Experimental Validation of a Prognostic Health Management System for Electro-Mechanical Actuators

Edward Balaban[1]
NASA Ames Research Center, Moffett Field, CA 94035, USA

Abhinav Saxena[2]
SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA

Sriram Narasimhan[3]
UARC, NASA Ames Research Center, Moffett Field, CA 94035, USA

Indranil Roychoudhury[4]
SGT Inc., NASA Ames Research Center, Intelligent Systems Division, CA 94035, USA

and
Kai Goebel[5]
NASA Ames Research Center, Moffett Field, CA 94035, USA

Electro-Mechanical Actuators (EMA) are gaining prominent roles in the next generation fly-by-wire aircraft and spacecraft. With these roles often being safety-critical (control surface or landing gear actuation, for instance), the key to faster adoption of EMA in aerospace applications is development of accurate and reliable prognostic health management (PHM) systems that not only detect and identify faults, but also predict how the identified they affect the remaining useful life (RUL) of both the faulty component and the actuator as a whole. Such information can be invaluable to pilots, controllers, and maintenance personnel in assessing how to complete or re-plan the desired mission with a sufficient safety margin. A team consisting of members of NASA Ames Diagnostic & Prognostic Group has developed a prototype PHM system for EMA that provides coverage for a number of faults modes typical to this type of actuators. The diagnostic portion of the system is implemented using a hybrid approach which utilizes both qualitative (bond graph, model-based) and quantitative (data-driven) reasoners to achieve low false positive and false negative detection rates and a high accuracy of diagnostic output. Once a fault has been diagnosed, the prognostic component, which is implemented using Gaussian Process Regression (GPR) principles, estimates the RUL of the component that is faulted. Experiments were conducted both in laboratory and flight conditions to validate the PHM system using an innovative Flyable Electromechanical Actuator (FLEA) test stand. The test stand allows experimental actuators to be subjected to environmental and operating conditions similar to what actuators on the host aircraft are experiencing, while providing researchers with the capability to safely inject and monitor propagation of various fault modes. Prognostic run-to-failure experiments were done in laboratory conditions on ball-screw jam and motor winding short faults. Flight experiments (albeit not run-to-failure) were conducted in collaboration with the US Army on UH-60 Blackhawk helicopters. The paper describes these experiments in detail and presents the results obtained from the PHM system with regard to the estimation of the RUL of the actuator.

[1] Research Scientist, Intelligent Systems Division, MS 269-3
[2] Research Scientist, Intelligent Systems Division, MS 269-4, AIAA member
[3] Research Scientist, Intelligent Systems Division, MS 269-4
[4] Research Scientist, Intelligent Systems Division, MS 269-3
[5] Senior Scientist, Intelligent Systems Division, MS 269-4

Nomenclature

PHM	=	Prognostic Health Management
FLEA	=	Flyable Electro-mechanical Actuator testbed
GP	=	Gaussian Process
GPR	=	Gaussian Process Regression
EMA	=	Electro-Mechanical Actuator
EMI	=	Electro-Magnetic Interference
IVHM	=	Integrated Vehicle Health Management
RUL	=	Remaining Useful Life
EoL	=	End of Life
y(t)	=	observed measurement
$\hat{y}(t)$	=	measurement estimate
r(t)	=	residual
f(x)	=	Gaussian process (GP)
m(x)	=	mean function for GP
k(x,x')	=	covariance function for GP
y	=	noisy observations from the system
x	=	set of training points
ε	=	additive IID Gaussian noise with $N(0,\sigma n2)$

I. Introduction

There is an increasing trend in the designs of new aircraft and spacecraft to move to fly-by-wire controls and away from the more traditional hydraulic control and actuation methods. There are several types of actuation mechanisms being utilized in such fly-by-wire designs, with electro-mechanical actuators being one of them. Given the fact that actuators are usually some of the more safety-critical components of an aerospace system, an undetected or unmanaged actuator failure can lead to serious consequences – as has happened on multiple occasions in the past. For instance, the tragedy of Alaska Airlines MD-83 Flight 261 occurred due to horizontal stabilizer electro-mechanical actuator failing because of insufficient lubrication and excessive wear of its jack screw[1]. Even though actuators have been studied extensively from a functional point of view – in order to help develop new and improved designs – studies from a health management point of view have been rather limited. The reason for that is largely attributed to unavailability of operational fault data from fielded applications and lack of experimental studies with seeded fault tests due to high risks and costs involved. EMAs in aerospace systems operate in complex environmental conditions, so their inherent characteristics need to be studied thoroughly in order to be distinguishable in flight environment and enable effective diagnostics and prognostics with reduced uncertainty. This calls for a systematic, methodical effort towards understanding the EMAs and their behavior under various fault conditions through affordable, but realistic experiments.

This is the type of approach that our team strived to adopt in pursuing this research. In the early stages of the project, a thorough literature review to understand the state-of-the-art at the time was conducted. Partnerships to exchange information were established with some of the other organizations active in this area. Examples of prior efforts in this field include research into potential EMA faults modes[2, 3], fault modes modeling for diagnostic and prognostic purposes[4-7], diagnostic and prognostic algorithm development[4, 5, 8, 9], and experimental testing[6-10]. While these and other worthy efforts addressed some of the issues required for EMA prognostic health management separately, the authors felt that there was still a need for further work on integrated diagnostic/prognostic health management systems for EMA and test methods for them in relevant conditions. This is what motivated the research and experiments described in this paper.

II. Approach

As mentioned in the introduction, the team strived to adhere to a methodical approach to EMA health management research. The work began by conducting an extensive literature review, where prior work at Moog Corporation, Impact Technologies, University of Texas Austin[11], among other institutions, was examined. The team also collaborated with Moog Corporation and Impact Technologies on reviewing existing Failure Modes, Effects, and Criticality Analysis (FMECA) documents in order to compile a prioritized list of fault modes for a further study.

Like any complex mechanism, EMAs could, potentially, experience a variety of fault modes, from mechanical jams to power transistor deterioration. It would have been impossible to cover all of them in this work; therefore the fault modes that were deemed the most likely and/or most consequential were included in the list.

Given the prioritized list of EMA faults, the Ames team (in collaboration, once again, with Impact, Moog, Georgia Institute of Technology, and others) embarked on physical modeling efforts. The physics-based approach was chosen as one with the most promise to provide an accurate prognostic picture of fault progression, but with the understanding that it would be complemented by data-driven machine learning techniques, when appropriate. Modeling efforts ranged, for instance, from creation of high-level models for EMA operation[10, 12], to more in-depth studies of motion of bearing balls inside ballscrew raceways and effects of lubricants on the collision of these balls with each other and the metal surfaces of the ballscrew. Models for winding short effects[9] and motor temperature in case of progressive winding shorts were also developed. While not all of these models ended up being selected for use with health management algorithms, all of them provided crucial insights into the specifics of EMA operation in different conditions.

Development of health management (diagnostic and prognostic) algorithms followed. First in line was a diagnostic system based on a neural network[13]. It was capable of diagnosing such fault modes as ballscrew return-channel jams, surface spalls, and sensor faults of varying magnitudes. For the next version of the diagnostic system a hybrid (model-based/data-driven) approach was adopted. This system is described in detail in Section III. Finally, a prognostic system covering a subset of faults was created using Gaussian Process Regression methodology (also described in Section III). Work to expand and improve both diagnostic and prognostic algorithms is currently underway.

In order to verify performance of the algorithms in relevant environment, EMA testbeds were utilized. The first one was constructed at Moog Corporation. Data collected on that test stand was used in verifying performance of the neural network diagnostic classifier[13]. While a laboratory test stand made it is possible to simulate some of the desired flight conditions on the ground, testing equipment and algorithms in the presence of vibrations, noise, G-loads, and temperature changes inherent to flight was deemed to be important. Thus was born the idea for the FLEA – Flyable Electromechanical Actuator testbed.

The FLEA is a self-contained, lightweight test fixture containing three actuators: one nominal, one injected with faults, and the third providing dynamic load. The load is switched in-flight from the healthy to the faulty test actuator, thus allowing to collect both baseline and off-nominal sensor data under the same conditions. The testbed flies on a host aircraft, mimicking one of its control surface actuators. Motion and load profiles for whichever test actuator is active at that moment are derived from the corresponding real-time values for the host aircraft actuator Data collected on the stand is routed to a 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. While the main purpose of the FLEA is to perform data collection in real-time flight conditions, experiments utilizing FLEA can (and have) been performed in a laboratory. Laboratory experiments are useful for verifying and troubleshooting performance using well-defined scenarios. Flight and laboratory experiments involving the FLEA are described extensively in Section V. Careful consideration was given to design of these 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 too are also covered in Section V.

III. FLEA Testbed

The key idea in designing and building the FLEA test stand used in the experiments was 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. 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 modify the 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 from the corresponding real-time values for one of the aircraft's control surfaces. 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 illustrated in Figure 1 and the actual FLEA is shown in Figure 2.

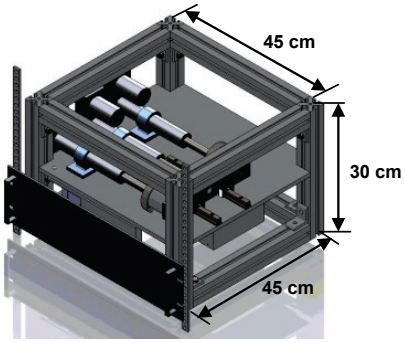

Figure 1 : FLEA Engineering Model

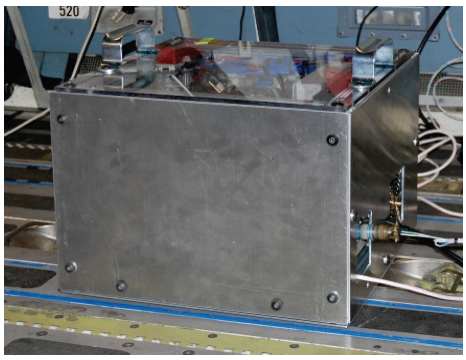

Figure 2 : FLEA aboard an aircraft with protective panels mounted

The 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. Rigidity of the central plate was an important design consideration; therefore analysis was performed that showed 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 in case of a crash and to provide EMI protection. The top portion of the stand, where EMI emissions from electronic components are contained by the center plate, is protected with a thick panel of high-strength Lexan™, which allows the operator to visually observe the test in progress.

The overall weight of the FLEA, with shielding attached, is approximately 35 kg. The stand is typically mounted either in the aforementioned instrumentation rack or strapped/bolted to the floor of the fuselage. The processing unit, running the operating system, data acquisition, control, and health management software, is based on an off-the-shelf Pentium4 3.2GHz ATX motherboard. Storage is provided by two solid-state drives, one for the operating and control software, the other dedicated to data storage. Solid-state drives were chosen over traditional hard drives for their ability to operate at high altitudes without the need for pressurization.

Table 1. FLEA Sensor Suite

Sensor	Qty	Type	Location
Load Cell	1	Omega LC703-75	Between the load and test actuators
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	2	UltraMotion E5DIFF optical encoder with differential output	On the test actuator motor
Linear Potentiometer	1	UltraMotion Precision linear potentiometer	Along the load actuator screw
Voltage Sensor	3	custom	Motor controller boards
Current Sensor	3	custom	Motor controller boards

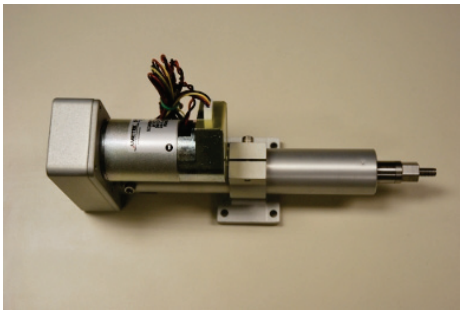

Figure 1 : UltraMotion Bug Actuator

The data acquisition system consists of two NI 6259 cards and supports a comprehensive sensor suite, described in Table 1. One of the cards collects low speed (1 KHz) data from current, voltage and temperature sensors. The other reads accelerometer channels at a higher speed (20 KHz). The acquired data is used for a variety of purposes, including plotting for visual inspection, saving to file for future analysis, and sending to diagnoser and prognoser for online reasoning. The actuators are controlled through a multi-axis Galil motion controller. Coupling of test actuators to the load actuator is accomplished via electro-magnets with only one test actuator at a time is normally coupled to the load actuator.

The test articles used in the FLEA at present are UltraMotion Bug actuators (illustrated in Figure 1). While architecturally equivalent to the larger (and considerably more expensive flight-qualified EMAs), these off-the-shelf units allow the team to conduct run-to-failure experiments in a cost-effective manner. The faults are injected into the test articles in the following manner:

- Jam: via a mechanism mounted on the return channel of the ballscrew that allows to stop circulation of the bearing balls through the circuit
- Spall: by introducing cuts of various geometries via a precise electro-static discharge process. The initial size and subsequent growth of these cuts are confirmed by using an optical inspection and measurement system
- Motor failure: injected by redirecting current from the affected motor into a sink load
- Sensor faults (bias, drift, scaling, and complete failure): software-injected into the measurements collected by the data acquisition system

The system software consists of an actuator control system, data acquisition system, flight interface system, a diagnostic system and a prognostic system. The higher level software is implemented in LabVIEW™; however the underlying algorithms for the diagnoser and prognoser are implemented in MATLAB™. The diagnostic system 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 prognostic system is tasked with monitoring the sensor data and determining how the fault is progressing and how long of a useful life is remaining for the system. The details of the diagnostic and prognostic algorithms are described in Section IV. The LabVIEW interface is designed to send the acquired data to the prognoser and display the remaining useful life estimates received from it. Other software elements include flight interface modules for getting data from different types of host aircraft and software for running experiments under various conditions in laboratory environment.

IV. Health Management System

A. Diagnostic System

Diagnostic approaches can be broadly divided into two types: model-based and data-driven[14]. Model-based schemes rely on a system model built from *a priori* knowledge about the system, while data-driven schemes do not require such models. They instead require large sets of exemplar failure data, which is often not available. Some of the sensors typical to EMAs, such as current and voltage sensors, can be modeled using physics-based differential equations and then utilized for model-based diagnosis of faults in EMA. The modeling of accelerometers, however, is non-trivial and therefore a data-driven, feature-based diagnosis approach would be better-suited to leverage accelerometer information for disambiguating faults. The diagnostic approach presented in this paper synergistically combines model-based and data-driven diagnosis techniques in order to improve upon either approach implemented individually.

The model-based approach works well when analytical models for all the faults under consideration (and how these faults affect observed measurements) can be derived. It may not be possible, however, to do so for all faults. On the other hand, the feature-driven approach requires large amounts of data under varying experimental conditions for training the diagnostic classifier. Additionally, when the classifier has to consider all faults and other experimental conditions, the size and complexity of the classifier often becomes intractable.

In this paper a hybrid method that combines these two approaches (TRANSCEND approach[14]) is adopted as the basis for the diagnostic system. The features to be used in the feature-driven approach are selected based on a diagnosability analysis. The approach consists of an offline and an online stage, described below.

1. Offline Stage

In the offline stage a Bond Graph (BG) model[15] of the system is derived. A qualitative diagnosability analysis on this model is then performed. The BG model can be used to generate qualitative signatures for all faults represented by changes in the parameters of the BG. By comparing the qualitative signatures the ambiguity groups (groups of faults that have the similar fault signatures) can be identified. These groups represent faults that need to be disambiguated using the feature-driven approach.

For each ambiguity group, a set of features is extracted (using domain knowledge or by experimentation). These features are identified to contain diagnostic information to disambiguate the maximal number of faults in that group. This results in a fault-feature table that indicates how specific features are influenced by faults.

2. Online Stage

The online stage is carried out in two phases. In the first phase the TRANSCEND approach[14] is used to observe the system, detect and qualitatively isolate fault ambiguity groups. Figure 4 illustrates the TRANSCEND diagnosis architecture. The *observer* takes as inputs the control inputs sent to the system, and the measurement readings obtained by the sensors to track the system dynamics, and estimate the unobservable system states. For *fault detection*, TRANSCEND uses a statistical Z-test 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 statistically significant deviation is detected in any one measurement, the symbol generation module is initiated, and every measurement residual, i.e. the difference of the estimated and measured sensor values,

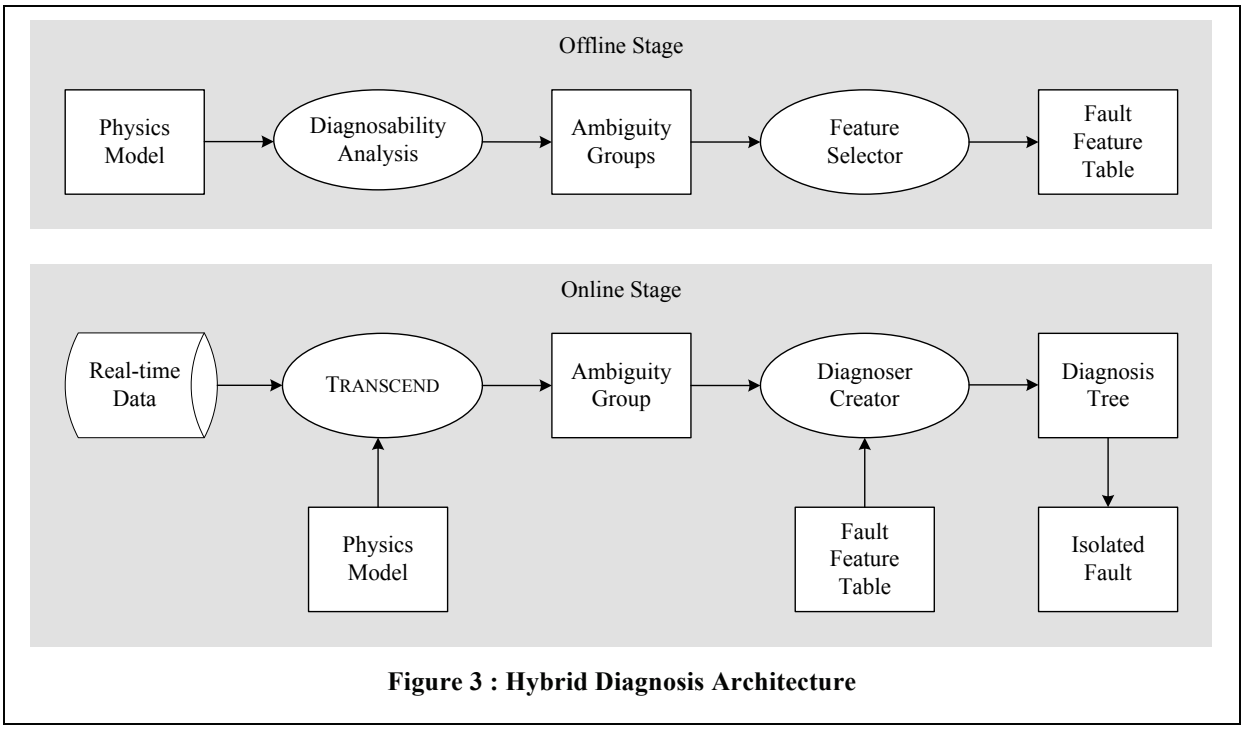

Figure 3 : Hybrid Diagnosis Architecture

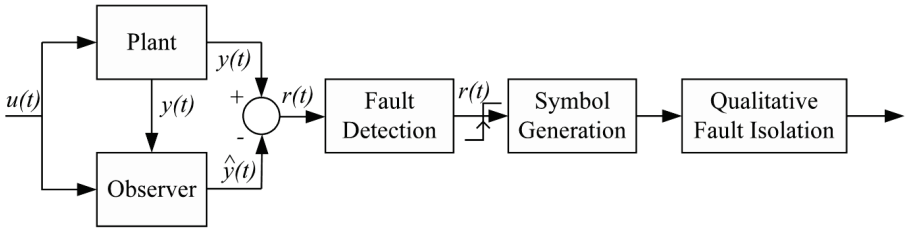

Figure 4: TRANSCEND Diagnosis Architecture

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. To generate the fault hypotheses, TRANSCEND use a qualitative diagnostic model called the Temporal Causal Graph (TCG), which is essentially a signal flow graph, whose nodes represent system variables, edges denote causality information, and edge-labels denote how one variable affects another, either immediately, or over time. Fault hypotheses are generated by propagating the first observed deviation backwards through the TCG. Then, for each fault hypothesis, the direction of change is propagated forward along the TCG to generate *fault signatures*, i.e. 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, the discussion is restricted to single faults). Given the fault signatures, qualitative fault isolation 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 from consideration, thereby refining the fault hypotheses set. If the qualitative deviations in measurements alone cannot help discriminate some faults, these are termed aggregate faults and considered belonging to an ambiguity group. Ideally, it is desired that the qualitative scheme reduces the fault hypotheses set to a singleton set. But this is not always possible in real world engineering systems. For example, in the FLEA, the spall and jam faults have similar effects on all non-accelerometer sensors. Hence, qualitative model-based approach alone is not sufficient. Therefore a data-driven approach is employed to further disambiguate faults and obtain better diagnosis results, constituting the second phase of the process.

In the second phase the isolated ambiguity group triggers the selection of rows from the fault-feature table. These rows correspond to the faults in the ambiguity group. The selected sub-table can then be converted to a diagnoser tree using the measurement selection procedure[15]. 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 starting ambiguity group. At each level, of the tree features are selected that partition the ambiguity group in the most balanced fashion. This can be formally specified as the partitioning with the least difference between the largest and smallest partitions. Once the best feature has been identified, the ambiguity group is partitioned into sub-groups corresponding to the possible values for the selected feature (one sub-group for each possible feature value). For each sub-group of size greater than 1 the next best feature is selected, which creates the most balanced partitions. This process continues until only sub-groups of size 1 are left or there are no more features left to select. If there are any sub-groups of size greater than 1 then it indicates indistinguishable faults.

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

The construction of the diagnoser tree can, actually, be done offline for the set of identified ambiguity groups in order to reduce some computation at run time. However if more computational power is available, a ‘lazy’ approach might be best. 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. This procedure (computing only first-best feature) is for the current ambiguity group. This process, again, is repeated until a single fault has been isolated or all features have been computed.

A. Prognostic System

Once a fault is detected and isolated, its rate of evolution must be tracked and assessment for the remaining useful life (RUL) should be made to manage contingency in a timely and safe manner. A Gaussian process regression (GPR) based prediction algorithm was chosen for this phase of the work. GPR does not need explicit fault growth models and can be made computationally less expensive by sampling techniques. Further, it provides variance bounds around the predicted trajectory to represent the associated confidence.

A Gaussian Process (GP) is a collection of random variables any finite number of which has a joint Gaussian distribution[18]. A real GP f(x) is fully specified by its mean function m(x) and co-variance function k(x,x') defined as:

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

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

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

Domain knowledge available from the process is encoded by the covariance function that defines the relationship between data points in a time series. Ideally, GPR requires prior knowledge about the form of covariance function, which may be inferred from the application domain. Covariance functions consist of various hyper-parameters that define the temporal characteristics of fault growth. Setting the right values of such hyper-parameters is key in learning the desired functions. A covariance function must be specified *a priori*, but corresponding hyper-parameters 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 hyper-parameters[3].

It is expected that once the diagnostic system indicate the onset of a fault mode, the prognostic system is triggered. Data is processed in real-time to extract relevant features, which are used by the GPR algorithm for training during initial period. The longer is the training period, the better are the chances for the algorithm to learn 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 is set for the training period. Once this limit is reached the algorithm starts predicting fault growth trajectories. End of Life (EoL) is subsequently determined by where these trajectories intersect the preset fault level threshold. As time progresses, the GPR model is updated with new observations and, subsequently, the predictions are updated as well.

Owing to its computational complexity of $O(n^3)$, GPR runs into scalability issues if a long data history is used. However, by sampling the training points from the accumulated observation data for the given system, this problem can be addressed. A suitable number of data points are sampled uniformly from the all available history data to train the system. Using a maximum-likelihood optimization these data points are used to determine the best fitting hyper-parameters for the chosen covariance function. It must be noted that since this process involves numerical methods, the outcome of the optimization often depends on the initialization of the hyper-parameters, which adds to uncertainty in the predicted outcomes. In the presented implementation this uncertainty is characterized and handled in two ways. First, at each prediction time instant (t_p) a large (p~50) number of samples of k data points are drawn from the observed data. Then p different GP models are trained based on these p different sample sets. By averaging the results from these p different GPRs we expect to eliminate variability emerging from random sampling, while still keeping the computational cost low. In other words, computational complexity is reduced from $O(n^3)$ to $O(p.k^3)$, where p, $k<<n$ as time passes by. A similar concept has been explored previously where temporal sampling was used to parallelize the algorithm and reduce its computational complexity[19].

V. Experiments

A. Diagnostic Experiments

One of the first steps for model-based diagnosis is to estimate the model parameters. The state-space equations representing the physics-based nominal model of the EMA are given below.

$$\frac{di}{dt} = \frac{1}{L}(V - Ri - K_m\omega), \tag{4}$$

$$\frac{d\omega}{dt} = \frac{1}{J}\left(K_t i - B\omega - \tau_L\right) \tag{5}$$

where i denotes the current drawn by the actuator; ω denotes the angular velocity; and K_t and K_m denote the torque and motor constants, respectively. The EMA is essentially modeled as a DC motor with input voltage, V; with winding inductance, L; winding resistance, R; damping coefficient, B; and opposing load torque, τ_L.

To aid parameter estimation, nominal data was collected by running the FLEA under different load profiles. Then a MATLAB Simulink model of the EMA was developed which accepts the same inputs that were given to the FLEA and estimates the variables that are measured by the available sensors. Note that in this estimation scheme, all but the accelerometer readings were used, since, as mentioned prior, modeling of accelerometers is non-trivial. Finally, an optimization routine was executed to estimate model parameters that would minimize the error between the actual and predicted values of the available measurements. The estimated parameter values are then included in the state-space equations of the particle filter observer to generate high fidelity estimates of unknown hidden states. The BG model of the actuator is used to derive the Temporal Causal Graph which, in turn, is instrumental in deriving the qualitative fault signatures for the different faults. These fault signatures are used in the qualitative fault isolation.

The fault-feature signature matrix for some EMA faults includes accelerometer features (standard deviations of each accelerometer reading, currently). The signatures represent whether or not a feature will be affected by a fault, and denoted by a '1' or '0', respectively. Based on this fault-feature signature matrix (and given the present set of fault hypotheses) a tree data-structure is generated. The data-structure 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.

Data is acquired continuously and sent to the diagnoser (which is initialized when the system control software starts). An observer synthesized from the bond graph models uses this data to determine if a fault has been detected. Qualitative fault isolation code attempts to isolate the faults, resulting in an ambiguity group. The next best feature selector, as well as the feature extractor, are also implemented in MATLAB. The fault detection flag, as well as the ambiguity group as it is being refined, are communicated back to the LabVIEW user interface module.

As mentioned in Section III, FLEA consists of two test actuators, one of which is nominal and other injected with a fault. Different types of faulty actuators (jammed, spalled, etc.) can be switched in to reproduce occurrence of the corresponding fault. The nominal actuator is active for certain duration of time, after which the control system switches operation to the faulty actuator (the switch itself should be transparent to the diagnoser). The diagnoser is expected to continuously monitor the data and determine if and when the switch occurred (identifying it as an occurrence of a fault in the nominal actuator rather than a switch event).

The diagnoser was run during execution of a set of pre-defined scenarios with varying position (sine, trapezoidal, triangular, sine sweep) profiles and load (constant load between -70 and +70 pounds) profiles. Some of the scenarios were kept nominal, others incorporated hardware-injected (jam, motor failure, and spall) and software-injected faults (sensor faults).

B. Prognostic Experiments

As determined during the literature review and conversations with actuator manufacturers and aircraft companies, jam in the return channel of a ball-screw actuator is a fault that is of a serious concern in EMA applications The fault scenario that was picked as motivation for this series of experiments was the following: assuming that the jam occurs in-flight and the actuator is still needed to land the aircraft safely, can we estimate the remaining useful life given that the motion and load profiles remain the same as for a healthy actuator? If a helicopter is used as the example aircraft and this particular actuator controls the pitch angle of the main rotor blades (a critical function to safety of flight), how much time would there be available before the vehicle needed to land?

To set this experiment up, jam was injected into the return channel of an actuator using the technique described in subsection A. A region then was picked from the manufacturer-defined performance specification where a healthy actuator can operate continuously for prolonged periods of time (i.e. rated for a 100% duty cycle). Motion and load profiles were designed to stay inside this region. Motion profile was a sine wave with 8 cm (3.15 in) peak-to-peak amplitude and 0.5 Hz frequency. Load was constant throughout a scenario at -50, +40, or +50 lbs. Motion was performed in 30 second intervals, with 15 second cool-down periods in-between. Throughout the experiments current to load and test actuator combined was limited to 6A at 28V DC at all times. Increased friction from the jam in the ballscrew 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 build-up inside motor housing - despite the cool-down periods between motion intervals.

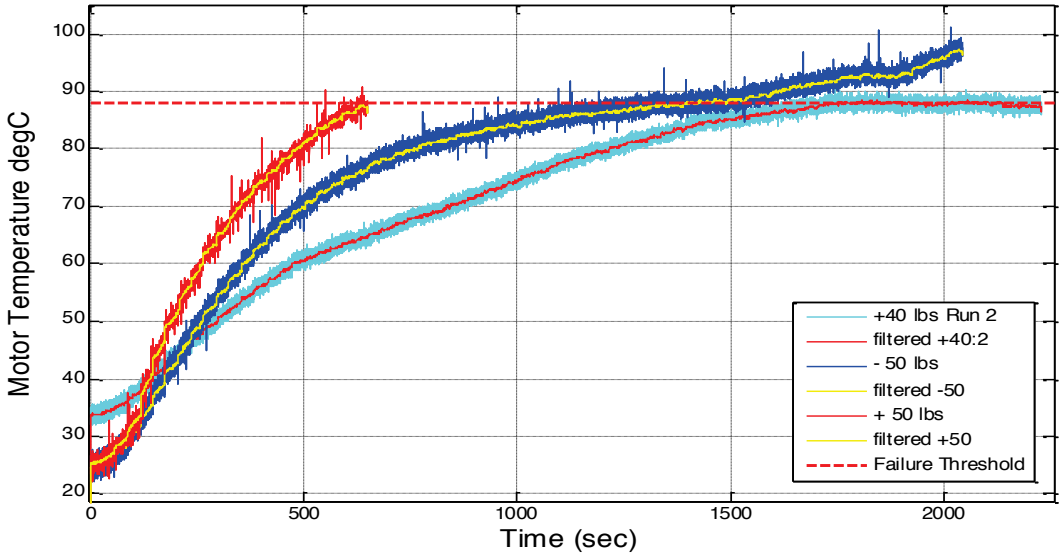

Figure 5 : FLEA Run-to-failure Data

Excessive heat eventually caused damage of winding insulation, short circuit, and failure of the motor. Figure 5 shows the run-to-failure experiments data.

C. Flight Experiments on UH-60 helicopters

There have been several FLEA experiments flown on UH-60 helicopters to date. The experiments, which were conducted at NASA Ames on the helicopters operated by US Army Aeroflightdynamics Directorate, provided valuable data sets on nominal and spall-injected actuators. This section offers a brief overview of the data collected on the UH-60, with a more extensive description and analysis to be done in a separate publication.

During the experiments, the test stand executed rigorous motion sequences, matching those of the target UH-60 actuator (forward primary servo, an actuator responsible for pitch control of the main rotor blades). Load profiles executed by the FLEA's load actuator were derived using flight conditions information (obtained from the aircraft data bus), as well as some of the models developed by NASA's Subsonic Rotary Wing Project. Figure 6 shows a typical motion profile executed over a period of about twenty minutes. Figure 7 shows the desired (computed) load profile. Spall fault was injected during the flight by switching the load path from the nominal to the faulty actuator. All the usual sensor readings were recorded.

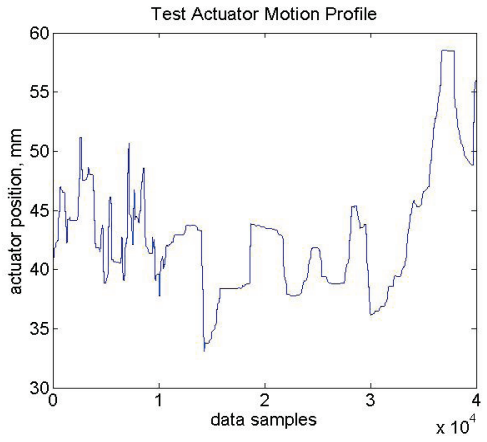

Figure 6: Test Actuator Motion Profile during a UH-60 Flight Segment

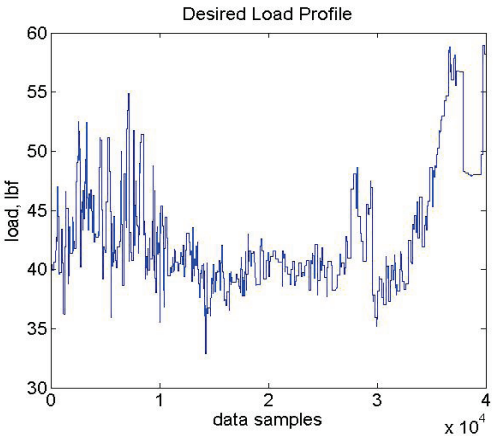

Figure 7: Desired Load Profile during a UH-60 Flight Segment

VI. Results

The diagnostic results are presented in Table 2. Recall that the final outcome of the 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.

Initial diagnostic experiments conducted aboard UH-60 helicopters (with jam and spall faults injected) showed promise as well, however more data needs to be collected to reach any quantitative conclusions.

In the case of the prognostic experiments, the initial set demonstrated that motor failure would typically occur when the temperature (as measured on the surface of the motor housing) reached approximately 88 degrees Celsius. Figure 5 illustrates fault progression for three of the runs. It may be observed 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 specimens. The prognostic algorithm was executed on collected data, using motor housing measurement as the feature, and its EoL predictions were then compared to the actual failure times for the test articles. Predictions were made at an approximately half-way point between the onset of detectable damage and EoL. 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 below, this point was chosen rather conservatively – at 40 degrees Celsius, the settling temperature of a nominal actuator executing the same motion profiles under the same class of loads.

Table 2. Diagnostic Results

Fault	Scenarios	Correct	Accuracy
Nominal	134	133	99.25
Current Sensor Bias	15	15	100.00
Current Sensor Failure	15	15	100.00
Current Sensor Drift	15	15	100.00
Position Sensor Failure	21	13	61.90
Current Sensor Scaling	15	15	100.00
Return Channel Jam	15	10	66.67
Motor Failure	15	15	100.00
Screw Surface Spall	15	15	100.00
Temp Sensor Bias	15	15	100.00
Temp Sensor Failure	15	15	100.00
Temp Sensor Drift	15	15	100.00
Temp Sensor Scaling	15	15	100.00
Total	320	306	95.625

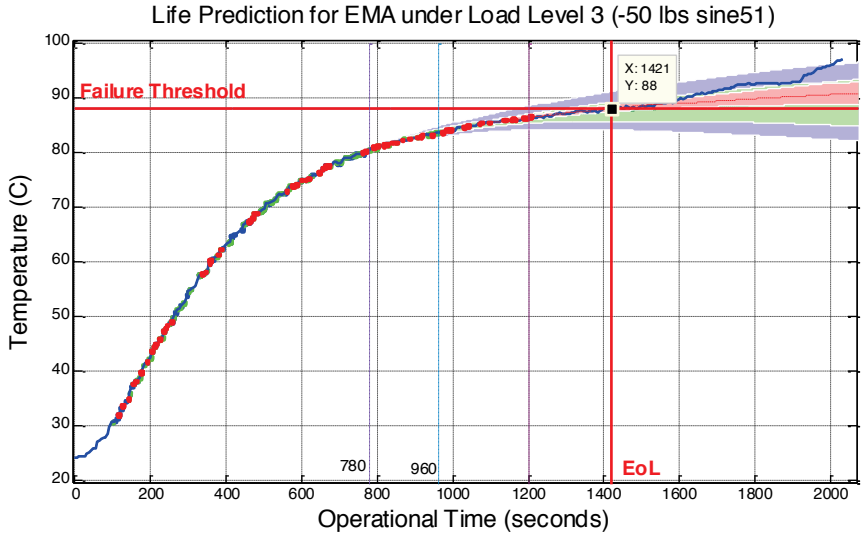

Figure 8 : Example of a Remaining Useful Life Prediction

It was assumed that operating above this temperature begins the process of winding insulation deterioration. In the first scenario, with a +40 lbs load the failure occured at around 1750 seconds. Failure for the motor in the next experiment, with a higher, +50 lbs (compressive) load, occured faster - in only about 640 seconds. The third scenario, depicted on Figure 9, demonstrated why it was important to test actuator loading not only in compressive direction, but also in tensile. The general feature trajectory to failure was somewhat different from the compressive runs and the EoL was not reached as quickly as in case of an equivalent compressive load (around 1420 seconds). Still, the prognostic algorithm, using the same covariance function and hyper-parameter initialization strategy, was able to adapt and predict the EoL with a good degree of error – only about 8%. Table 3 summarizes the prediction results for the above experiments. In addition to predictions made at t_{p1}, similar predictions were made at approximately t_{p2}=70% and t_{p3}=85% of remaining useful life.

Table 3: Prognostic Predictions

Load Level (lbs)	Direction	t_p	MAPE (%)	2σ(s)
+40	Push	890	6.62	215
+50	Push	360	4.76	60
-50	Pull	890	8.02	338

VII. Conclusions

The work described herein is aimed to advance prognostic health management solutions for electro-mechanical actuators and, thus, increase their reliability and attractiveness to designers of the next generation aircraft and spacecraft. In pursuit of this goal the team adopted a systematic approach by starting with EMA FMECA reviews, consultations with EMA manufacturers, and extensive literature reviews of previous efforts. Based on the acquired knowledge, nominal/off-nominal physics models and prognostic health management algorithms were developed. In order to aid with development of the algorithms and validate them on realistic data, a testbed capable of supporting experiments in both laboratory and flight environment was developed. Test actuators with architectures similar to potential flight-certified units were obtained for the purposes of testing and realistic fault injection methods were designed. Several hundred fault scenarios were created, using permutations of position and load profiles, as well as fault severity levels. The diagnostic system was tested extensively on these scenarios, with the test results demonstrating high accuracy and low numbers of false positive and false negative diagnoses. The prognostic system was utilized to track fault progression in some of the fault scenarios, predicting the remaining useful life of the actuator. A series of run-to-failure experiments were conducted to validate its performance, with the resulting error in predicting time to failure generally lesser than 10% error. While a more robust validation procedure would require dozens more experiments executed under the same conditions (and, consequently, more test articles destroyed), the current results already demonstrate the potential for predicting fault progression in this type of devices. More prognostic experiments are planned for the next phase of this work, including investigation and comparison of other prognostic algorithms (such as various types of Particle Filter and GPR), addition of new fault types, and execution of prognostic experiments in flight environment.

Acknowledgments

The authors would like to acknowledge contributions of their colleagues and collaborators at NASA Ames Research Center, California Polytechnic State University, Oregon State University, and US Army (Aeroflightdynamics Directorate – AFDD). Michael Koopmans, originally at California Polytechnic State University and now at Oregon State University led a student team to create the original FLEA prototype. Special gratitude goes to AFDD Flight Projects Office Chief Lt Col. Steve Braddom for his unwavering support of this research. The funding for this work was provided by the NASA Aviation Safety Program, IVHM Project and System-Wide Safety and Assurance Technologies (SSAT) Project.

References

[1]Jensen, S.C., G.D. Jenney, and D. Dawson. *Flight Test Experience with an Electromechanical Actuator on the F-18 Systems Research Aircraft*. in *Digital Avionics Systems Conference*. 2000. Philadelphia, PA, USA.

[2]Byington, C.S., et al., *A model-based approach to prognostics and health management for flight control actuators*, in *IEEE Aerospace Conference*. 2005: Big Sky MT.

[3]Mardia, K.V. and R.J. Marshall, *Maximum Likelihood Estimation for Models of Residual Covariance in Spatial Regression.* Biometrika, 1984. **71**(1): p. 135-146.
[4]Brown, D., et al., *Particle Filter Based Anomaly Detection for Aircraft Actuator Systems*, in *IEEE Aerospace Conference*. 2009: Big Sky MT.
[5]Gokdere, L.U., et al., *Lifetime control of electromechanical actuators*, in *Aerospace Conference, 2005 IEEE*. 2005: Big Sky, MT. p. 3523-3531.
[6]Kunst, N. and C. Lynn, *An Innovative Approach to Electromechanical Actuator Emulation and Damage Propagation Analysis*, in *Annual Conference of the Prognostics and Health Management Society*. 2009: San Diego, CA.
[7]Lybeck, N., S. Marble, and B. Morton, *Validating Prognostic Algorithms: A Case Study Using Comprehensive Bearing Fault Data*, in *IEEE Aerospace Conference*. 2007: Big Sky MT.
[8]Bodden, D.S., et al., *Seeded Failure Testing and Analysis of an Electro-Mechanical Actuator*, in *Aerospace Conference, 2007 IEEE*. 2007: Big Sky, MT. p. 1-8.
[9]Smith, M.J., et al., *Experimental and analytical development of health management for Electro-Mechanical Actuators*, in *IEEE Aerospace Conference*. 2009: Big Sky MT.
[10]Balaban, E., et al., *A Diagnostic Approach for Electro-Mechanical Actuators in Aerospace Systems*, in *IEEE Aerospace Conference* 2009: Big Sky MT.
[11]Arvallo, A., *Condition-based Maintanence Of Actuator Systems Using A Model-Based Approach*. 2000, University of Texas, Austin: Austin.
[12]Balaban, E., et al., *Experimental Data Collection and Modeling for Nominal and Fault Conditions on Electro-Mechanical Actuators*, in *Annual Conference of the Prognostics and Health Management Society*. 2009: San Diego, CA.
[13]Balaban, E., et al., *Modeling, Detection, and Disambiguation of Sensor Faults for Aerospace Applications.* IEEE Sensors Journal, 2009. **9**(12): p. 1907-1917.
[14]Gertler, J.J., *Fault Detection and Diagnosis in Engineering Systems*. 1998, New York, NY: Marcel Dekker, Inc.
[15]Karnopp, D.C., D.L. Margolis, and R.C. Rosenberg, *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*. 3rd ed. 2000, New York, NY: John Wiley & Sons, Inc.
[16]Mosterman, P.J. and G. Biswas, *Diagnosis of continuous valued systems in transient operating regions.* IEEE Transactions on Systems, Man and Cybernetics, Part A, 1999. **29**(6): p. 554-565.
[17]Narasimhan, S., P.J. Mosterman, and G. Biswas. *A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems*. in *International Workshop on Diagnosis Principles*. 1998. Cape Cod, MA.
[18]Rasmussen, C.E. and C.K.I. Williams, *Gaussian Processes for Machine Learning*. 2006: The MIT Press.
[19]Saha, S., et al., *Distributed Prognostic Health Management With Gaussian Process Regression*, in *IEEE Aerospace Conference*. 2010: Big Sky MT.