SIL/HIL Replication of Electric Aircraft Powertrain Dynamics and Inner-Loop Control for V&V of System Health Management Routines

Brian Bole[1], Christopher Teubert[2], Cuong Chi Quach[3], Edward Hogge[4], Sixto Vazquez[5], Kai Goebel[6], George Vachtsevanos[7]

[1] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*
bbole3@gatech.edu

[2] *SGT, Inc., NASA Ames Research Center, Moffett Field, CA, 94035*
Christopher.A.Teubert@nasa.gov

[3] *NASA Langley Research Center, Hampton, VA 23681*
cuong.c.quach@nasa.gov

[4] *Northrop Grumman Technical Services, Hampton, VA 23681*
edward.f.hogge@nasa.gov

[5] *NASA Langley Research Center, Hampton, VA 23681*
sixto.l.vazquez@nasa.gov

[6] *NASA Ames Research Center, Moffett Field, CA 94035*
kai.goebel@nasa.gov

[7] *Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332*
george.vachtsevanos@ece.gatech.edu

ABSTRACT

Software-in-the-loop and hardware-in-the-loop testing of failure prognostics and decision making tools for aircraft systems will facilitate more comprehensive and cost-effective testing than what is practical to conduct with flight tests. A framework is described for the offline recreation of dynamic loads on simulated or physical aircraft powertrain components based on a real-time simulation of airframe dynamics running on a flight simulator, an inner-loop flight control policy executed by either an autopilot routine or a human pilot, and a supervisory fault management control policy. The offline testing framework is described for the example of battery charge depletion failure scenarios onboard a prototype electric unmanned aerial vehicle.

1. INTRODUCTION

An early investment of resources into the development of an offline verification and validation (V&V) testing infrastructure for prognostics and supervisory health management algorithms is easily justified for complex systems in which online testing is substantially more time consuming and costly than offline testing. The V&V process is used to confirm that algorithms meet requirements and perform in a way that is consistent with stakeholder expectations. Flight tests prior to algorithm V&V can be dangerous to the vehicle, pilot, and ground crew. Offline tests to V&V algorithms in a laboratory setting prior to flight tests will not only improve flight test safety, but, as many issues can be resolved during offline tests, it reduces the number of real flight tests required for V&V, therefore reducing cost and development time.

Offline V&V tests of supervisory failure prognosis and decision making routines will allow developed supervisory algorithms to interact with onboard flight controllers and measured flight data exactly as they would during flight tests. The offline testing of health management algorithms may be con-

ducted using software-in-the-loop (SIL) or hardware-in-the-loop (HIL) procedures. SIL testing refers to tests conducted using only software simulations of system physics and embedded control routines. HIL testing refers to tests that include some hardware components from the target system.

The SIL/HIL testing framework described in this paper uses the X-Plane[1] flight simulator package and an X-Plane Toolbox for MATLAB to facilitate prognostic based control algorithm V&V over a range of potential operating conductions. Examples of other offline testbeds making use of X-Plane for aerodynamics simulation and Matlab/Simulink softwares for simulation of control routines are found in (Ibeiro & Oliveira, 2010; Brown & Garcia, 2009; Sagoo et al., 2010). The SIL/HIL testbed presented in this paper improves present capabilities for performing offline testing with X-Plane aerodynamics simulations, by including a structure for simulating internal aircraft dynamics and component fault scenarios. The communications architecture developed to interface supervisory control routines running in MATLAB to SIL/HIL tests and an aerodynamics simulation running in X-Plane is intended to be distributed open-source in the near future.

When conducting V&V of supervisory control algorithms, injecting faults and testing to failure can provide valuable knowledge of the algorithm's behavior during potential failure scenarios. It is often not feasible to test to failure during flight tests without compromising the safety of the vehicle, onboard crew (for manned aircraft), or the ground crew. It is therefore valuable to have a method for the offline V&V of algorithm performance during failure scenarios. The offline V&V framework will also facilitate testing over a wide range of potential environmental conditions, including extreme conditions that are rarely encountered in practice. That said however, offline V&V testing is limited by the accuracy of SIL and HIL replications of nominal and off-nominal system dynamics, and flight testing is still a necessary part of the algorithm development and V&V process.

The general framework for SIL/HIL testing is described in Section 2. The development of an SIL/HIL simulation structure for the offline testing of battery charge management algorithms onboard an Edge-540T flight vehicle is presented in Section 3.

2. GENERAL FRAMEWORK

This section introduces an SIL/HIL framework for testing component failure prognostics and real-time supervisory decision making algorithms that are intended to run onboard a flight vehicle. Block diagrams illustrating the structure of control loops used for online and offline flight testing are shown in Figure 1. The symbols used in Figure 1 and elsewhere in the paper are defined in the Nomenclature table at the end of the paper. An inner-loop controller is assumed to

(a) Inner and outer control loops for online testing

(b) Inner and outer control loops for offline testing
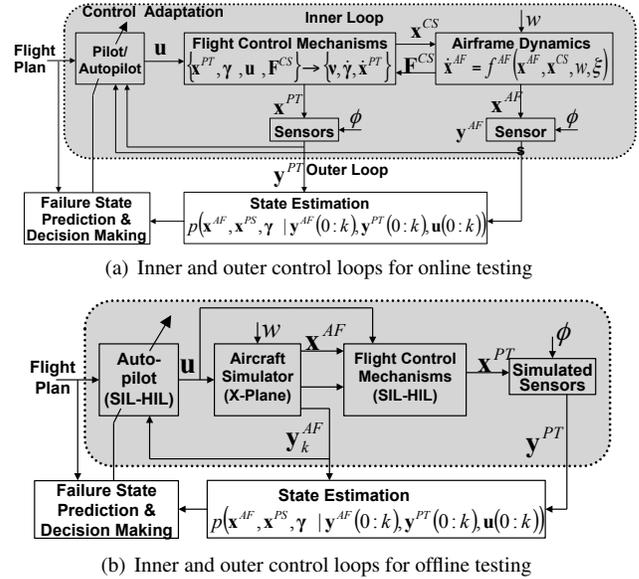
Figure 1. Inner and outer control loops for online and offline controls testing

update flight control inputs based on a known flight plan and observations of the system state. The inner-loop controller is denoted by the 'Pilot/Autopilot' blocks in Figures 1(a) and 1(b). Failure prognostics and supervisory decision making operations are performed by an outer-loop process, denoted by the 'State Estimation', 'Failure State Prediction', and 'Decision Making' blocks in the figure. Both the inner-loop and outer-loop controllers would be unchanged in either online or offline flight testing.

2.1. Inner-Loop Control Dynamics

Inner-loop control of the aircraft is assumed to be provided by either a human pilot or a pre-programmed autopilot. Both human pilot and autopilot will henceforth be referred to as just 'the pilot', for convenience. The pilot will update the control vector, u, based on the observed states of the aircraft and a desired system state directed by a given flight plan.

The 'Flight Control Mechanisms' block shown in Figures 1(a) and 1(b) represents the internal electrical and mechanical dynamics of the vehicle's powertrain. The inner-loop control signals sent by the pilot, and the aerodynamic forces exerted on the vehicle's control surfaces by the surrounding environment, $\mathbf{F}^{CS}$, are inputs to this block. The $\mathbf{F}^{CS}$ vector consists of forces like the drag on the aircraft propeller, or the torque on control flaps. These inputs result in the loading of powertrian components, represented by the vector $\boldsymbol{\nu}$, which in turn determine the dynamics of powertrain component states, $\dot{x}^{PT}$, and the dynamics of potential fault modes. The magnitudes of potential fault modes are represented here by the fault mode vector, $\boldsymbol{\gamma}$, where fault modes are assumed

2

to be measurable indicators of component degradation such as crack length, spall width, or pitting depth.

The loads exerted by the vehicle's active components at a given time index, k, are expressed as a function of control input signals, the current states of powertrain components, and the states of component fault modes that may reduce component effectiveness,

$$\boldsymbol{\nu}\left(k\right) = f^{PT}\left(\boldsymbol{u}\left(k\right), \boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\gamma}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \qquad (1)$$

where the additional $\boldsymbol{\xi}$ term is used to represent a vector of unknown or uncertain model parameters.

The 'Airframe Dynamics' block shown in Figure 1(a) represents the aerodynamic interactions between the vehicle airframe, vehicle control surfaces, and the operating environment. The inputs to this block are the current states of aircraft control surfaces, $\boldsymbol{x}^{CS}$, and the current state of the operating environment, $\boldsymbol{w}$. Environmental states represented by $\boldsymbol{w}$ may include atmospheric pressure, air temperature, wind speed, and turbulence. The current state of the airframe is represented by, $\boldsymbol{x}^{AF}$; it includes the position, heading, linear and rotational speed, and linear and rotational acceleration of the airframe in a set coordinate system.

Vehicle control surfaces are mechanically connected to powertrain components, so they should be a known function of $\boldsymbol{x}^{PT}$,

$$x^{CS} = f^{CS}\left(x^{PT}\right) \qquad (2)$$

The forces exerted on the vehicle's control surfaces due to their motion through surrounding air is represented here as a generic non-linear function of the airframe state, the states of vehicle control surfaces, and current environmental states,

$$\boldsymbol{F}^{CS}\left(k\right) = f^{F^{CS}}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{x}^{CS}\left(k\right), \boldsymbol{w}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \quad (3)$$

where the additional $\boldsymbol{\xi}$ term is used to represent a vector of unknown or uncertain model parameters.

Powertrain state dynamics and airframe dynamics are generically expressed in terms of the loading vectors $\boldsymbol{\nu}$ and $\boldsymbol{F}^{CS}$ as:

$$\dot{\boldsymbol{x}}^{PT}\left(k\right) = f^{PT}\left(\boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\nu}\left(k\right), \boldsymbol{F}^{CS}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \quad (4)$$

$$\boldsymbol{y}^{PT}\left(k\right) = h^{PT}\left(\boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\phi}\left(k\right)\right) \qquad (5)$$

$$\dot{\boldsymbol{x}}^{AF}\left(k\right) = f^{AF}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{x}^{CS}\left(k\right), \boldsymbol{w}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \quad (6)$$

$$\boldsymbol{y}^{AF}\left(k\right) = h^{AF}\left(\boldsymbol{x}^{AF}\left(k\right), \boldsymbol{\phi}\left(k\right)\right) \qquad (7)$$

The progression of component health degradation is represented as:

$$\dot{\boldsymbol{\gamma}}\left(k\right) = f^{\boldsymbol{\gamma}}\left(\boldsymbol{x}^{PT}\left(k\right), \boldsymbol{\gamma}\left(k\right), \boldsymbol{\nu}\left(k\right), \boldsymbol{\xi}\left(k\right)\right) \qquad (8)$$

where component failure is considered to occur when fault magnitudes exceed a defined threshold that renders the component ineffective. The deterioration of control surfaces and electromechanical components in aircraft powertrains as a function mechanical loading forces has been a topic of study for some time; examples include: electromechanical actuators (Balaban et al., 2010) and composite wing structures (Gobbato et al., 2012), to name a few. The degradation and failure of electrical components as a function of electrical power loading has also been examined for aircraft components such as batteries (Saha et al., 2009) and power electronics (Celaya et al., 2011).

Measurements from simulated vehicle powertrain components, $\boldsymbol{y}^{PT}$, are generated in offline testing using models for both the underlying component dynamics, and sensor response dynamics. Data from actual system hardware is obtained in offline testing by applying mechanical loads to hardware components in real-time, in accordance with the environmental loads, $\mathbf{F}^{CS}$, reported in by the X-Plane simulator.

A hardware-only recreation of the 'Flight Control Mechanisms' portion of the inner-loop vehicle dynamics, illustrated in Figure 1(b), could be accomplished in a laboratory setting using an aircraft battery pack, power electronic motor/actuator drivers, electromechanical components, and associated interconnection cabling. Pilot controls could be sent directly to an electrical power distribution system assembled in the laboratory, and additional loading hardware could be used to apply mechanical loads to the electromechanical components of the powertrain in order to recreate the environmental loads estimated by the aircraft simulator. This approach is similar in nature to dynamometer testing commonly performed in the testing of automotive systems (Kelly et al., 2002; Tsang et al., 1985). Software models may be switched in for some or all of the hardware components in this setup; however, small errors in modeling the behavior of a given component may have outsized effects in observed system behavior over long time periods.

Measurements of the airframe states are represented by the vector, $\boldsymbol{y}^{AF}$. The X-Plane simulator, used in offline testing, includes sensor models to generate $\boldsymbol{y}^{AF}$, with the possibility of injecting sensor error and sensor noise. Communication between an autopilot board and the framework is facilitated by the open-source program APM Mission Planner[2]. The X-Plane Toolbox for MATLAB was used to communicate with APM Mission Planner, X-Plane, and outer-loop supervisory control routines running in MATLAB. The toolbox, currently being developed at NASA Ames Research Center, provides various MATLAB functions that allow for UDP communication with an associated X-Plane plug-in and APM Mission Planner. The team developing the toolbox intends to release it open-source upon completion.

Adequate control of aircraft does not in most cases require a pilot to understand environmental dynamics or the inter-

---

[2]http://code.google.com/p/ardupilot-mega/wiki/Mission

nal dynamics of the flight vehicle in great detail. Pilots are autopilots are considered here to make decisions based on an internal decision making policy that maps observations of $y^{AF}(k)$ and $y^{PT}(k)$ at time-index k to appropriate control outputs, $u(k)$. An autopilot will use an embedded control policy to map $(y^{AF}(k), y^{PT}(k)) \to u$. For human pilots, the mapping $(y^{AF}(k), y^{PT}(k)) \to u$ will be determined by the pilot's situational awareness and judgment. The mechanism for interaction between an autopilot and supervisory failure prognostics and decision making routines can be for the decision making routines to directly update the autopilot's control policy. Policy updates for human pilots could be prompted indirectly by presenting the pilot with system health information and suggested risk mitigating actions, as described in (Bukov, Chernyshov, Kirk, & Schagaev, 2007).

2.2. Outer-loop Failure Prognostics and Decision Making

Supervisory outer-loop control routines make use of sensor measurements to estimate current and future system states given approximations of system state dynamics and physics of failure models. Probability distributions for belief in the current states of $\mathbf{x}^{PT}$, $\mathbf{x}^{AF}$, and γ, based on a history of observations of $\mathbf{y}^{PT}$, $\mathbf{y}^{AF}$, and $\mathbf{u}^{AF}$ are generically represented in Figure 1 as:

$$p\left(\mathbf{x}^{PT}, \mathbf{x}^{AF}\gamma | \mathbf{y}^{PT}(0:k), \mathbf{y}^{AF}(0:k), \mathbf{u}(0:k)\right) \quad (9)$$

Many Bayesian and machine learning methods have been published for the estimation of such probability distributions in the aviation domain (Lopez & Sarigul-Klijn, 2010; Napolitano et al., 1998).

Prediction of the evolution of future system states may be performed by propagating input uncertainty, model uncertainty, and state uncertainty forward in time. Estimates of component remaining useful life (RUL) are generated by extending predictions of the evolution of component loads and corresponding predictions of fault state evolution into the future until there is sufficient confidence in the occurrence of either component failure or completion of a prescribed flight plan. Particle filtering (Arulampalam et al., 2002), extended Kalman filtering (Ray & Tangirala, 1996), and Markov modeling (Guidaa & Pulcini, 2011) are examples of predictive filtering techniques used to propagate current state and model uncertainties forward in time.

Because inner-loop control policies may be modified by outer-loop supervisory control actions, the outer-loop prognostics and decision making routines could also be factored into the computation of future component load estimates. (Bole et al., 2012) describes the incorporation of outer-loop control policies into inner-loop fault growth predictions. The role of stochastic estimates of future loading in prognostic predictions is described in (Sankararaman et al., 2013; Tang et al., 2009). Stochastic beliefs about the manner in which

Figure 2. Edge-540T on runway

the environment or system will evolve over time may be validated in offline simulations against repeated randomized simulations of flight scenarios.

3. A CASE STUDY: UAV BATTERY CHARGE DEPLETION MODELING

The implementation of the proposed SIL/HIL framework for the offline simulation of battery charge depletion onboard a prototype electric UAV platform is discussed here. The aircraft platform used for this case study is a commercial-off-the-shelf (COTS) 33% scale model of the Zivko Edge 540T airplane, pictured in Figure 2. The wingspan of the Edge 540T is 87 inches. The mass of the instrumented vehicle is approximately 44lbs. The propeller of the UAV is driven by two tandem mounted outrunner brushless DC motors that are each powered by a series connection of two lithium polymer battery packs. Each of the battery packs consist of five series connections of two 4.2V 3900mAh lithium polymer pouch cells wired in parallel.

The electrical and mechanical connections in the UAV powertrain are illustrated in Figure 3. Power flow from the battery packs to the driving motors is controlled by a Jeti 90 Pro Opto electric speed controller (ESC). The ESC sends synchronized voltages to the propeller motors at a duty cycle determined by a throttle input. The throttle input is either sent by remote control from a pilot, or by an onboard autopilot.

During both remote control and autonomous flight, a human pilot will maintain line of sight with the aircraft, and stand ready to execute a landing maneuver when the command is given by other operators on the ground. The gound operators assissting the pilot monitor the battery end-of-discharge prognostic estimates and decision making outputs generated in real-time by outer loop supervisory routines.

Charge estimation and end of charge prediction for UAV powertrain batteries have previously been examined in several publications by Bhaskar Saha at NASA ARC, Quach Chong Chi at NASA LaRC, and others (Saha, Quach, & Goebel, 2011; Saha, Koshimoto, et al., 2011). A separate battery system is used to power the data acquisition and other flight com-
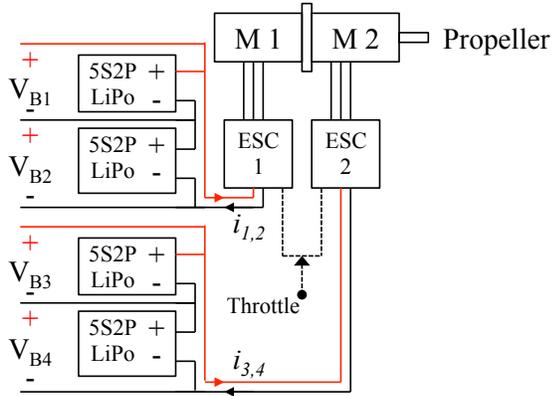
Figure 3. Electrical and mechanical connections of an Edge-540T UAV powertrain

munications and control hardware. The two battery systems are sized such that it is very likely that the batteries powering the propeller motors will be the first to be depleted. For that reason, onboard battery discharge prognostic algorithms and supervisory decision making actions are considered to only be concerned with the propeller driving batteries.

The offline replication of Edge 540T powertrain dynamics is described in three parts: inner-loop controls, battery demand modeling, and equivalent circuit battery modeling.

3.1. The Inner-Loop Controller

Vehicle flight plans are considered to be given in terms of an ordered set of 3D coordinates to be visited by the UAV, and a desired airspeed for making the translation from one way-point to the next. Autonomous control of the Edge 540T is performed using an ArduPilot board. The ArduPilot sends control commands to the aircraft ESCs and flight control surfaces based on a set of proportional integral derivative (PID) control parameters that are tuned prior to flight, and periodic measurements of vehicle airspeed, heading, and position.

As was described in Section 2.1, the X-Plane simulator is used to simulate vehicle aerodynamics in offline simulations. Plane Maker, a design tool within the X-Plane package, was used to specify the aircraft mass, balance, and geometry for use in X-Plane aerodynamic simulations. There is some unavoidable error between the actual geometry, drag, and mass distribution of the aircraft and that used in the X-Plane aerodynamics models; however, because the control system is closed-loop small errors in simulating aircraft aerodynamics will not typically accumulate into large errors.

X-Plane can simulate various weather conditions and hardware configurations, and the ArduPilot can be tested with various flight plans. This configuration allows for the thorough testing of algorithm performance and safety before conducting flight tests.

3.2. Battery Demand Modeling

The proposed SIL/HIL testbed separates the simulation of aerodynamics and powertrain dynamics into two functional blocks. Connecting these two blocks requires that the airframe loads reported by the aerodynamics simulation be translated into loads on the system's powertrain components. It is difficult to collect direct measurements of airframe loads such as component forces and torques in flight. This measurement difficulty makes validating the load mapping used in offline simulations a complex proposition.

The tuning and validation of a propeller load mapping function is separated into two steps in this paper. First, a series of characterization experiments are performed in X-Plane to identify a nonlinear mapping between propeller output power and aircraft angle of climb, speed, and acceleration. Second, the modeled propeller power is mapped to a required battery power using a fixed power conversion efficiency coefficient and a proportional drag correction coefficient.

The nonlinear relationship between propeller output power and aircraft angel of climb, speed, and acceleration is observed for a flight simulator representation of an aircraft, by simulating a series of climbing and descending maneuvers at various angle of climb and throttle setpoints. Unlike actual flight tests, there is no difficulty in observing the precise loads on aircraft components in simulated flight tests. The results of repeated experiments at different throttle and angel of climb setpoints are used to fit a general set of aircraft aerodynamics and energy conservation equations, presented below.

The equations below are developed using the following assumptions: 1) the propeller is mounted on the aircraft nose; 2) the angle between the thrust vector generated by the propeller and the velocity vector of the aircraft is small; 3) Aircraft turning forces are small in comparison to the thrust and drag forces on the aircraft in its direction of travel.

The sum of the forces acting in the aircraft direction of travel is:

$$T_{x_w} = D(v) + m \cdot g \cdot \sin(\alpha) + m \cdot \dot{v} \qquad (10)$$

where T_{x_w} represents the thrust produced by the aircraft in the direction of travel, D represents the drag force acting in the opposite direction of aircraft motion, v represents the aircraft speed, $\dot{v}$ represents acceleration, α represents angle of climb, m represents the vehicle mass, and g represents the earth's gravity.

The drag force on the airframe is represented by the following polynomial function of airspeed and angle of climb.

$$D(v, \gamma) = c_1 + c_2 \cdot v + c_3 \cdot v^2 + c_4 \cdot \alpha \qquad (11)$$

Figure 4(a) shows a fit of the drag model to the averaged drag force reported by the X-Plane simulator over several steady speed climbing and descending maneuvers. The fitted param-

eter values are: $c_1 = 13.47, c_2 = -0.6, c_3 = 0.019, c_4 = 0.14$. During take-off and landing maneuvers when the aircraft speed is less than $15m/s$ the drag force is approximated as $D = 3 \cdot v$.

The measured and estimated propeller thrust versus airspeed is shown in Figure 4(b). The plot was generated using the thrust model given in Eq. (10), and the drag model given in Eq. (11).

The aircraft ESC throttle command is fit as a linear function of thrust and airspeed,

$$\text{Throttle} = a_1 + a_2 \cdot T_{x_w}(v, \dot{v}, \gamma) + a_3 \cdot v \qquad (12)$$

where the fitted parameter values are: $a_1 = -19.64$, $a_2 = 0.95$, $a_3 = 1$.

Figure 4(c) shows the measured and estimated steady state ESC throttle commands observed in X-Plane simulations of various steady state climbing and descending maneuvers.

The product of thrust and airspeed gives the motive power exerted by the aircraft,

$$P_p = \frac{1}{\eta_p} \cdot T_{x_w} \cdot v \qquad (13)$$

where P_p represents propeller output power and η_p represents the approximate propeller output power conversion efficiency.

Figure 4(d) shows the modeled propeller power and that reported by the X-Plane simulator for several steady speed climbing and descending maneuvers. The η_p parameter for the modeled aircraft is fit to $\eta_p = 0.7652$.

A fixed battery power conversion efficiency is assumed here for the aircraft motors and power electronics. Conversion between the net propeller output power and the net battery output power required to maintain a particular airspeed and angle of climb setpoint is achieved by applying a fixed power factor,

$$P_b = \eta_e \cdot P_p \qquad (14)$$

where η_e represents power conversion efficiency and P_b represents net battery output power.

A proportional factor is also introduced here to correct for discrepancies between the drag model given in Eqn. 11, and the drag forces estimated for the actual aircraft. The corrective factor is expressed as:

$$D_A(v, \gamma) = \lambda_D \cdot D_M(v, \alpha) \qquad (15)$$

where D_A and D_M represent the drag force estimated for the actual aircraft and drag force estimated for the X-Plane model respectively. λ_D represents a constant corrective factor that may be fitted by comparing modeled and actual aircraft powertrain load dynamics over sample flights.
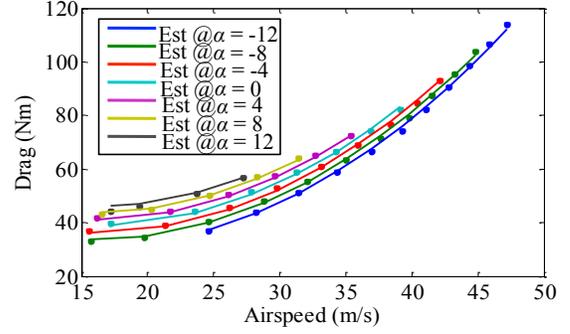
(a) Estimated and measured drag vs airspeed at various angles of climb
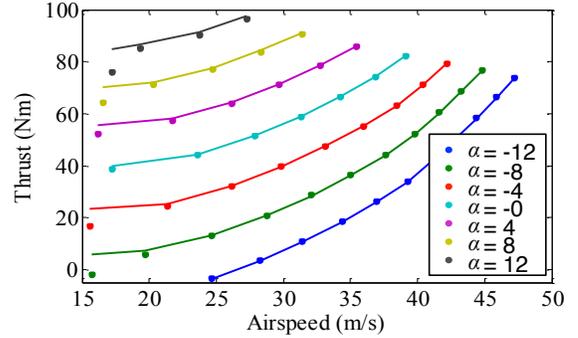
(b) Estimated and measured thrust vs airspeed at various angles of climb
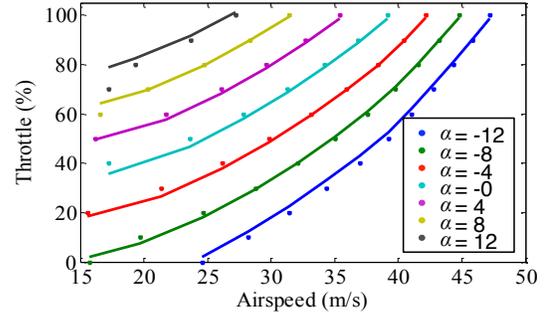
(c) Estimated and measured throttle vs. airspeed at various angles of climb
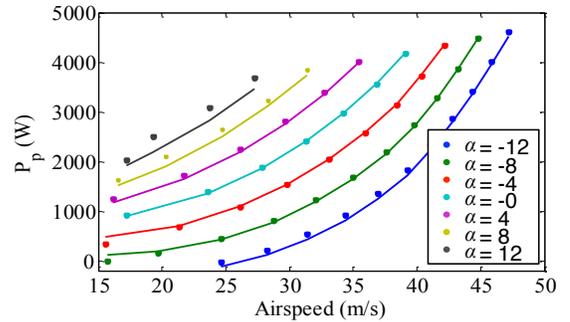
(d) Estimated and measured propeller power output vs airspeed at various angles of climb

Figure 4. Model fitting results for X-Plane flight load characterization tests
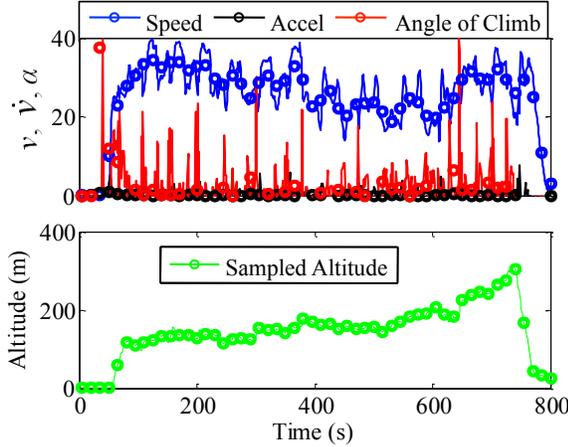
Figure 5. Approximate aircraft airspeed, acceleration, and angle of climb measurements derived from GPS samples

A roughly proportional deviation between the modeled and actual drag force is attributed to slight errors in modeling the aircraft geometry and surface aberrations. Small errors in modeling aircraft drag will cause only small effects on the aircraft handling from the perspective of a pilot or an autopilot, thus the drag correction need not necessarily be made for the SIL testing of inner-loop controllers. However, small errors in approximating loads on onboard energy storage devices will accumulate into large errors over a simulated flight.

Substitution of Eqns. (14) and (15) into Eqns. (10)-(13) yield the approximate battery power required to fly at a particular airspeed and angle of climb.

$$
\begin{aligned}
P_B &= \frac{1}{\eta_e \eta_p} \cdot T_{x_w} \cdot v \\
P_B &= \frac{v}{\eta_e \eta_p} \cdot (D_A(v, \alpha) + mg \cdot \sin(\alpha) + m\dot{v}) \\
P_B &= \frac{v}{\eta_e \eta_p} \cdot (\lambda_D D_M(v, \alpha) + mg \cdot \sin(\alpha) + m\dot{v})
\end{aligned}
\tag{16}
$$

Figure 5 shows approximate aircraft airspeed, acceleration, and angle of climb measurements derived from GPS samples, taken at 1 second intervals, over a sample aircraft flight. Figure 6 shows the predicted and measured battery power draw over the recorded aircraft flight, using the average airspeed, acceleration, and angle of climb values falling within fifteen second intervals over the flight. The proportional constants η_e and λ_D used in Eq. 16 were fitted to $\eta_e = 0.85$ and $\lambda_D = 0.9$ to obtain the model fit shown in Figure 6.

The battery power predictions shown in Figure 6 are seen to lay on top of the observed power draw over the sample flight, aside from an apparent under prediction of battery power required during takeoff, which occurs from about 50 seconds to 100 seconds in the plots given. The under prediction of power required during takeoff could arise in part from the assumption that the angle between the trust vector and the velocity vector is small, which is not necessarily the case during takeoff. The battery power demand modeling used here also
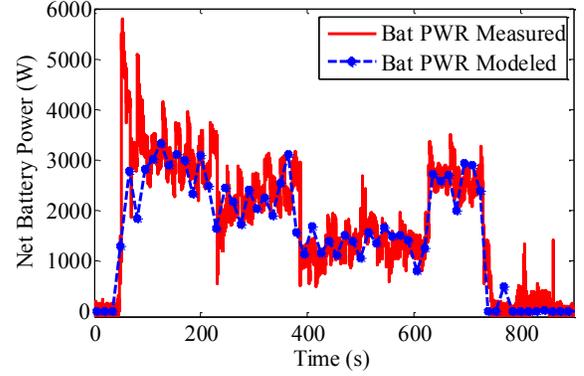
Figure 6. Measured and modeled battery power output

does not account for the fact that motor power conversion efficiency is typically very low during initial spin up.

The net battery power, P_B, input to aircraft ESCs is given by the summation of battery power input to ESC1 and ESC2. The proportion of the net battery output powers that goes to each ESC is represented by:

$$
\lambda_{ESC} = \frac{P_1 + P_2}{P_3 + P_4}
\tag{17}
$$

where λ_{ESC} represents the ratio of battery power drawn by each of the onboard ESCs.

The power output from the two strings of series connected battery packs is equal to the product of current and voltage,

$$
P_B = I_{1,2} \cdot (V_{B1} + V_{B2}) + I_{3,4} \cdot (V_{B3} + V_{B4})
\tag{18}
$$

where V_{Bi} represents the terminal voltage of battery i, $I_{1,2}$ and $I_{3,4}$ represents the current flowing through the two sets of series connected batteries.

Substitution of Eq. (18) into Eq. (17), and solving for I gives:

$$
\begin{aligned}
I_{1,2} &= \frac{\lambda_{ESC} P_b}{(\lambda_{ESC}+1) \cdot (V_{B1}+V_{B2})} \\
I_{3,4} &= \frac{P_b}{(\lambda_{ESC}+1) \cdot (V_{B3}+V_{B4})}
\end{aligned}
\tag{19}
$$

Figure 7 shows the observed ratio of battery power drawn from each of the onboard ESCs over a sample flight. The ratio of ESC power draw is currently uncontrolled, and it is seen to drift around a value of $\lambda_{ESC} \approx 0.7$ over the sample flight. The approximation for λ_{ESC} to be used in SIL and HIL testing of the vehicle powertrain may be improved in future work by incorporating possible dependencies on time, battery pack voltage, throttle command, and other inputs control inputs.

3.3. An Equivalent Circuit Battery Model

Battery voltage-current dynamics may be recreated over simulated flights in a laboratory by loading real or simulated batteries with a current indicative of flight loads. It should be
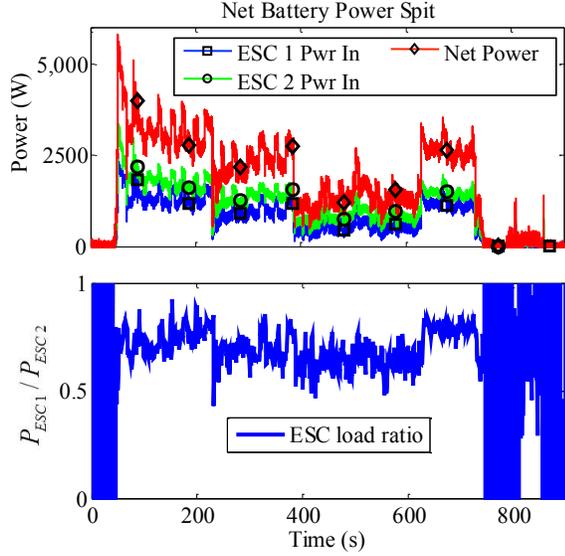
Figure 7. Measured battery power input to ESCs (Top) and observed ESC power ratio over a sample flight (bottom)

noted however that battery dynamics will vary substantially as a function battery health and temperature (Jossen, 2006). Differences in state of health and thermal loading of real and simulated batteries may cause results from SIL/HIL cycling of batteries in a laboratory to diverge from the observed battery dynamics in flight test.

Aircraft powertrain batteries are simulated in SIL testing using the equivalent circuit model shown in Figure 8. The equivalent circuit model used here is an extended version of the model explained in (Daigle, Saxena, & Goebel, 2012). This battery model uses six electrical components that are tuned to recreate the observed current-voltage dynamics of Edge 540T powertrain batteries. Battery charge is stored in the capacitor, C_b. The R_s, C_s and R_{cp}, C_{cp} circuit element pairs capture battery internal resistance drops and concentration polarization effects, respectively. The resistor R_p accounts for the slow battery self-discharge that is seen to occur over weeks or months of storage.

The current and voltage dynamics of the equivalent circuit model are defined as:

$$x^B = \begin{bmatrix} q_b & q_{cp} & q_{Cs} \end{bmatrix}^T \tag{20}$$

$$\dot{x}^B = \begin{bmatrix} -\frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & -\frac{1}{C_{cp} R_p R_{cp}} & \frac{1}{C_s R_p} \\ \frac{1}{C_b R_p} & \frac{1}{C_{cp} R_p} & \frac{1}{C_s R_p} \end{bmatrix} x + \begin{bmatrix} i \\ i \\ i \end{bmatrix} \tag{21}$$

$$y^B = V_p = \begin{bmatrix} \frac{1}{C_b} & \frac{1}{C_{cp}} & \frac{1}{C_s} \end{bmatrix} \cdot x \tag{22}$$

where q_b, q_{cp}, and q_{cs} represent the charge stored in capacitors C_b, C_{cp}, and C_{cs} respectively. The total voltage drop across the battery terminals, V_p, is given by the sum of the
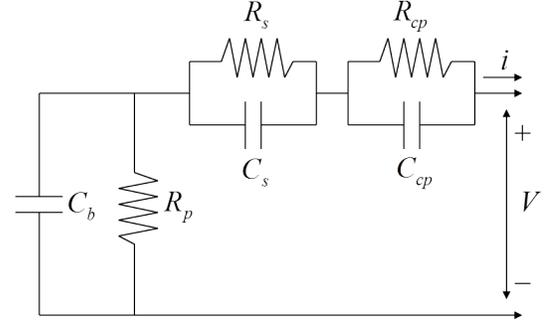
Figure 8. Equivalent circuit battery model

voltage drops across the each of the three capacitors in the equivalent circuit model.

Because battery current-voltage dynamics are known to vary as a function of battery SOC, some of the resistive and capacitive (RC) components in the equivalent circuit model must be parameterized as functions of battery state of charge (SOC) (Zhang & Chow, 2010). It was decided based on qualitative observation that defining C_b, C_{cp}, and R_{cp} as parameterized functions of battery SOC gave an acceptable trade-off between the number of parameters to be identified and model error.

Battery SOC is defined as:

$$\text{SOC} = 1 - \frac{q_{max} - q_b}{C_{max}} \tag{23}$$

where q_b is the charge stored in the battery, q_{max} is the maximum charge of the battery, and C_{max} is the maximum charge that can be drawn from the battery. The term coulombic efficiency is used to refer to the portion of stored charge that can be withdrawn over repeated charge and discharge cycling of a battery. Resting a battery can temporarily unlock some of its lost charge storage capacity, however the overall trend is inevitably downward.

The C_b, C_{cp} and R_{cp} terms in the equivalent circuit battery model are parameterized as:

$$C_b = C_{Cb0} + C_{Cb1} \cdot \text{SOC} + C_{Cb2} \cdot \text{SOC}^2 + C_{Cb3} \cdot \text{SOC}^3 \tag{24}$$

$$C_{cp} = C_{cp0} + C_{cp1} \cdot \exp\left(C_{cp2}\left(1 - \text{SOC}\right)\right) \tag{25}$$

$$R_{cp} = R_{cp0} + R_{cp1} \cdot \exp\left(R_{cp2}\left(1 - \text{SOC}\right)\right) \tag{26}$$

Each battery pack used in Edge 540T flight tests should be characterized individually prior to testing, in order to account for any manufacturing and SOH variations. Two battery characterization experiments are used to identify the RC parameters in the battery equivalent circuit model.

The first experiment is a low current discharge of a battery from a fully charged state until a cutoff voltage of 17.5V is
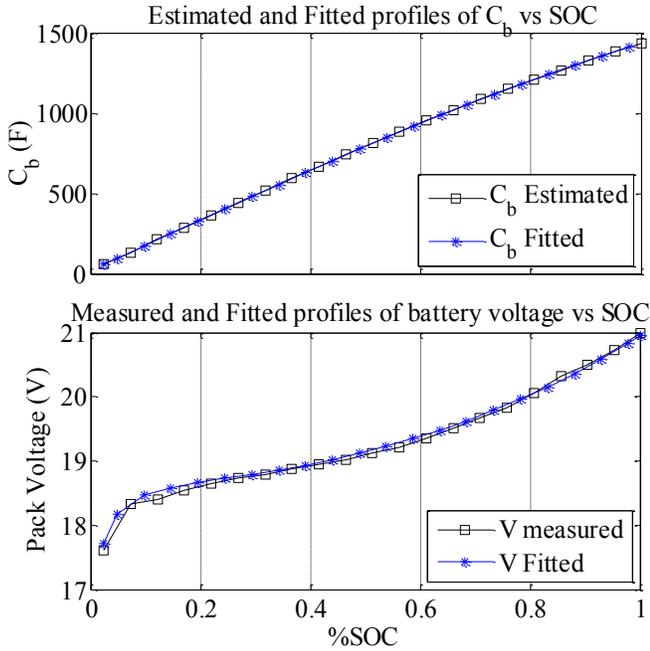
Figure 9. Measured and fitted profiles for C_b and battery voltage

reached. This type of discharge is mostly affected by the C_b, q_b, q_{max}, and C_{max} parameters in the model. Figure 9 shows a polynomial fit of C_b as a function of SOC, and the battery voltage fit for the tuned parameter values for C_{Cb0}, C_{Cb1}, C_{Cb2}, C_{Cb3}, q_{max}, and C_{max}.

Next, a pulsed loading experiment is used to fit the remaining parameters in the equivalent circuit model to the observed changes in battery hysteresis behavior as a function of SOC. A gradient descent search is used to identify the remaining model coefficients using the pulsed loading data. Figure 10 shows the battery voltage fit over a pulsed loading profile, using the tuned parameters identified in the low current experiment and the newly tuned values of R_s, C_s, R_{cp0}, R_{cp1}, R_{cp2}, C_{cp0}, C_{cp1}, and C_{cp2}. Values for all of the RC components and parameterization coefficients used in the equivalent circuit model of an Edge battery are defined in Table 1.

Observed battery loading over a piloted flight of the Edge 540T is shown in Figure 11. An asymmetric loading of the two propeller motors over the sample flight is apparent from the battery loads given in Figure 11. Motor M2 is known to consistently draw more current than motor M1 on the Edge 540, due to unregulated coupling of the two motor ESCs. Predicted and measured voltage profiles for batteries B1 and B3 using the recorded battery current profiles are shown in Figure 11. The close match between observed battery voltages and open-loop predictions over a given loading profile provides a measure of the validity of the software model.

The tuned battery model may be used to estimate the internal
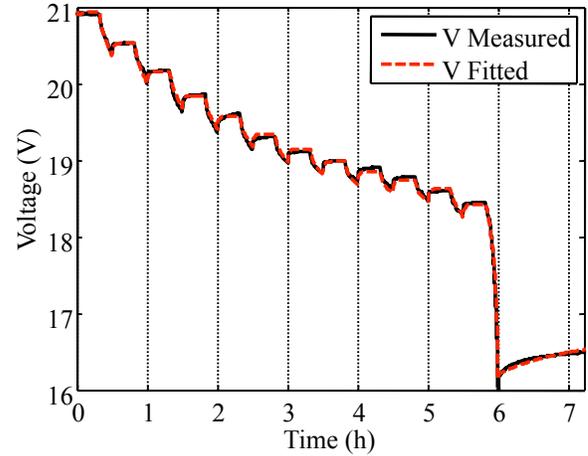
Figure 10. Measured and fitted profiles for battery voltage during pulsed loading
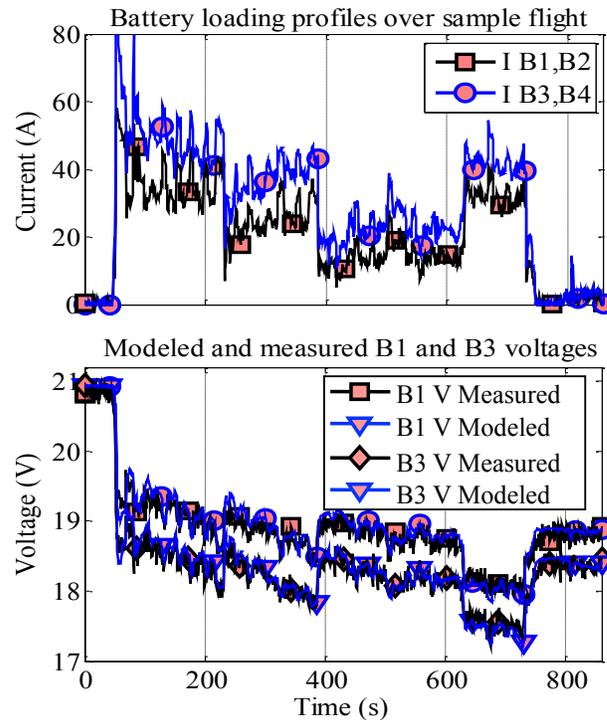
Figure 11. Modeled and measured voltages of batteries B1 and B3 for a sample flight loading profile

SOC of powertrain batteries based on sampled voltage and current data. The output of model based filtering approaches such as Kalman filtering will be much less susceptible to initialization and measurement errors than the Coulomb counting method currently used in many battery monitoring systems (Dai, Wei, & Sun, 2006).

9

Table 1. Parameter values used in equivalent circuit model

Parameter	Value	Parameter	Value
q_{max}	2.88×10^4 C	C_s	89.3 F
C_{max}	2.85×10^4 C	R_{cp0}	1.60×10^{-3} Ω
C_{Cb0}	19.4 F	R_{cp1}	8.45
C_{Cb1}	1576 F	R_{cp2}	-61.9
C_{Cb2}	41.7 F	C_{cp0}	2689 F
C_{Cb3}	-203 F	C_{cp1}	-2285 F
R_s	2.77×10^{-2}	C_{cp2}	-0.73 F

4. CONCLUSIONS

A framework is described for the offline recreation of dynamic loads on simulated or physical aircraft powertrain components based on a real-time simulation of airframe dynamics, an inner-loop flight control policy executed by either an autopilot routine or a human pilot, and a supervisory outer-loop control policy. The creation of an offline framework for verifying and validating supervisory outer-loop prognostics and decision making routines is described for the example of battery charge depletion failure scenarios onboard a prototype Edge 540T UAV with electric propulsion. The SIL/HIL testbed described in this paper is intended to be used to perform much more comprehensive and cost-effective testing of aircraft fault prognostics and decision making tools than would be practical to conduct in flight testing.

ACKNOWLEDGMENT

NOMENCLATURE

$\boldsymbol{x}^{AF}$	airframe state vector
$\boldsymbol{y}^{AF}$	observation of airframe state vector
$\boldsymbol{x}^{PT}$	electrical power dist. system state vector
$\boldsymbol{y}^{PT}$	observation of $\boldsymbol{x}^{PT}$ states
$\boldsymbol{u}$	pilot or autopilot control output vector
$\boldsymbol{\nu}$	mechanical loads on electromechanical components
$\boldsymbol{F}^{CS}$	net mechanical loads exerted by airframe
$\boldsymbol{w}$	environmental state parameter vector
γ	magnutude state vector for potential faults modes
ξ	captures uncertainties in physics of failure models
ϕ	captures noise in sensor measurements

REFERENCES

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174-188.

Balaban, E., Saxena, A., Narasimhan, S., Roychoudhury, I., Goebel, K., & Koopmans, M. (2010). Airborne electro-mechanical actuator test stand for development of prognostic health management systems. In *Annual conference of the prognostics and health management society.*

Bole, B., Goebel, K., & Vachtsevanos, G. (2012). Stochastic modeling of component fault growth over a derived domain of fiesible output control effort modifications. In *Annual conference of the prognostics and health management society.*

Brown, A., & Garcia, R. (2009). Concepts and validation of a small-scale rotorcraft proportional integral derivative (pid) controller in a unique simulation environment. *Unmanned Aircraft Systems*, *1*, 511-532.

Bukov, V., Chernyshov, V., Kirk, B., & Schagaev, I. (2007). Principle of active system safety for aviation: Challenges, supportive theory, implementation, application and future. In *ASTEC.*

Celaya, J., Kulkarni, C., Biswas, G., & Goebel, K. (2011). A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging. In *Proceedings of the annual conference of the PHM society.*

Dai, H., Wei, X., & Sun, Z. (2006). Online SOC estimation of high-power lithium-ion batteries used on HEVs. In *IEEE international conference on vehicular electronics and safety.*

Daigle, M., Saxena, A., & Goebel, K. (2012). An efficient deterministic approach to model-based prediction uncertainty. In *Annual conference of the prognostics and health management society.*

Gobbato, M., Conte, J., Kosmatka, J., & Farrar, C. (2012). A reliability-based framework for fatigue damage prognosis of composite aircraft structures. *Probabilistic Engineering Mechanics*, *29*, 176-188.

Guidaa, M., & Pulcini, G. (2011). A continuous-state Markov model for age- and state-dependent degradation processes. *Structural Safety*, *33*(6), 354-366.

Ibeiro, L., & Oliveira, N. M. (2010). UAV autopilot controllers test platform using MATLAB/Simulink and X-Plane. In *Frontiers in education conference (FIE).*

Jossen, A. (2006). Fundamentals of battery dynamics. *Journal of Power Sources*, *154*, 530-538.

Kelly, K., Mihalic, M., & Zolot, M. (2002). Battery usage and thermal performance of the toyota prius and honda insight during chassis dynamometer testing. In *Seventeenth annual IEEE battery conference on applications and advances.*

Lopez, I., & Sarigul-Klijn, N. (2010). A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: Challenges and opportunities. *Progress in Aerospace Sciences*, *46*, 247-273.

Napolitano, M., Windon, D., Casanova, J., Innocenti, M., & Silvestri, G. (1998). Kalman filters and neural-network schemes for sensor validation in flight control systems. *IEEE Transactions on Control Systems Technologyy.*

Ray, A., & Tangirala, S. (1996). Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, *4*, 443-451.

Sagoo, G., Gururajan, S., Seanor, B., Napolitano, M., Perhinschi, M., Gu, Y., & Campa, G. (2010). Evaluation of a fault-tolerant scheme in a six-degree-of-freedom motion flight simulator. *Journal of Aerospace Computing, Information, and Communication*, *7*, 47-67.

Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009). Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Transactions on Instrumentation and Measurement*, *58*(2), 291-296.

Saha, B., Koshimoto, E., Quach, C., Hogge, E., Strom, T., Hil, B., . . . Goebel, K. (2011). Battery health management system for electric UAVs. In *IEEE aerospace conference*.

Saha, B., Quach, P., & Goebel, K. (2011). Exploring the model design space for battery health management. In *Annual conference of the prognostics and health managment society*.

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *IEEE aerospace conference*.

Tang, L., Kacprzynski, G., Goebel, K., & Vachtsevanos, G. (2009). Methodologies for uncertainty management in prognostics. In *IEEE aerospace conference*.

Tsang, P., Jacko, M., & Rhee, S. (1985). Comparison of chase and inertial brake dynamometer testing of automotive friction materials. *Wear*, *103*, 217-232.

Zhang, H., & Chow, M.-Y. (2010). Comprehensive dynamic battery modeling for PHEV applications. In *IEEE power and energy society general meeting*.

BIOGRAPHIES

Brian M. Bole graduated from the FSU-FAMU School of Engineering in 2008 with a B.S. in Electrical and Computer Engineering and a B.S. in Applied Math. Brian received a M.S. degree in Electrical Engineering from the Georgia Institute of Technology in 2011, and he is currently pursuing a Ph.D. Brian's research interests include: analysis of stochastic processes, risk analysis, and optimization of stochastic systems. Brian is currently investigating the use of risk management and stochastic optimization techniques for optimal adaptation of active component load allocations in robotic and aviation applications. In a previous project, Brian worked with the Georgia Tech EcoCar team to develop an energy management controller for optimizing the fuel economy of a charge sustaining hybrid electric vehicle.

Christopher Teubert received his B.S. in Aerospace Engineering from Iowa State University in 2012. While at Iowa State University, he conducted research on asteroid deflection mission design and asteroid fragment propagation for Iowa State University's Asteroid Deflection Research Center (ADRC). Previous to his current position he worked as a spacecraft systems engineer for a Mars sample return mission as part of the NASA Academy Program. He is currently researching systems and algorithms for diagnostics, prognostics, and system health management for Stinger Ghaffarian Technologies, Inc. at NASA Ames Research Center's Prognostic Center of Excellence (PCoE). He plans to begin pursuing a M.S. in 2014.

Quach Cuong Chi Cuong C. Quach got his M.S. from the School of Physics and Computer Sciences at Christopher Newport University in 1997. He is a staff researcher in the Safety Critical Avionics Systems Branch at NASA Langley Research Center. His research areas include development and testing of software for airframe diagnosis and strategic flight path conflict detection.

Edward Hogge received a B.S. in Physics from the College of William and Mary in 1977. He has provided engineering services to the government and currently is employed by Northrop Grumman Technical Services. He has recently been supporting aviation safety research through the implementation of electronic systems for subscale remotely piloted aircraft and through commercial aircraft simulation. He is a member of the American Institute of Aeronautics and Astronautics.

Sixto Vazquez obtained his MSEE degree from Old Dominion University in 1990 and BSEE from the University of Puerto Rico in 1983. He has developed real-time 3D graphical simulations to aid in the visualization and analysis of complex sensory data. Has developed techniques to interactively process, analyze, and integrate sensory data from multiple complex, state-of-the-art sensing technologies, i.e. FMCW Coherent Laser Radar range measuring system, Bragg grating Fiber Optic Strain Sensing system, etc., into simulation.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universitt Mnchen, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics & Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the Associate Principal Investigator for Prognostics of NASAs Integrated Vehicle Health Management Program. He worked at General Electrics Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds eleven patents and has published more than 100 papers in the area of systems health management.

Vachtsevanos George is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, a M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at Georgia Tech where faculty and students are conducting research in intelligent control, neurotechnology and cardiotechnology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 240 technical papers and is a senior member of IEEE. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.