrict our discussion to single faults). After the fault signatures are generated, qualitative diagnosis involves comparing an observed deviation with the expected fault signatures of each fault for that measurement and removing any fault hypothesis that does not explain the observed deviation. If the qualitative deviations in measurements alone cannot help discriminate some faults, we term these faults as aggregate faults and consider them to belong to an ambiguity group. It is desired to reduce the fault hypotheses set to a singleton set. However, this is not always possible in a real engineering system. For example, in the FLEA, spall and jam faults have similar effects on all non-accelerometer sensors. The qualitative model-based approach alone is therefore not sufficient, and a data-driven approach is employed to further disambiguate faults and obtain better diagnosis results.

B. Data-Driven Fault Isolation

The qualitative diagnoser described previously uses low-sampling-frequency data to make comparisons between the actual system output and the model output. This allows a fast and computationally inexpensive first-stage screening in real time. However, once a fault ambiguity set appears, a more detailed analysis is required that uses higher-resolution higher-sampling-frequency data, such as accelerometer signals. Figure 3 shows the architecture of a generic feature-driven diagnosis approach, which works by reasoning over a set of features or condition indicators from sensor data and can distinguish (although perhaps not individually) between fault classes of interest.

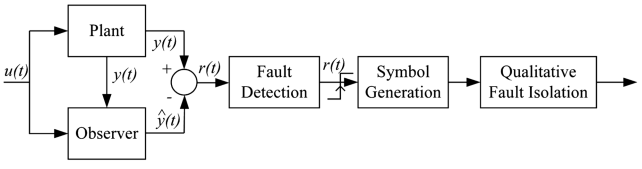


Fig. 2 TRANSCEND diagnosis architecture.

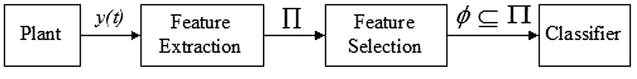


Fig. 3 Feature-driven diagnosis architecture.

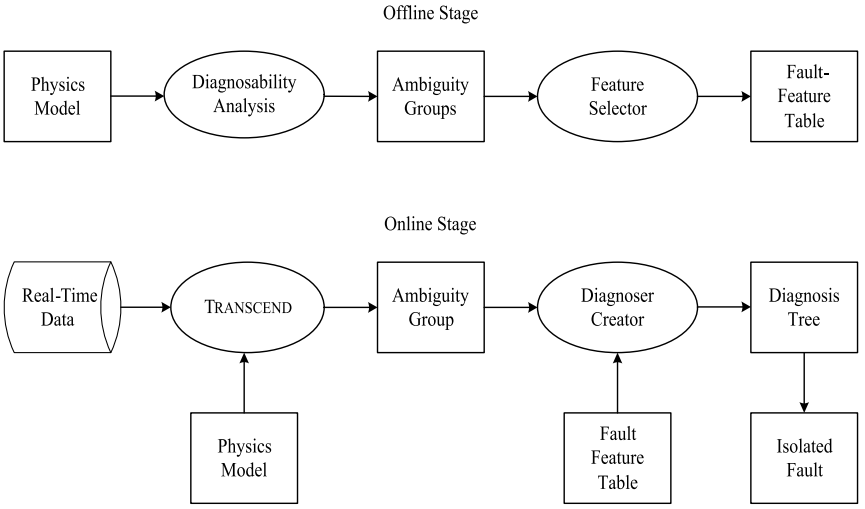


Fig. 4 Hybrid diagnosis architecture.

In most cases, a suitable set of features must be identified and computed from the training data during the training phase. Then, at runtime, the same features need to be computed from the incoming condition monitoring data and supplied to the classifier, which then detects faults and returns the most probable root causes. This data-driven mode can be relatively computationally expensive, depending on the feature extraction methods and the size of the system. However, this complexity can be reduced if the scope of fault identification is narrowed to only ambiguous situations. To accomplish this, a hybrid approach is adopted by combining qualitative and data-driven diagnostics.

C. Hybrid Diagnosis Approach

The model-based approach works well when one is able to derive analytical models for all the faults under consideration, as well as their effects. However, it may not be possible to do so for all the faults (due to resource constraints, for example). The feature-driven approach, on the other hand, usually requires a large amount of training data under varying experimental conditions. Additionally, when the classifier has to consider all faults and experimental conditions, the size and complexity of the classifier may become intractable. We present a hybrid method that combines these two approaches, as illustrated in Fig. 4. This approach consists of an offline stage and an online stage.

1. *Offline Stage*

In the offline stage, a bond graph (BG) model of the system is derived [20] and a qualitative diagnosability analysis is performed on it. The BG model can be used to generate qualitative signatures for all the faults represented by changes in the BG parameters. By comparing the qualitative signatures, we can identify the ambiguity groups (groups of faults that have similar fault signatures). These groups represent faults that need to be distinguished using the feature-driven approach.

For each ambiguity group, a set of features is extracted (using domain knowledge or by experimentation) that is identified to contain diagnostic information to disambiguate the maximum number of faults in that group. This results in a fault-feature table that indicates how specific features are influenced by faults. Table 1 shows an example of a fault-feature table. Each row of this table corresponds to a fault, and each column corresponds to a feature. The cell entry corresponding to Fault_n and Feature_k contains a qualitative signature that represents how the fault affects the feature. The qualitative signature belongs to a vocabulary of signatures for that particular feature. For example, Feature_1 can have a vocabulary of *high*, *nominal*, *zero*, and *highlow*, whereas Feature_2 can have a vocabulary of "1" and "0," denoting whether a particular fault affects this feature or not, respectively.

2. *Online Stage*

The online stage is carried out in two phases. In the first phase, the TRANSCEND approach is used to observe the system, detect, and qualitatively isolate fault ambiguity groups. In the second phase, an isolated ambiguity group triggers the selection of rows from the fault-feature table. These rows correspond to the faults in the ambiguity group. The goal of this phase is to determine the fewest sets of features to compute in order to fully resolve the fault ambiguity. This problem can be proven to be nondeterministic polynomial-time hard (NP-hard); hence, no polynomial-time solution is known. Using the most discriminative features first is an example of a fast, greedy search strategy that, however, does not guarantee optimality. To obtain an optimal solution, we use a heuristic-driven best-first search, which is detailed in the next paragraph. This approach can be easily replaced with any other optimal/exponential-time or suboptimal/polynomial-time method.

The selected fault-feature subtable can then be converted to a diagnoser tree using the measurement selection procedure detailed in [22]. The nodes of this diagnosis tree are groups of faults, and the edges represent specific values for features. The root of the tree is the initial ambiguity group. At each level, some of the tree features are selected that partition the ambiguity group in the most balanced fashion. Figure 5 shows an example diagnosis tree. This can be formally specified as partitioning with the least difference between the largest and smallest partitions. Once the best feature has been identified, the ambiguity group is partitioned into subgroups corresponding to the possible values for the selected feature (one subgroup for each possible feature value). For each subgroup of a size greater than one, the next best feature (which creates the most balanced

Table 1 Fault-feature table

Fault	Feature_1	Feature_2	...	Feature_k
Fault_1	Vocabulary $\{\text{Feature}_1\}$	Vocabulary $\{\text{Feature}_2\}$	...	Vocabulary $\{\text{Feature}_k\}$
...	...	...	...	...
Fault_n	Vocabulary $\{\text{Feature}_1\}$	Vocabulary $\{\text{Feature}_2\}$	...	Vocabulary $\{\text{Feature}_k\}$

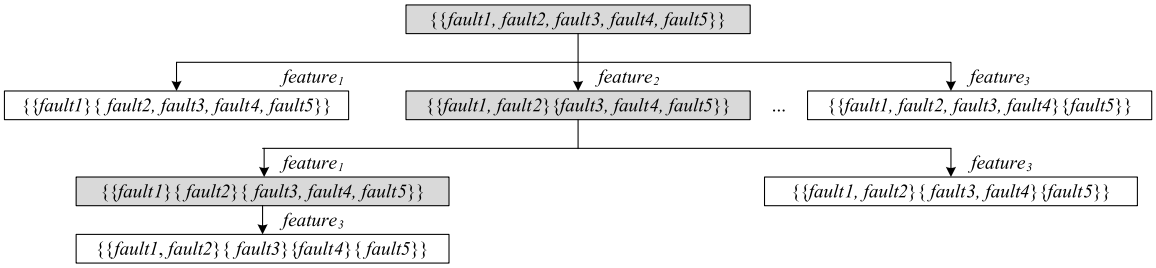


Fig. 5 Example diagnosis tree.

partitions) is selected. This process continues until only subgroups of a size of one are left or there are no more features left to select. If there are subgroups of a size greater than one, then this indicates indistinguishable faults. Algorithm 1 presents a method for identifying the ordered list of features that will discriminate within the ambiguity fault set in the quickest and the most balanced manner.

Once the diagnoser tree has been identified, fault isolation is performed by walking down the tree from the root node. First, the feature associated with the edges from the root node is computed. Depending on the value of the feature, the corresponding partition of the ambiguity group becomes the current belief. Again, we compute the feature associated with that node to further reduce the size of the current ambiguity group until we reach a leaf node of the tree. At this point, fault isolation is completed and the final ambiguity group (or aggregate faults) can be reported as the diagnosis.

The construction of the diagnoser tree can be carried out offline for a set of identified ambiguity groups, in order to speed up computation at runtime. However, if there is sufficient computational power, the *lazy* approach might be more suitable. In the *lazy* approach, the entire diagnoser tree is not constructed. Rather, only the first best feature is identified. This feature is then computed, and the ambiguity group reduced. The procedure is repeated (computing only the first best feature) for the current ambiguity group. The process concludes when a single fault has been isolated or all features have been computed.

V. Prognostic System

The next step, after a fault is detected and isolated, is to track its rate of growth and estimate the remaining useful life. There are numerous ways in which a prediction algorithm can be implemented; in this phase of the work, a GPR-based prediction algorithm was chosen [23]. This choice was primarily guided by two factors:

1) GPR is a data-driven technique, and therefore does not require users to specify explicit fault growth models (which are often not available during early research phases).

2) GPR can analytically process the uncertainty associated with the data.

GPR learns the trends based on evidence from data in a probabilistic framework and provides variance bounds for the predicted trajectory, which can be interpreted as the subjective confidence in the predictions. Such estimates of uncertainties could play an important role for decision making based on prognostic predictions. The choice of GPR for this effort originated from continuation of previous work, where GPR was successfully employed as a prediction algorithm. Using an intelligent sampling scheme, the computational cost was brought down to within the constraints of real-time prognostics [24], which made it an easy choice for this research effort. Performance of GPR has been compared with competing data-driven methods in other publications, where GPR was shown to perform reasonably well in similar settings [25,26]. A brief description of GPR is provided next.

A Gaussian process (GP) is a collection of random variables, any finite number of which has a joint Gaussian distribution. A real GP $f(x)$ is fully specified by its mean function $m(x)$ and covariance function $k(x, x')$, defined as

$$m(x) = \mathbb{E}[f(x)] \tag{1}$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{2}$$

$$f(x) \sim \mathrm{GP}(m(x), k(x, x')) \tag{3}$$

where x and x' are input vectors, $\mathbb{E}$ is expectation, and GP is the Gaussian process function.

Algorithm 1 Selection of order of features

1: **Inputs:** Fault-feature table, $FaultFeatTable$; set of n ambiguous faults, $FaultSet$; set of features $FeatureSet$
2: **Initialize:** Fault partition, $FaultPartition = \{FaultSet\}$, such that it contains one partition with all ambiguous faults in it; ordered first-in first-out queue of features, $OrderedFeatureList = \emptyset$
3: **while** $|FeatureSet| = \emptyset$ or $|FaultPartition| = n$, **do**
4: $BestPartition = FaultPartition$
5: **for** each $feature$ in $FeatureSet$, **do**
6: Identify new fault partition $FaultPartitionNew$ such that $BestPartition$ is partitioned into using $feature$ and looking up $FaultFeatTable$
7: **if** $FaultPartition$ is more balanced than $BestPartition$, then
8: $BestPartition = FaultPartition$
9: **end if**
10: Push $feature$ into $OrderedFeatureList$
11: Remove $feature$ from $FeatureSet$
12: **end for**
13: **end while**
14: **return** $OrderedFeatureList$, $FaultSet$

The index set $\mathcal{X} \in \mathbb{R}$ is the set of possible inputs, which need not necessarily be a time vector (although it is in our case). Given prior information about the GP and a set of training points $\{(x_i, f_i)|i = 1, \ldots, n\}$, the posterior distribution over functions is derived by imposing a restriction on a prior joint distribution to contain only those functions that agree with the observed data points. These functions can be assumed to be noisy, since in real-world situations we only have access to noisy observations rather than exact function values. More precisely, $y_i = f(x) + \epsilon$, where ϵ is independent, identically distributed, additive, Gaussian (normal) noise, with zero mean and σ_n^2 variance $[N(0, \sigma_n^2)]$. Once we have a posterior distribution, it can be used to assess predictive values for the test data points [23].

Domain knowledge available from the process is encoded by the covariance function $k(X, X')$ that defines the relationship between data points in a time series. Ideally, GPR can tap into prior knowledge about the form of covariance function, which may be inferred from the application domain. Covariance functions consist of various hyperparameters that define the temporal characteristics of fault growth. Setting the right values of such hyperparameters is key in learning the desired functions. A covariance function must be specified a priori, but corresponding hyperparameters can be learned from the training data using a gradient-based optimizer, such as maximizing the marginal likelihood of the observed data with respect to hyperparameters [27].

Having briefly described the fundamentals of GPR, we now outline the methodology used in the FLEA prognostic experiments. The experiments serve a dual role. First, they facilitate demonstration of an integrated diagnostic and prognostic system in real-time conditions. Second, data collected in flight operating conditions are valuable in developing algorithms robust against environmental noise. However, these two disparate goals impose some non-overlapping requirements on the data-collection methodology. It is expected that the prognostic system is triggered once the diagnoser indicates the onset of a fault mode. Data collected henceforth are processed in real-time by computing relevant features. These features are then fed to the GPR algorithm for a certain period of time, in order to estimate GPR model parameters. Generally, longer training periods result in better chances of the algorithm learning the true fault growth characteristics. However, to strike a balance between the length of the training period and the risk of missing out on a sufficient prediction horizon, a limit on the training period must be set. After training is complete, the algorithm starts predicting trajectories of fault growth. End of Life (EOL) is subsequently determined by where these trajectories intersect the predetermined failure level threshold. Estimated EOL values can then be specified in relative terms by computing the RUL values, if needed.

As more data are collected, the GPR model and, subsequently, the predictions are updated as well. It must be noted that GPR may run into scalability issues if a long data history is used, as its computational complexity is $\mathcal{O}(n^3)$, where n here is the total number of input observations. This problem is addressed by sampling the training points from the accumulated observation data. A fixed upper bound was used for the size of the covariance matrix and, subsequently, the number of training data points k. These k data points are sampled uniformly from all the history data available until the current time index. Using these points, the algorithm carries out a maximum-likelihood optimization [28] to determine the best-fitting hyperparameters for the chosen covariance function. Since this process involves numerical methods, the outcome of the optimization often depends on the initialization of the hyperparameters. This further contributes to the uncertainty in the predicted outcomes. In the implementation presented here, this uncertainty is characterized and handled in two ways. First, at each prediction time instant t_p, a large ($q \approx 50$) number of sets of k data points each is drawn from the observed data. Then, q different GP models are trained based on these q sample point sets. Each GP model yields a mean and variance function resulting in a family of distributions to represent the variability in the RUL. The uncertainty in the distribution parameters (mean and variance) is integrated to calculate a single probability density function for the RUL through a predictive distribution computation method, described in [29]. By averaging the results from the q distinct GPRs, it is expected that variability resulting from random sampling is largely eliminated, whereas the computational costs are kept low. In other words, computational complexity is now reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(qk^3)$, where $q \ll n, k \ll n$ as time passes by. A similar concept was explored in [24], where temporal sampling was used to parallelize the algorithm and reduce its computational complexity.

VI. Testbed

The key ideas in designing and building the FLEA test stand were for it to be lightweight, self-contained, and capable of supporting three different actuators: one nominal, one injected with faults, and the third providing dynamic load (Fig. 1). The load is switched in flight from the healthy to the faulty test actuator, thus providing the fault-injection capability for the test stand without having to alter an actuator in flight. The stand is connected to the aircraft data bus, and the motion profiles for the test actuators (as well as the load applied to them) are derived dynamically from the corresponding values for one of the aircraft's control actuators. Being a largely self-contained unit, the FLEA only requires interfaces to the aircraft data bus and power. An engineering model of the FLEA is shown in Fig. 6.

A. Mechanical

The test stand frame is constructed from T-slotted extruded aluminum segments connected with brackets and fasteners. The 1-cm-thick center plate is attached to the frame and used for mounting the actuators and other components of the stand. The rigidity of the central plate was an important design consideration; therefore, an analysis was performed that ascertained only negligible bending under the expected loads. Before a flight, the sides of the chassis (except for the top) are covered with 3-mm-thick aluminum plates. These plates serve a dual purpose: as an additional safety measure and to provide EMI protection.

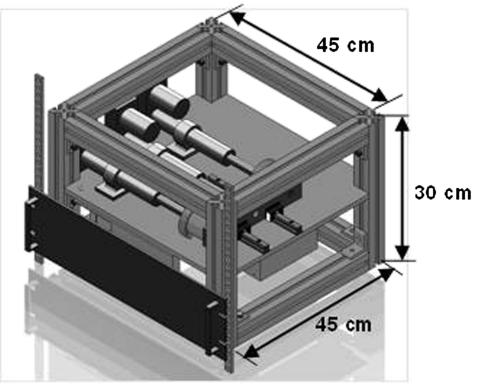


Fig. 6 FLEA engineering model.

Fig. 7 UltraMotion Bug actuator.

The onboard computer, running the operating system, data acquisition, control, and health-management software, is based on an off-the-shelf Pentium 4 3.2 GHz motherboard. Storage is provided by two solid-state drives: one for the operating and control software, and another dedicated to data storage. The actuators used in the FLEA at present are UltraMotion Bug actuators (see Fig. 7). They are controlled via a Galil 4030 multiaxis controller. Coupling of test actuators to the load actuator is accomplished via an electromagnetic system (presented earlier in Fig. 1). Only one test actuator at a time is normally coupled to the load actuator.

B. Fault Injection

Faults are injected into the test articles in the following manner:

1) A jam fault is injected via a mechanism mounted on the return channel of the ball screw that allows us to stop circulation of the bearing balls through the circuit.

2) A spall fault is injected by introducing cuts of various geometries via a precise electrostatic discharge process. The initial size and subsequent growth of these cuts are confirmed by using an optical inspection and measurement system.

3) Motor failure is introduced by redirecting current from the affected motor into a sink load.

4) Sensor faults (bias, drift, scaling, and complete failure) are software-injected into the measurements collected by the data acquisition system.

C. Sensor Suite and Data Acquisition System

The data acquisition system consists of two National Instruments 6259 cards and the Galil motor controller, and it supports the FLEA sensor suite (Table 2). Low-speed data are acquired at a 1 kHz sampling rate and the high-speed data are acquired at a 20 kHz sampling rate. The accelerometers are connected through custom-fabricated conditioner boards that supply them with excitation voltage and remove the dc portion of the return signal. The current sensors are built in the Galil motor controller. Additionally, real-time data for calculating position and load profiles are acquired when the FLEA is flown aboard an aircraft. Aircraft-specific software interface modules implement the needed bus communication protocols and equations for translating aerodynamic data into commands for the load actuator. The acquired data are displayed for visual inspection, saved to data files for analysis, and sent to the reasoning software. The overall control architecture is illustrated in Fig. 8.

D. Reasoning Software

Two reasoning modules are deployed on the FLEA: 1) diagnostic, and 2) prognostic (the respective algorithms were described earlier in Secs. IV and V). The diagnoser is responsible for monitoring the sensor data and determining whether any faults are present in the system. After the diagnoser has determined that a fault has occurred in the system, the prognoser is responsible for determining how the fault will progress and how long the system's remaining useful life is. Both modules are created in MATLAB [30] and interface with the LabVIEW [31] code of the control and data acquisition system through queues and other data synchronization structures.

VII. Laboratory Diagnostic Experiments

The FLEA testbed allows measurement of the current drawn from each motor, the motor temperatures, the load torque applied, and the position of each actuator. Given these measurements, the following two models for the actuators were built:

Model 1: Model 1 takes in the actuator position and load torque applied as inputs to estimate the motor current and the motor temperature. The equations for Model 1 are as follows:

$$i = \frac{1}{K_t}\left(J\frac{\mathrm{d}\omega}{\mathrm{d}t} + B\omega + \tau_L\right) \tag{4}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{C_T}(Ri^2 - K_T(T - T_a)) \tag{5}$$

Table 2 FLEA sensor suite

Sensor	Quantity	Type	Location
Load cell	1	Omega LC703-150	Between the load actuator and the test actuator coupling mechanism
Accelerometer	2	Endevco 7253C	On the nut of the ball screw
Thermocouple	4	T type	On the ball-screw nut and motor housing
Rotary encoder	3	Ultramotion E5DIFF optical encoder with differential output	On the actuator motors
Current sensor	3	Galil	On the Galil motor controller

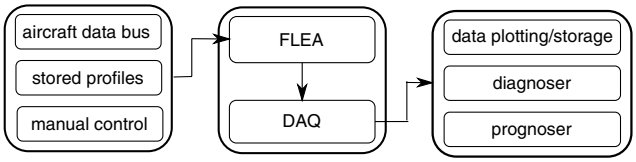


Fig. 8 FLEA control architecture.

where i is the current drawn by the actuator motor; ω is the angular velocity of the motor; J is the motor moment of inertia; B is the damping (viscous friction) coefficient; τ_L represents the opposing load torque; K_t is the torque constant; T is the motor temperature, T_a is the ambient temperature; C_T is the thermal inertia coefficient; R is the motor electrical resistance; and h is the heat transfer coefficient. In the experiments using Model 1, we assume there are no faults in sensors ω and τ_L.

Model 2: Model 2 takes in as inputs the motor current, motor temperature, and load torque applied to estimate the actuator position and motor temperature. The equations for Model 2 are as follows:

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = K_1(i + i_{\text{load}}) + K_2(i - i_{\text{load}}) + K_3 i + K_4 \tau_L \tag{6}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{C_T}(Ri^2 - K_T(T - T_a)) \tag{7}$$

where i_{load} denotes the current drawn by the load actuator motor; K_1, K_2, K_3, and K_4 are constants; and all other parameters are the same as those used in equations of the first model. In the experiments using Model 2, we assume there are no faults in sensors i and τ_L.

The diagnosis experiments using Models 1 and 2 cover a large variety of faults, including sensor faults in motor current, motor temperature, and actuator position sensors. Optimization routines are used to estimate the model parameters so as to minimize the error between the actual measurements and the measurement values predicted using the model equations. The estimated parameter values are then used in the model equations to generate high-fidelity estimates of the measurements.

A. Diagnostic Experiments Using Model 1

The actuator models are instrumental in deriving qualitative fault signatures for the different faults for motor current and motor temperature sensors, which are then used for fault isolation. Table 3 shows the different fault signatures for Model 1. Note that we have differentiated between "positive" and "negative" versions of some of the faults, such as jam and dead current sensor faults. This is because the current can be negative or positive, depending on the direction of movement of the actuator; therefore, a positive or negative deviation in the current measurements is anticipated.

Table 3 Fault signature table for diagnostic experiments with Model 1

Fault	Motor current	Motor temperature
Motor temperature positive drift	00X	0+X
Motor temperature negative drift	00X	0−X
Motor temperature positive bias	00X	+0X
Motor temperature negative bias	00X	−0X
Motor temperature positive scaling	00X	+0X
Motor temperature negative scaling	00X	−0X
Motor temperature dead sensor	00X	−∗Z
Motor temperature positive drift	0+X	00X
Motor temperature negative drift	0−X	00X
Motor temperature positive bias	+0X	00X
Motor temperature negative bias	−0X	00X
Motor temperature positive scaling	+0X	00X
Motor temperature negative scaling	−0X	00X
Motor temperature dead sensor positive	+∗Z	00X
Motor temperature dead sensor negative	−∗Z	00X
Actuator jam positive	+0X	0+X
Actuator jam negative	−0X	0−X
Actuator spall	00X	00X
Motor failure	−0Z	0−X

Table 4 Fault-feature table for diagnostic experiments with Model 1

Fault	Vibration energy
Actuator jam	*Low compared to nominal*
Actuator spall	*High compared to nominal*
Motor fault	*Zero*
Current sensor fault	*Nominal*
Temperature sensor fault	*Nominal*

Table 5 Results of diagnostic experiments with Model 1

Fault	Scenarios	Correct	% accuracy
Nominal	134	133	99.25
Motor current bias	15	15	100.00
Motor current dead	15	15	100.00
Motor current drift	15	15	100.00
Actuator position sensor failure	21	13	61.90
Motor current scaling	15	15	100.00
Jam	15	10	66.67
Motor failure	15	15	100.00
Spall	15	15	100.00
Motor temperature bias	15	15	100.00
Motor temperature dead	15	15	100.00
Motor temperature drift	15	15	100.00
Motor temperature scaling	15	15	100.00
Total	320	306	95.63

Table 4 is an example of a fault-feature table for one of the accelerometer features: *vibration energy*. The feature indicates whether the vibration energy is *zero*, *nominal*, *low compared to nominal*, or *high compared to nominal*. Based on this fault-feature matrix, and given the present set of fault hypotheses, we generate a tree data structure that provides a subset of features and the sequence in which they should be used to refine the fault hypotheses set to (ideally) a singleton set the fastest.

The hybrid diagnoser is invoked by the control module to perform online diagnosis. The data acquisition module acquires the data continuously and sends the data to the diagnoser. An observer synthesized from the system equations uses these data to determine if a fault has been detected. A qualitative fault isolation code attempts to isolate the faults resulting in an ambiguity group. The fault-detection notification, as well as the ambiguity group (as it is being refined), is communicated back to the user interface module.

The diagnoser was run on a set of predefined scenarios with varying positions (sine, trapezoidal, triangular, and sine sweep) and loads (constant load between -70 lb/-31.75 kg and $+70$ lb/31.75 kg) profiles. Some of these scenarios were nominal, while others incorporated hardware-injected faults (jam, motor failure, and spall) and software-injected faults (sensor spoofing). The results are listed in Table 5. Recall that the final outcome of our combined diagnosis approach can be a set of ambiguous faults. A diagnosis is considered to be correct as long as the ambiguous fault set is minimal and the true injected fault is included in the set.

The hybrid diagnosis method is illustrated via a jam fault example as follows. The actuator is operated in a sinusoidal position profile with a frequency of 0.5 Hz, an amplitude of 4 cm, and a load of $+40$ lb/18.14 kg. The scenario is 30 s long, with a jam fault being introduced at $t = 17$ s. At $t = 23.8$ s, a positive deviation is observed in the motor current. Based on the fault signatures shown in Table 3, the possible fault candidates are *motor current positive drift, motor current positive bias, motor current positive scaling, motor current positive dead sensor, and actuator jam fault*. All other faults are dropped from consideration. However, no other measurement deviation is observed by the time the scenario is completed. Hence, a feature-driven diagnosis approach is employed. *The energy* feature is computed and compared to nominal actuator readings, operated with a similar sinusoidal position profile. *The vibration energy* feature, computed from the experimental run, is found to be *low*. From the fault-feature table for the FLEA (Table 4), this helps to isolate (correctly) the actuator jam fault; hence, all other fault hypotheses are dropped from consideration.

B. Diagnostic Experiments Using Model 2

The experiments with Model 2 focused primarily on faults in position and temperature sensors. Table 6 shows the various fault signatures. The same features are used in this set of experiments as in the case of Model 1 (see Table 4). Note that, for diagnosing "stuck" faults, a running window scheme is implemented that checks if the incoming sensor signals are the same for a predetermined number of time steps (W_{stuck}). If so, a *stuck* fault is detected and isolated.

As an example, let us consider a motor temperature sensor drift fault with a slope of 0.3 F/s (0.167 °C/s), introduced at $t = 21$ s. The observed signatures are actual values from sensors that are inherently noisy due to the properties of the sensor and the data acquisition system. The estimated values are values calculated using a model and hence do not include any noise. As shown in Fig. 9, the estimated value is within the noise bounds of the observed signal until time $t = 28.8$ s, at which point an increase in the temperature was detected because the estimated signal is not consistent with the observed signal, even when considering the noise. Comparing this increase in temperature with the signatures shown in Table 6, the diagnoser indicates that the possible fault candidates are *motor temperature positive bias, motor temperature positive drift, and motor temperature positive stuck faults*. At $t = 41.59$ s, the $+0X$ symbol is captured, indicating that the increase in the temperature is a gradual one. As a result, all candidates, other than the true fault (*motor temperature positive drift*), are dropped from consideration, and the correct diagnosis is obtained. In this scenario, the features shown in Table 4 are not used.

The results of the diagnosis experiments using Model 2 are summarized in Table 7. Recall that a diagnosis is considered to be correct if the true injected fault is included in the diagnosis results.

Table 6 Fault signature table for diagnostic experiments with Model 2

Fault	Actuator position	Motor temperature
Actuator position positive stuck	$+*X$	$0X$
Actuator position negative stuck	$-*X$	$0X$
Motor temperature positive bias	$0X$	$+0X$
Motor temperature negative bias	$0X$	$-0X$
Motor temperature positive drift	$0X$	$0+X$
Motor temperature negative drift	$0X$	$0-X$
Motor temperature positive stuck	$0X$	$+*X$
Motor temperature negative stuck	$0X$	$-*X$

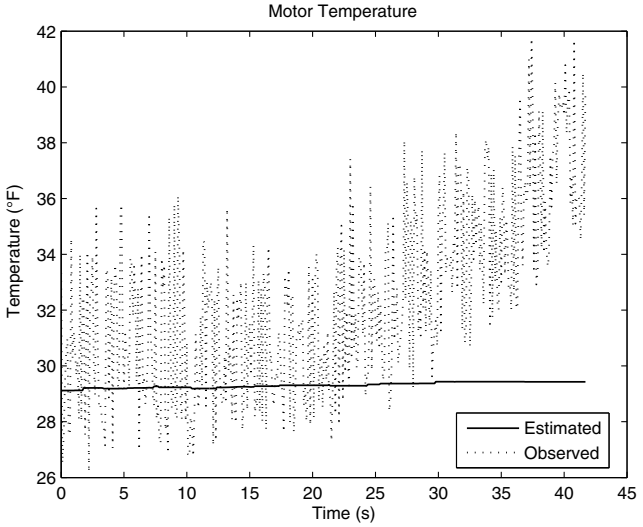


Fig. 9 Estimated and observed temperatures for the motor temperature drift fault scenario with Model 2.

VIII. Laboratory Prognostic Experiments

As determined during the initial stages of this work, jam in the return channel of a ball-screw actuator is one of the faults that is of primary concern in EMA applications. The fault scenario that was selected as motivation for this series of experiments was: a jam occurs in flight and the actuator is still needed to land the aircraft safely; estimate the remaining useful life given that the future motion and load profiles for the faulty EMA remain the same as for a healthy actuator.

To set up this experiment, a jam fault was injected into the return channel of an actuator using the technique described in Sec. VI. An operating region was then picked from the manufacturer's performance diagram (Fig. 10), where a healthy actuator can function continuously for prolonged periods of time (i.e., rated for a 100% duty cycle). Motion and load profiles were designed to stay within this region. The selected motion profile was a sine wave with an 8 cm (3.15 in.) peak-to-peak amplitude and 0.5 Hz frequency. Load was constant throughout a scenario, at −22.68 kg (−50 lb), 18.14 kg (+40 lb), or 22.68 kg (+50 lb). Motion was performed in 30 s intervals, with 15 s cooldown periods in between. Throughout the experiments, the combined electrical current to load and test actuators was limited to 6 A at 28 V dc at all times. Increased friction from the jam in the ball-screw nut resulted in additional current directed by the controller into the test actuator motor in order to attempt performing the same motion profile under the same load as a nominal actuator. This above-nominal current resulted in gradual heat buildup inside motor housing, despite the cooldown periods between motion intervals. Excessive heat eventually caused damage to winding insulation, followed by a short circuit and, ultimately, failure of the motor.

Initial experiments demonstrated that motor failure would typically occur when the temperature, as measured on the surface of the motor housing, reached approximately 85°C. Figure 11 illustrates fault progression for three different runs. The reader may observe that one of the motors on the chart lasted measurably longer than the other two; although, judging by the symptoms, damage to its motor windings insulation started to occur in approximately the same temperature region as for the other two test articles. The prognostic algorithm was executed on the collected data, using the motor housing temperature measurement as the feature, and its EOL predictions were then compared to the actual failure times. Predictions were updated as time progressed and more data became available. For illustration purposes, we show only three prediction updates at approximately 50, 70, and 85% of the total time between onset of detectable damage and EOL (Fig. 12). The results are quantitatively summarized in Table 8. The point of onset of detectable damage in an experiment such as this can be defined in a number of different ways. For the analysis presented next, this point was chosen rather conservatively at 40°C, which is the average temperature of a nominal actuator executing the same motion profiles under the same magnitude of loads. It was assumed that operating above this temperature begins the process of winding insulation deterioration. Predictions were initiated when the temperature crossed the 40°C level. The predicted mean fault progression trajectories generated at each of the three prediction points are shown in Fig. 12 with dashed lines, and the 2σ uncertainty bounds for each are illustrated with different levels of grayscale shading. As can be observed from the plots, the cone of uncertainty shrinks as the algorithm sees more data from the faulty actuator.

In the first run-to-failure experiment, the opposing load is set to 18.14 kg (+40 lb) in the compressive direction, with a sine wave position profile executed. The amplitude of the wave is 80 mm and the frequency is 0.25 Hz (the same position profile was used for the other two experiments illustrated). The failure of the motor occurs at 24.6 min (1476 s). Failure for the motor in the next experiment, with a higher, 22.68 kg (+50 lb) (compressive) load, occurs faster, in only about 9.73 min (584 s). The last scenario illustrates why it was deemed important to exercise the actuators not only in compressive direction but also in tensile. The general trajectory to failure appears to be somewhat different from the compressive runs and the EOL is not reached as quickly as in the case of an equivalent compressive load. Still, the prognostic algorithm, using the

Table 7 Results of diagnostic experiments with Model 2

Fault	Scenarios	Correct	% accuracy
Position stuck	20	20	100.00
Temperature bias	60	45	75.00
Temperature drift	60	37	61.67
Temperature stuck	60	57	95.00
Total	200	159	79.50

Fig. 10 Performance specifications for UltraMotion Bug actuators (courtesy of UltraMotion Corporation).

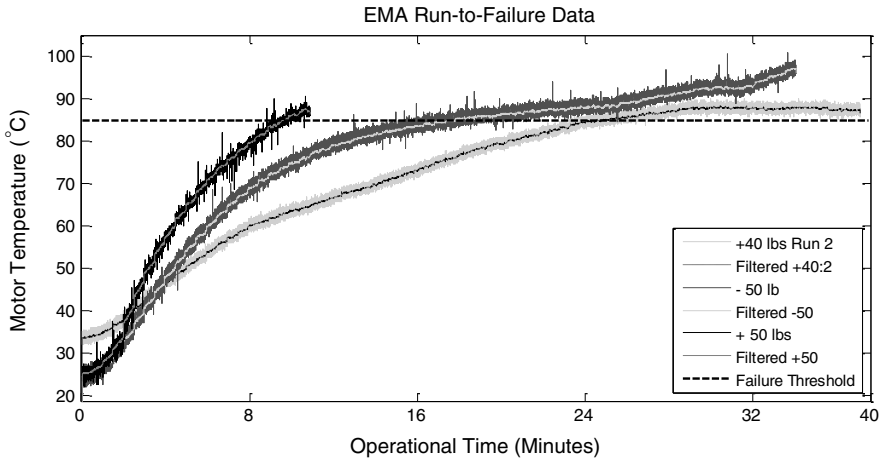


Fig. 11 FLEA run-to-failure data for motor winding failure.

same covariance function and hyperparameter initialization strategy, was able to adapt and predict the EOL with a manageable degree of error. The plots in the right column of Fig. 12 graphically show the $\alpha - \lambda$ performance of the predicted RUL in the three cases, as it evolves with time. In an ideal case, the predictions made at any given time t_p should line up with the ground truth RUL line (shown as the dotted line in the middle of the cone) and the spread should be minimal. It can be observed that, generally, in all three cases, not only the accuracy improves with time but the prediction spread is also reduced. Table 8 further summarizes the prediction results numerically using the prognostic performance metrics described in [32]. The prediction horizon (PH) metric represents the first point in time when the predicted RUL reaches within 10% of the actual failure time. A longer PH can be seen as an early warning, potentially allowing a timely mitigation action. It may be noted that, although for the −50 lb case the predictions enter the α cone on the 10th minute, the PH is computed based on the 12th-minute predictions, since predictions were not consistently inside the cone before that. Outcomes of the $\alpha - \lambda$ metric, represented as 0 or 1 in Table 8, denote whether the error is contained within the 10% error cone defined as the α bounds at various prediction time points (specified by λ). This metric visually summarizes performance (prediction accuracy and uncertainty) for a quick overview. Further, accuracy is also computed quantitatively as percentage relative accuracy (RA), where the error is normalized by the actual remaining life at any given time λ. This represents the notion that errors should reduce as end-of-life time approaches. A detailed discussion of the prognostics metrics can be found in [32].

In Table 8, "NAN" entries for RA indicate that a prediction was not available with sufficient confidence. Also noteworthy is that, even though the absolute errors in predictions generally reduce as time passes by, the relative accuracy may deteriorate close to the end of life. This shows that the GP model constructed for this application is not able to adapt well to the process dynamics toward the end. An explanation and repercussions of such behavior are also presented in [32].

IX. Flight Experiments

There have been several FLEA experiments on aircraft to date. Flights on C-17 aircraft served to mature FLEA's hardware and software, whereas subsequent experiments on UH-60 helicopters provided a rich test environment for the initial validation of the diagnostic and prognostic systems with several fault types. The UH-60 experiments were conducted over the course of eight flight hours, involving all of the usual flight regimes (straight and level flight, accelerated flight, ascents, descents, hover, and banked turns). The flights were conducted in the San Francisco Bay area airspace. Although further flight experiments are needed to draw statistically significant, quantitative conclusions (as was done in the preceding sections for laboratory experiments), the experiments conducted so far demonstrate the viability of the overall concept.

The test aircraft that hosted the FLEA, shown in Fig. 13, is an EH-60L helicopter (the electronic countermeasures variant of the UH-60L). All of the electronic countermeasures mission equipment has been removed, with the exception of the Honeywell H-423 inertial navigation unit and associated control interfaces.

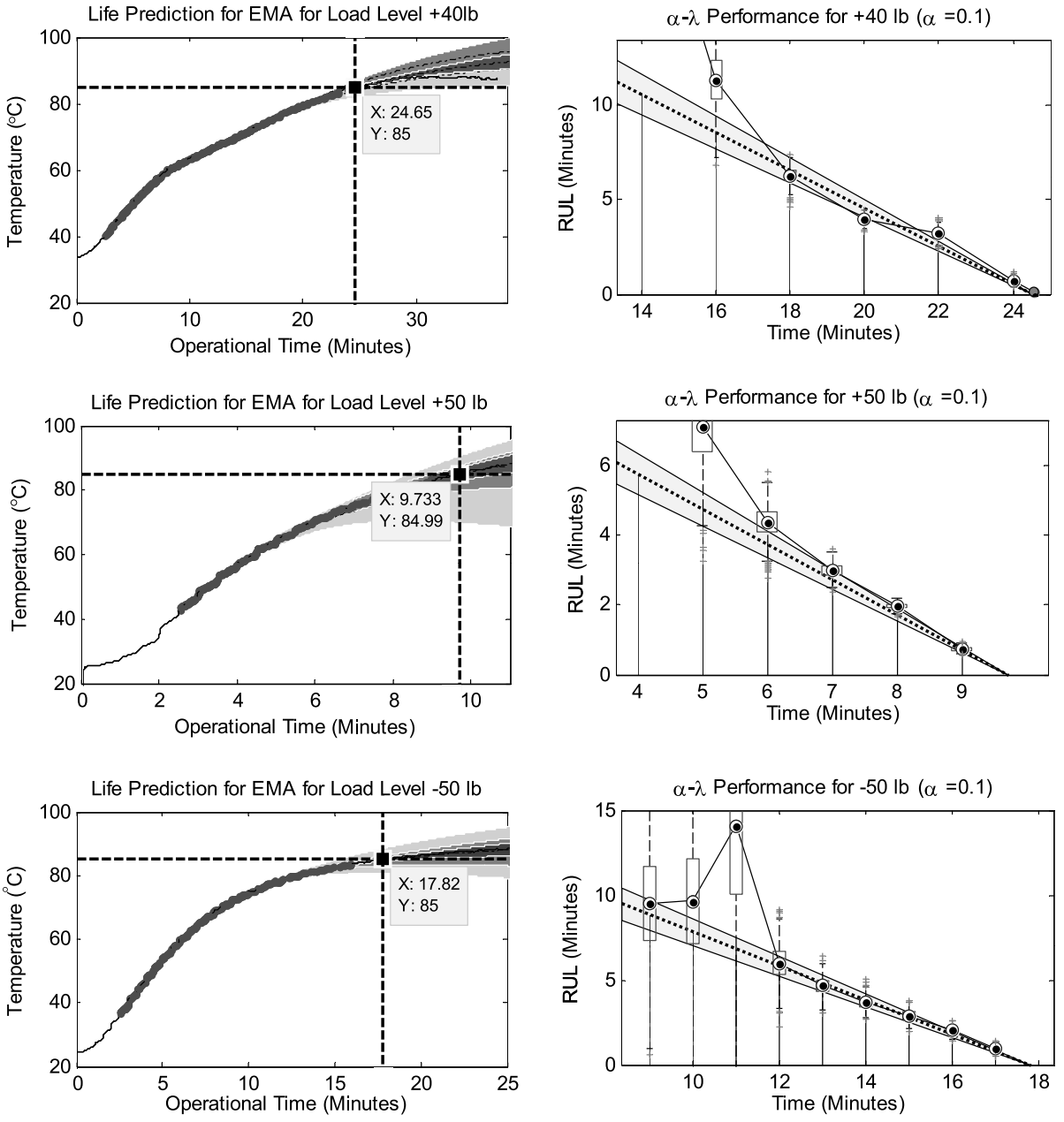


Fig. 12 RUL prediction and $\alpha - \lambda$ performance under three different load conditions.

Instrumentation racks and an airframe data system (ADS) have been installed in the cabin, and sensors have been mounted throughout the control system to provide position information on the flight controls, stability augmentation system actuators, and the primary servos. The ADS is capable of recording aircraft state data that can then be downlinked to a telemetry station. The FLEA is mounted to the floor of the aircraft cabin and is controlled by a system operator via a laptop computer. The relevant telemetry data are supplied to the FLEA by ADS.

To set up the experiments, a target actuator was selected on the aircraft. In this case, it was the forward primary servo, which is the actuator responsible for pitch control of the main rotor blades. The FLEA was connected to the vehicle's data bus, obtaining vehicle and flight environment parameters in real time (e.g., positions of the servos, rotor blades pitch angles, airspeed, altitude, and air density). During the experiments, the test stand executed motion sequences matching those of the target actuator. Load profiles executed by the FLEA's load actuator were derived using the real-time flight dynamics data and simplified load models based on the experimental data and analysis described in [33–38]. In particular, the equivalent retreating indicated tip speed (ERITS) [33] equations are used in conjunction with fixed forward pitch link peak-to-peak loads vs ERITS data for different flight regimes (examples can be seen in Figs. 14a and 14b):

$$\text{ERITS} = \left(\Omega R\sqrt{\frac{\rho}{\rho_{\text{SL}}}} - V_i\right)\sqrt{\frac{W_0}{N_Z W}} \tag{8}$$

where N_z is the load factor along the body z axis, R is the rotor blade radius, V_i is the aircraft indicated airspeed, W is the aircraft current weight, W_0 is the aircraft nominal weight, Ω is the main rotor angular velocity, ρ is the current air density, and ρ_{SL} is the sea level standard air density.

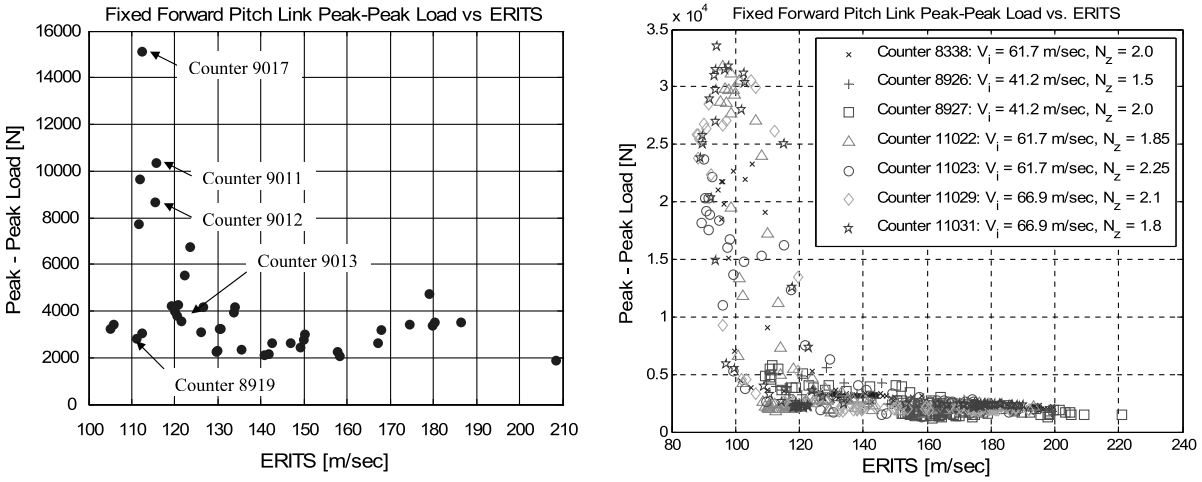
Figure 15a shows a typical motion profile executed over a period of about 20 min. Figure 15b shows the desired (computed) load profile.

Table 8 Prognostic predictions

t_p, min	$\alpha - \lambda$	RA, %
	Load = +40 lb, direction = push, EOL = 24.6 min, PH = 6.6 min	
14	0	NAN
16	0	68.0
18	1	95.2
20	0	86.0
22	0	75.3
24	0	79.1
	Load = +50 lb, direction = push, EOL = 9.73 min, PH = 3.73 min	
4	0	NAN
5	0	50.0
6	0	83.4
7	1	90.5
8	0	87.0
9	1	99.0
	Load = −50 lb, direction = pull, EOL = 17.8 min, PH = 5.8 min	
9	0	92.1
20	1	75.7
11	0	NAN
12	1	97.0
13	1	96.9
14	1	96.0
15	1	96.2
16	0	88.1
17	0	83.0

Prognostic experiments executed in flight were designed around the same fault type as the experiments described in Sec. VIII (mechanical jam leading to motor overheating). Figure 16 shows an example motor temperature prediction curve obtained during a flight experiment, where thermocouple measurements are on the left-hand side of the figure (exhibiting a certain amount of noise) and the prediction line of the GPR

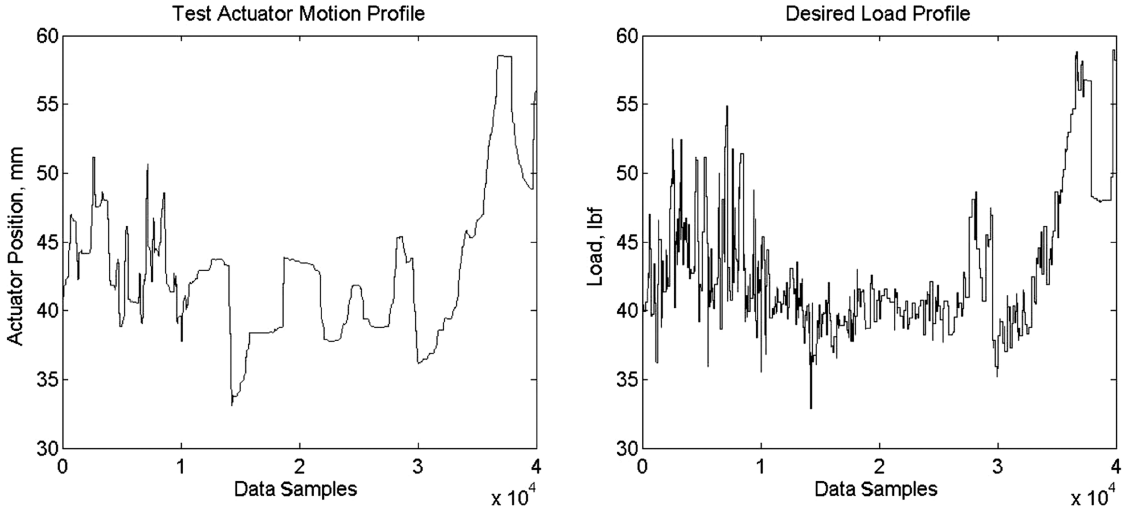
Fig. 13 FLEA mounted on the test aircraft.



a) Level-flight fixed forward pitch link load variation

b) Pullup maneuver, fixed forward pitch link load variation

Fig. 14 Example ERITS [33] data for different flight regimes.



a) Test actuator motion profile **b) Desired load profile**

Fig. 15 Test actuator motion and desired load profiles during UH-60 flight segment.

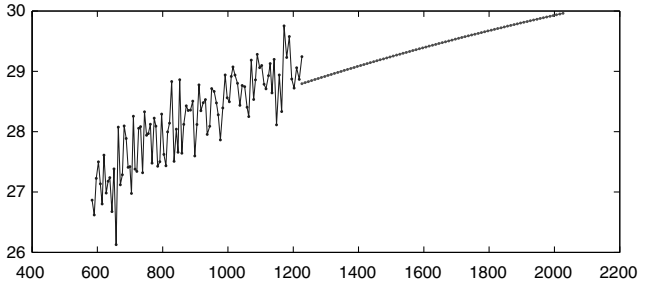


Fig. 16 Prognostic output example from a flight experiment.

algorithm extending from $t_p = 1226$ s (20 min, 26 s). Due to safety considerations, in these initial prognostics experiments, the FLEA actuators were not run until complete motor failure but were stopped when the faulty actuator motor temperature reached 60°C.

It is worth noting that this example considers a scenario where the timescale of fault growth is within the duration of a single flight; therefore, the implemented prognostics algorithm is a reasonable choice. If, however, the fault grows so slowly that there may still be one or more flights that can take place safely before it crosses the failure threshold, a long-term prediction based on suitable (potentially different) health indicators may be needed.

X. Conclusions

The work described herein is aimed at advancing prognostic health-management solutions for electromechanical actuators and thus increase their reliability and viability to designers of next-generation aircraft and spacecraft. In pursuit of this goal, a team adopted a systematic approach, starting with EMA FMECA study reviews, consultations with EMA manufacturers, and literature reviews on the related efforts. Based on the acquired knowledge, nominal and off-nominal physics models and diagnostic and prognostic algorithms were developed. To aid with development of the algorithms and validate them with realistic data, a testbed capable of supporting experiments in both laboratory and flight conditions was developed. The testbed allows for the injection of various types of faults and is equipped with a comprehensive data acquisition system. Test actuators with architectures similar to potential flight-certified units were used for testing and realistic fault-injection methods were implemented. The fault modes for the experiments were selected from electrical, data acquisition, and mechanical subsystems. Several hundred fault scenarios were created, using various position profiles, load profiles, and fault magnitude levels.

The diagnostic system was tested extensively on the scenarios, with the test results generally demonstrating high accuracy and low false-positive and false-negative rates. The prognostic system, which tracks progression of the fault once it has been detected and predicts the remaining useful life of the actuator, was demonstrated as well. A series of run-to-failure experiments were conducted to validate its performance, with generally high accuracy of predicting the time of failure. Although a more rigorous validation procedure would require many more experiments executed under the same conditions, the current results already demonstrate the potential of prognostic technology for predicting fault progression in this type of devices. Future work could include investigation and comparison of other types of prognostic algorithms, addition of new fault modes, and execution of more complex prognostic experiments in the flight environment.

Acknowledgments

The funding for this work was provided by the NASA Aviation Safety Program, Integrated Vehicle Health Management and Systemwide Safety Assurance Technologies projects. We would like to thank our colleagues at NASA Ames Research Center for both their help with this research and in preparation of the manuscript. A special thanks goes to Catlin Mattheis and Austin Lawrence, who helped to create the original Flyable electromechanical actuator (FLEA) testbed prototype while at California Polytechnic State University. Steven Braddom, formerly of the U.S. Army Aeroflightdynamics (AFFD) Directorate Flight Projects Office, provided steadfast support for the initial UH-60 flight tests. The flight tests would also not have been possible without the support of AFFD engineers, technicians, and pilots (Casey Blaskowski, Gary Leong, Gary Fayaud, Ellazar Barrientos, Richard Huber, Jay Fletcher, Juan Saucedo, Scott Miller, Ernie Moralez, Munro Dearing, Randall Watson, and Samuel Caires).

We also express our gratitude to Steven Fletcher, Bruce Felt, Pete Chaplin, Phillip Jensen, and Minh Wong for their assistance in fabricating the various components of the FLEA.

References

[1] "Loss of Control and Impact with Pacific Ocean Alaska Airlines Flight 261 McDonnell Douglas MD-83," National Transportation Safety Board TR-PB2002-910402, 2002.
[2] Jensen, S., Jenney, G., and Dawson, D., "Flight Test Experience with an Electromechanical Actuator on the F-18 Systems Research Aircraft," *IEEE/AIAA Digital Avionics Systems Conference*, Vol. 1, IEEE, Piscataway, NJ, 2000, pp. 2E3/1–2E310.
[3] Bodden, D., Scott Clements, N., Schley, B., and Jenney, G., "Seeded Failure Testing and Analysis of an Electro-Mechanical Actuator," *2007 IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2007, pp. 1–8.
[4] Byington, C., Watson, M., Edwards, D., and Stoelting, P., "A Model-Based Approach to Prognostics and Health Management for Flight Control Actuators," *IEEE Aerospace Conference*, Vol. 6, IEEE, Piscataway, NJ, 2004, pp. 3551–3562.
[5] Potter, F., and Locken, N., "Controlling and Monitoring Linear Actuators via the Power Distribution System," SAE International Technical Paper 2010-01-1764, Warrendale, PA, 2010.
[6] Smith, M., Byington, C., Watson, M., Bharadwaj, S., Swerdon, G., Goebel, K., and Balaban, E., "Experimental and Analytical Development of Health Management for Electro-Mechanical Actuators," *IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2009, pp. 1–14.
[7] Chen, Z., "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond," *Statistics*, Vol. 182, No. 1, 2003, pp. 1–69.
[8] Brown, D., Georgoulas, G., and Bole, B., "Prognostics Enhanced Reconfigurable Control of Electro-Mechanical Actuators," *Annual Conference of the Prognostics and Health Management Society*, PHM Society, Rochester, NY, 2009, pp. 1–17.
[9] Gokdere, L., Chiu, S., Keller, K., and Vian, J., "Lifetime Control of Electromechanical Actuators," *IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2005, pp. 3523–3531.
[10] Keller, K., Swearingen, K., Sheahan, J., Bailey, M., Dunsdon, J., Przytula, K., and Jordan, B., "Aircraft Electrical Power Systems Prognostics and Health Management," *IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2006, pp. 1–12.
[11] Kunst, N., and Lynn, C., "An Innovative Approach to Electromechanical Actuator Emulation and Damage Propagation Analysis," *Annual Conference of the Prognostics and Health Management Society*, PHM Society, Rochester, NY, 2009, pp. 1–8.
[12] Balaban, E., Narasimhan, S., Daigle, M., Celaya, J., Roychoudhury, I., and Saha, B., "A Mobile Robot Testbed for Prognostics-Enabled Autonomous Decision Making," *Annual Conference of the Prognostics and Health Management Society*, PHM Society, Rochester, NY, 2011, pp. 1–16.
[13] Narasimhan, S., Balaban, E., Daigle, M., Roychoudhury, I., Sweet, A., Celaya, J., and Goebel, K., "Autonomous Decision Making for Planetary Rovers Using Diagnostic and Prognostic Information," *8th International Federation of Automatic Control Symposium (SAFEPROCESS)*, IFAC, Laxenburg, Austria, 2012, pp. 289–294.
[14] Balaban, E., Bansal, P., Stoelting, P., Saxena, A., Goebel, K., and Curran, S., "A Diagnostic Approach for Electro-Mechanical Actuators in Aerospace Systems," *IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2009, pp. 1–13.
[15] Balaban, E., Saxena, A., Goebel, K., Byington, C., Watson, M., Bharadwaj, S., and Smith, M., "Experimental Data Collection and Modeling for Nominal and Fault Conditions on Electro-Mechanical Actuators," *2009 PHM Annual Conference of the Prognostics and Health Management Society*, PHM Society, Rochester, NY, 2009, pp. 1–15.
[16] Balaban, E., Saxena, A., Bansal, P., Goebel, K., and Curran, S., "Modeling, Detection, and Disambiguation of Sensor Faults for Aerospace Applications," *IEEE Sensors Journal*, Vol. 9, No. 12, 2009, pp. 1907–1917.
[17] Narasimhan, S., Roychoudhury, I., Balaban, E., and Saxena, A., "Combining Model-Based and Feature-Driven Diagnosis Approaches—A Case Study on Electromechanical Actuators," *21st International Workshop on Principles of Diagnosis (DX 10)*, DX Workshop Series, Palo Alto, CA, 2010, pp. 249–256.
[18] Hamscher, W., Console, L., and Kleer, J. D., *Readings in Model-Based Diagnosis*, Morgan Kaufmann, San Mateo, CA, 1992.
[19] Srivastava, A., "Discovering System Health Anomalies Using Data Mining Techniques," *Proceedings of the Joint Army, Navy, NASA, Air Force Conference on Propulsion*, JANNAF Interagency Propulsion Committee, Columbia, MD, 2005, pp. 1–11.
[20] Mosterman, P. J., and Biswas, G., "Diagnosis of Continuous Valued Systems in Transient Operating Regions," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 29, No. 6, 1999, pp. 554–565.
[21] Basseville, M., and Nikorov, I., *Detection of Abrupt Changes: Theory and Application*, Prentice–Hall, Upper Saddle River, NJ, 1993.
[22] Narasimhan, S., Mosterman, P. J., and Biswas, G., "A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems," *9th International Workshop on Principles of Diagnosis (DX)*, DX Workshop Series, Palo Alto, CA, 1998, pp. 94–101.
[23] Rasmussen, C. E., and Williams, C., *Gaussian Processes for Machine Learning*, MIT Press, 2006.
[24] Saha, S., Saha, B., Saxena, A., and Goebel, K., "Distributed Prognostic Health Management with Gaussian Process Regression," *IEEE Aerospace Conference*, IEEE, Piscataway, NJ, 2010, pp. 1–8.
[25] Goebel, K., Saha, B., and Saxena, A., "A Comparison of Three Data-Driven Techniques for Prognostics," *62nd Meeting of the Society for Machinery Failure Prevention Technology (MFPT)*, MFPT Society, Dayton, OH, 2008, pp. 119–131.
[26] Saxena, A., Celaya, J. R., Roychoudhury, I., Saha, B., Saha, S., and Goebel, K., "Designing Data-Driven Battery Prognostic Approaches for Variable Loading Profiles: Some Lessons Learned," *First European Conference of the Prognostics and Health Management Society 2012*, PHM Society, Rochester, NY, 2012, pp. 69–79.
[27] Mardia, K. V., and Marshall, R. J., "Maximum Likelihood Estimation of Models for Residual Covariance in Spatial Regression," *Biometrika*, Vol. 71, No. 1, 1984, pp. 135–146.
doi:10.1093/biomet/71.1.135
[28] Lehmann, E., and Casella, G., *Theory of Point Estimation*, Springer, New York, 1998, pp. 443–487.
[29] Sankararaman, S., and Mahadevan, S., "Likelihood-Based Representation of Epistemic Uncertainty Due to Sparse Point Data and/or Interval Data," *Reliability Engineering and System Safety*, Vol. 96, No. 7, 2011, pp. 814–824.
[30] MATLAB, Software Package, Ver. 7.11.0.584 (2010b), The MathWorks Inc., Natick, MA, 2010.
[31] LabVIEW, Software Package, Ver. 8.6, National Instruments Corporation, Austin, TX, 2008.
[32] Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K., "Metrics for Offline Evaluation of Prognostic Performance," *International Journal of Prognostics and Health Management*, Vol. 1, No. 1, 2010, pp. 4–23.
[33] Grill, I., Dhingra, M., and Prasad, J., "Methods for Real-Time Rotor Stall Detection," *34th European Rotorcraft Forum*, Royal Aeronautical Society, London, 2008, pp. 2443–2482.
[34] Bousman, W., "Characterization of Dynamic Stall on the UH-60A," NASA TR-TM-98-112240, 1998.
[35] Coleman, C., and Bousman, W., "Aerodynamic Limitations of the UH-60A Rotor," NASA TR-TM-110396, 1996.
[36] Kufeld, R., and Bousman, W., "High Load Conditions Measured on a UH-60A in Maneuvering Flight," *Journal of the American Helicopter Society*, Vol. 43, No. 3, 1998, pp. 202–211.
[37] Kufeld, R., Balough, D., Cross, J., Studebaker, K., Jennison, C., and Bousman, W., "Flight Testing the UH-60A Airloads Aircraft," *Proceedings of the 50th Annual Forum of American Helicopter Society*, American Helicopter Soc., Fairfax, VA, 1994, pp. 557–578.
[38] Kufeld, R., Balough, D., Cross, J., Studebaker, K., Jennison, C., and Bousman, W., "The Effects of Control System Stiffness Models of the Dynamic Stall Behavior of a Helicopter," *Proceedings of the 54th Annual Forum of American Helicopter Society*, American Helicopter Soc., Fairfax, VA, 1998, pp. 589–601.

A. Srivastava
Associate Editor