PROGNOSTICS APPLIED TO ELECTRIC PROPULSION UAV

Kai Goebel
NASA Ames Research Center
Moffett Field, CA 94035
USA

Bhaskar Saha
Palo Alto Research Center
Palo Alto, CA 94304
USA

Abstract (A 250 word brief about the chapter.)

Health management plays an important role in operations of UAV. If there is equipment malfunction on critical components, safe operation of the UAV might possibly be compromised. A technology with particular promise in this arena is equipment prognostics. This technology provides a state assessment of the health of components of interest and, if a degraded state has been found, it estimates how long it will take before the equipment will reach a failure threshold, conditional on assumptions about future operating conditions and future environmental conditions. This chapter explores the technical underpinnings of how to perform prognostics and shows an implementation on the propulsion of an electric UAV. An accurate run-time battery life prediction algorithm is of critical importance to ensure the safe operation of the vehicle if one wants to maximize in-air time. Current reliability based techniques turn out to be insufficient to manage the use of such batteries where loads vary frequently in uncertain environments. A particle filter is shown as the method of choice in performing state assessment and predicting future degradation. The method is then applied to the batteries that provide power to the propeller motors.

Introduction

Prognostics Health Management (PHM) is a engineering discipline that aims at maintaining nominal system behavior and function, and assuring mission safety and effectiveness under off-nominal conditions. PHM encompasses a set of wide-ranging sub-disciplines that address the design, development, operation, and lifecycle management of subsystems, vehicles, and other operational systems. Starting with simple time-temperature recorder for the engine hot-section on the F-8 aircraft (during deployment in Vietnam), and later the A-7 aircraft engine health monitoring program of the early 80's, PHM principles found their way in various aircraft. Analysis of vibration and acoustic emissions data from rotorcraft drivetrains have led to breakthroughs in predicting impending failures of these complex mechanical systems, resulting in the development of relative mature Health and Usage Monitoring Systems (HUMS) for rotorcraft (Revor and Bechhoefer 2004). Service providers like GE, PW, and Rolls-Royce have employed PHM principles to remotely monitor jet engines around the clock to detect early signs of damage as part of guaranteed uptime service agreements.

Prognostics

Prognostics is a core element of PHM. It is a younger member in the family of health management techniques but has recently received considerable attention due to its game-changing potential. Prognostics is the science of determining the remaining useful life of a component or subsystem given the current degree of wear or damage, the component's load history, and anticipated load and environmental conditions. A quantification of the degree of a component's wear or damage and the estimate of end-of-life gives decision makers important information about the health of a system. This information can be used on UAV for risk reduction in go/no-go decision, cost reduction through the scheduling of maintenance as-needed, and improved asset availability. Prognostics employs techniques that are often based on a detailed analysis of historical data or an analysis of the fault modes and the modeling of the physics of both the component itself as well as the attributes that characterize the fault. For the latter, the idea is to model the progression of damage which includes the effects of damage accelerators or stressors (such as load or environmental conditions). Next, algorithms that estimate the remaining life use estimation techniques that propagate the anticipated degradation into the future and provide as output the point where the component does no longer meet its desired functionality. The algorithms use these physics-based models as well as measurements from the system as input. Output of the algorithms is the remaining life estimate. An alternative to physics-based models are data-driven techniques. These synthesize a behavioral representation from a large number of example run-to-failure trajectories via machine learning methods.

Prognostics can be developed for almost any critical component as long as one has either some knowledge about the underlying physics or a sufficient amount of run-to-failure data exists. The efficacy of prognostics has been demonstrated for a wide range of diverse components ranging from mechanical components to electro-chemical components to electronics. Specific application areas range from rotating machinery (Marble and Tow 2006) to batteries (Saha et al. 2007) , from printed circuit boards (Gu et atl. 2007) to solid rocket motors (Luchinsky et al. 2008) . In the U.S. military, two significant weapon platforms were designed with a prognostics capability as an integral element of the overall system architecture: the Joint Strike Fighter Program (Hess et al. 2004) and the Future Combat Systems Program (Barton 2007) . Prognostic technology is also finding its way into future NASA launch vehicles and spacecraft (Osipov et al. 2007) as well as UAV (Valenti et al. 2007). As the technology matures further, it is expected that prognostics will play an important role in the design and operation of commercial systems such as passenger aircraft, automobiles, ships, the energy infrastructure, and consumer electronics.

Modeling

Underlying any prediction is a model that describes how the component of interest behaves under nominal conditions and how it will evolve as it experiences wear or a fault condition. To that end, one needs to capture that process in the form of a mathematical model. The model may be derived from laws of physics, captured by empirical relations, or learned from data. Models can also use a combination of these approaches. For example, parts of a component that are well-understood may be constructed using physics-based models, with unknown physical parameters learned from data using appropriate system identification techniques.

Modeling of physics can be accomplished at different levels, for example micro and macro levels. At the micro level, physical models are embodied by a set of dynamic equations that define relationships, at a given time or load cycle, between damage (or degradation) of a component and environmental and operational conditions under which the component is operated. The micro-level models are often referred to as damage propagation model. For example, Yu and Harris's fatigue life model for ball bearings, which relates the fatigue life of a bearing to the induced stress (Yu and Harris 2001), Paris and Erdogan's crack growth model (Paris and Erdogan 1963), and stochastic defect-propagation model (Li et al. 2000) are other examples of micro-level models. Since measurements of critical damage properties (such as stress or strain of a mechanical component) are rarely available, sensed system parameters have to be used to infer the stress/strain values. Micro-level models need to account in the uncertainty management the assumptions and simplifications, which may pose significant limitations of that approach.

Macro-level models are characterized by a somewhat abridged representation that makes simplifying assumptions, thus reducing the complexity of the model (typically at the expense of accuracy). An example is a lumped parameter model which assumes that the attributes of the component have idealized behavior and the non-ideal characteristics are characterized with equivalent elements that suffice for a first-order approximation. When such a system is designed well, it will often (but certainly not always) result in satisfactory results, depending on performance requirements. It should be noted that for complex systems (e.g., a gas turbine engine), even a macro-level model may be rather time-consuming and labor intensive. The resulting simplifications may need to be accounted for via explicit uncertainty management.

Nominal Behavior

The system model describes the characteristics of the system under nominal conditions. Ideally, such a model should be able to factor in the effects of operational and environmental conditions as well as any other conditions that cause different system response under nominal conditions. The model should also be able to adapt to changes of the system that are not considered abnormal. To that end, the system model could learn system behavior from examples, for instance using machine learning techniques or it could integrate domain expertise and be implemented using rules.

Damage Modeling

A damage propagation model describes how the damage is expected to grow in the future. It should, similar to the system model, account for operational and environmental conditions as well as any other conditions that have an impact on the damage. While one often thinks of damage as a monotonically increasing phenomenon, it is possible for the domain in which damage is evaluated to have non-monotonic attributes. These could be either intrinsic attributes (for example recovery effects in batteries or power semiconductors) or extrinsic effects such as partial repair actions. Depending on the fault mode, damage propagation may exhibit different symptoms and it may be necessary to consider dedicated damage propagation models for different fault modes.

Prognostics Algorithms

The role of the prognostic algorithm is applying the damage propagation model into the future. It needs to ensure that it properly considers the effects of environmental and operational conditions, healing phenomena, as well as how to account for the different sources of uncertainty. Depending on the implementation, the damage propagation model and the prognostic algorithm may not be separable. For the general case, the damage propagation model and the prognostic algorithm will be treated as separate.

Model-Based Algorithms

In order to make a prediction, the current health state of the system must first be known. Determining the current state of the system health is generally known as health state estimation. Given the current state of health, the model-based prediction algorithm propagates this estimate forward in time using damage propagation model equations up to the end-of-life (EOL) threshold. As mentioned earlier, the evolution of the state depends on future stressors. Therefore, these must be postulated.. For many applications, some knowledge about future stressors exists where tasks are scheduled or repeated. In other cases, one has only statistical information about future stressors. The accuracy and precision of the predictions depend on both the quality of the model and the uncertainty of future inputs.

The prediction algorithm of choice depends on the type of model used, on what information is needed to describe the EOL (which could be in form of a distribution), and the amount of computation that may be performed.

This section takes a look at Particle Filters, one of the most prevalent algorithms for model-based prediction.

Particle Filter

Particle methods assume that the state equations can be modeled as a first order Markov process with additive noise and conditionally independent outputs (Arulampalam et al. 2002). Let:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \omega_{k-1} \qquad (1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \nu_k \,. \qquad (2)$$

While there are several flavors of particle filters, the focus is here on *Sampling Importance Resampling* (SIR), in which the posterior filtering distribution denoted as $p(\mathbf{x}_k \mid \mathbf{Z}_k)$ is approximated by a set of N weighted particles $\{\langle x_p^i, w_p^i \rangle; i = 1,\ldots,N\}$ sampled from a distribution $q(\mathbf{x})$ that is "similar" to $\pi(\mathbf{x})$, i.e., $\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$. The *importance weights* w_k^i are then normalized:

$$w_k^i = \frac{\pi(\mathbf{x}_k^i)/q(\mathbf{x}_k^i)}{\sum_{j=1}^{N} \pi(\mathbf{x}_k^j)/q(\mathbf{x}_k^j)} \qquad (3)$$

such that $\Sigma_i w_k^i = 1$, and the posterior distribution can be approximated as:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \qquad (4)$$

Using the model in Eq. (1) the prediction step becomes:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^i \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^i). \qquad (5)$$

The weights are updated according to the relation:

$$\overline{w}_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)\,p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\mathbf{z}_k)}, \qquad (6)$$

$$w_k^i = \frac{\overline{w}_k^i}{\sum_{j=1}^{N} \overline{w}_k^j}. \qquad (7)$$

It is possible that all but a few of the importance weights degenerate such that they are close to zero. In that case, one has a very poor representation of the system state (and also wastes computing resources on unimportant calculations). To address that, *resampling* of the weights can be used (Saha et al. 2007). The basic logical flowchart is shown in Figure 1.

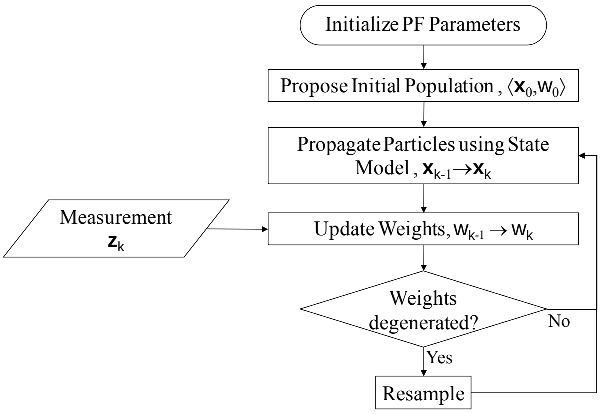

Figure 1: Particle filtering flowchart

During prognosis this tracking routine is run until a long-term prediction is required, say at time t_p, at which point Eq. (1) will be used to propagate the posterior pdf given by $\{\langle x^i_p, w^i_p\rangle; i = 1,\ldots,N\}$ until $\mathbf{x}^i$ fails to meet the system specifications at time t^i_{EOL}. The RUL pdf, i.e., the distribution $p(t^i_{EOL} - t_p)$, is given by the distribution of w^i_p. Figure 2 shows the flow diagram of the prediction process.

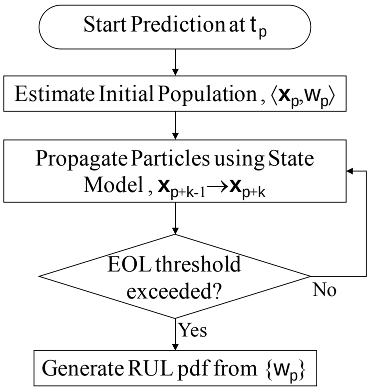

Figure 2: Prediction flowchart

Data-Driven Algorithms

As the name implies, data-driven techniques utilize monitored operational data related to system health. Given the availability of data, data-driven approaches are appropriate when the understanding of first principles of system operation is not easy to come by or when the system is sufficiently complex that developing an accurate model is prohibitively expensive. The principal advantage of data driven approaches is that they can often be deployed quicker and cheaper compared to other approaches, and that they can provide system-wide coverage. On the other hand, data driven approaches require a substantial amount of data for training which is a fundamental limitation for most systems, since full trajectories to failure are not recorded in large numbers for components in high-value systems like aircraft. Data-driven approaches can be further subcategorized into fleet-based statistics and sensor-based conditioning. In addition, data-driven techniques also subsume cycle-counting techniques that may include domain knowledge.

There are two basic data-driven strategies that involve either (1) modeling cumulative damage (or, equivalently, health) and then extrapolating out to a damage (or health) threshold, or (2) learning directly from data the remaining useful life.

As mentioned, a principal bottleneck is the difficulty in obtaining run-to-failure data, in particular for new systems, since running systems to failure can be a lengthy and rather costly process. Even where data exist, the efficacy of data-driven approaches is not only dependent on the quantity but also on the quality of system operational data. These data sources may include temperature, pressure, oil debris, currents, voltages, power, vibration and acoustic signal, spectrometric data as well as calibration and calorimetric data. Features must be extracted from noisy, high-dimensional data.

Case Study: Prognostics for Batteries Used in Electric UAV

This section will use batteries used for energy storage in electric UAV as an illustrative example to show the principles of prognostics. The section will start with a discussion on battery characteristics, followed by the model chosen and the implantation on the UAV

Battery Characteristics

Batteries are essentially energy storage devices that facilitate the conversion, or *transduction*, of chemical energy into electrical energy, and vice versa (Huggins 2008) . They consist of a pair of *electrodes* (*anode* and *cathode*) immersed in an *electrolyte* and sometimes physically divided by a *separator*. The chemical driving force across the cell is due to the difference in the chemical potentials of its two electrodes, which is determined by the difference between the *standard Gibbs free energies* the products of the reaction and the reactants. The theoretical *open circuit voltage*, E^o, of a battery is measured when all reactants are at 25^{o}C and at 1M concentration or 1 atm pressure. However, the voltage during use differs from the theoretical voltage because of various passive components like the electrolyte, the separator, terminal leads, etc. The voltage drop due to these factors can be mainly categorized into Ohmic drop, activation polarization, and concentration polarization.

Ohmic drop ΔE_{IR} refers to the diffusion process where Li-ions migrate to the cathode via the electrolytic medium. The internal resistance to this ionic diffusion process is also referred to elsewhere as the IR drop. For a given load current this drop usually decreases with time due to the increase in internal temperature that results in increased ion mobility.

Self-discharge is caused by the residual ionic and electronic flow through a cell even when there is no external current being drawn. The resulting drop in voltage has been modeled to represent the *activation polarization* of the battery ΔE_{AP}. All chemical reactions have a certain activation barrier that must be overcome in order to proceed and the energy needed to overcome this barrier leads to the activation polarization voltage drop. The dynamics of this process is described by the Butler–Volmer equation (Bockris and Reddy , 1973). This process was represented by an exponential function in (Saha and Goebel 2009) . However, a log function is a more accurate representation, as abstracted from the Butler–Volmer equation.

The *concentration polarization* ΔE_{CP} represents the voltage loss due to spatial variations in reactant concentration at the electrodes. This is mainly caused when the reactants are consumed by the electrochemical reaction faster than they can diffuse into the porous electrode, as well as due to variations in bulk flow composition. The consumption of Li-ions causes a drop in their concentration along the cell, which in turn causes a drop in the local potential near the cathode. The value of this factor is low during the initial part of the discharge cycle and grows rapidly towards the end of the discharge or when the load current increases.

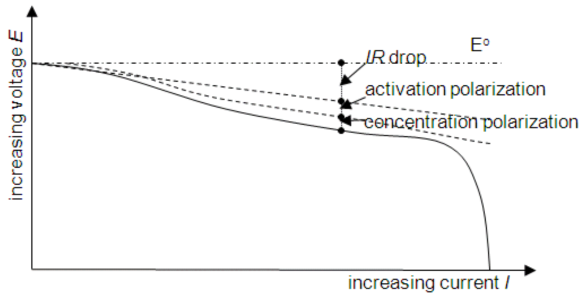

Figure 3: Typical polarization curve of a battery (Saha and Goebel 2009)

Figure 3 depicts the typical polarization curve of a battery with the contributions of all three of the above factors shown as a function of the current drawn from the cell. The voltage drop usually increases with increasing output current.

The output current plays a prominent role in determining the losses inside a battery and is therefore an important parameter to consider when comparing battery performance. The term most often used to indicate the rate at which a battery is discharged is the *C-Rate (Huggins 2008)* . The discharge rate of a battery is expressed as C/r, where r is the number of hours required to completely discharge its nominal capacity. So, a 5 Ah battery discharging at a rate of $C/10$ or 0.5 A would last for 10 hours. The terminal voltage of a battery, as also the charge delivered, can vary appreciably with changes in the C-Rate. Furthermore, the amount of energy supplied, related to the area under the discharge curve, is also strongly C-Rate dependent. Figure 4 shows the typical discharge of a battery and its variation with C-Rate. Each curve corresponds to a different C-Rate or C/r value (the lower the r the higher the current) and assumes constant temperature conditions.

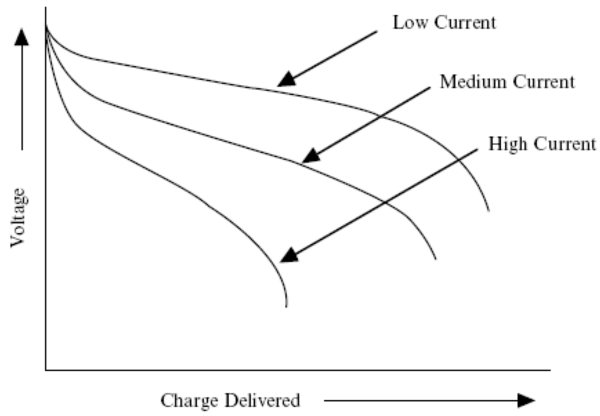

Figure 4: Schematic drawing showing the influence of the current density upon the discharge curve (Reproduced from Figure 1.14 in (Huggins 2008))

Model Adaptation

For many engineered systems models for nominal operation are available, but damage propagation models like Arrhenius model or Paris' law are comparatively rare. Developing these models may require destructive testing for model validation which may not be possible in many cases. In some cases, testing may be done on subscale systems, but there may be difficulty in generalizing the models learned. Additionally, the parameter values of these models are often system specific, and thus need to be re-learned for every new application. The PF framework described above can help in these cases by adapting the prognostic/aging model in an online fashion.

One of the key motivating factors for using Particle Filters for prognostics is the ability to include model parameters as part of the state vector to be estimated. This allows *model adaptation* in conjunction with state tracking, and thus, produces a tuned model that can used for long term predictions.

Let system health x_k, state evolution model $\mathbf{f}$ and measurement model $\mathbf{h}$ with known noise distributions ω and ν respectively. Additionally, the parameter values of $\mathbf{h}$ are assumed to be known (without lack of generality). The system health state is assumed to be 1-dimensional. Stationary or (better) non-stationary measurement models can be used to account for progressive degradation in sensors caused by corrosion, fatigue, wear, etc. The parameters of $\mathbf{f}$, denoted by $\alpha_k = \{a_{j,k}; j = 1,\ldots,n_f\}$, $n_f \in \mathbb{N}$, are combined with x_k to give the state vector $\mathbf{x}_k = [x_k\ \alpha_k]^T$, where T represents the transpose of a vector or matrix. Equations (1) and (2) can then be rewritten as:

$$x_k = \mathbf{f}(x_{k-1}, \alpha_{k-1}) + \omega_{k-1} \quad (8)$$

$$z_k = \mathbf{h}(x_k) + \nu_k . \qquad (9)$$

To formulate the state equations for α_k, one can choose a *Gaussian random walk* such that:

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k-1} \qquad (10)$$

where $\omega_{j,k-1}$ is drawn from a normal distribution, $\mathcal{N}(0,\sigma_j^2)$, with zero mean and variance σ_j^2. Given a suitable starting point $\alpha_{j,0}$, and variance σ_j^2, the PF estimate will converge to the actual parameter value $\bar{\alpha}_j$, according to the *law of large numbers*. In this way, model adaptation has been introduced into the PF framework, adding n_f extra dimensions, yet achieving convergence (and without incurring the curse of dimensionality).

The notion of a good proposal density, though, comes into play in the choice of the values of $\alpha_{j,0}$ and σ_j^2. If the initial estimate $\alpha_{j,0}$ is far from the actual value and variance σ_j^2 is small, then the filter may take a large number of steps to converge, if at all. The variance value may be chosen to be higher in order to cover more state-space, but that can also delay convergence. One way to counter this is to make the noise variance itself a state variable that increases if the associated weight is lower than a preset threshold, i.e., the estimated parameter value is far from the true value, and vice-versa. Thus:

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k-1}; \;\; \omega_{j,k-1} \sim \mathcal{N}\left(0,\sigma_{j,k-1}^2\right),$$

$$\sigma_{j,k} = c_{j,k} . \sigma_{j,k-1}; \begin{cases} c_{j,k} < 1, \text{if } w_{k-1} > w_{th}, \\ c_{j,k} = 1, \text{if } w_{k-1} = w_{th}, \\ c_{j,k} > 1, \text{if } w_{k-1} < w_{th}. \end{cases} \qquad (11)$$

The multiplier $c_{j,k}$, is a positive valued real number, while the threshold w_{th} is some value in the interval (0, 1). The intent is to increase the search space when the error is high and tightening the search when one is close to the target. Note that although this produces a better proposal density, it introduces a further n_f dimensions to the state vector.

It is quickly evident that it is not feasible to take this approach for all the parameters of a sufficiently high order model. This motivates the use of sensitivity analysis techniques (SA) to determine the more sensitive parameters that need to be estimated online. SA is essentially a methodology for systematically changing parameters in a model to determine the effects on the model output. There are several methods to perform SA like local derivatives (Cacuci 2003), sampling (Helton et al 2006), Monte Carlo sampling (Saltelli et al 2004), etc. Depending upon the form of the system model any of these methods may be used assess which parameters to target.

Assuming that the model function $\mathbf{f}$ in Eq. (8) is differentiable, i.e., $\partial \mathbf{f} / \partial \alpha_j$, (the time index k dropped for the sake of generality) it can be computed at any point in the state space defined by $\mathbf{x}_k = [x_k \; \alpha_k]^T$. If the partial derivative is positive, then the value of the function increases with an increase in the parameter value and vice-versa.

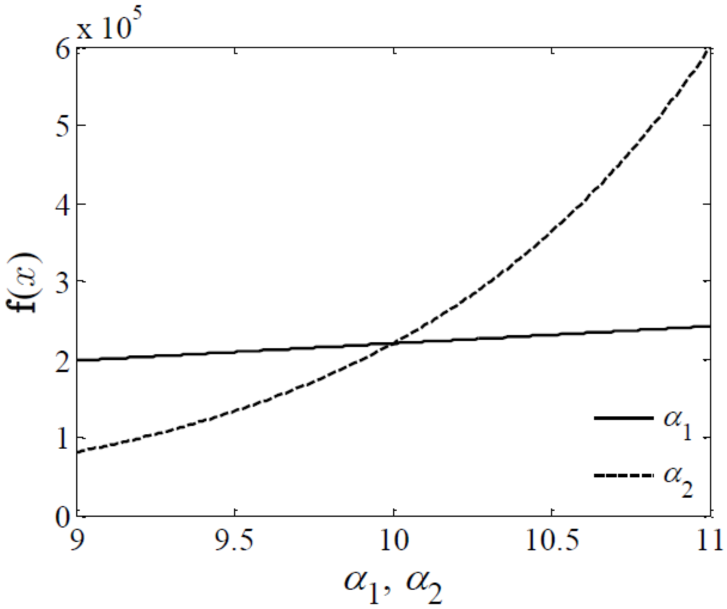

Figure 5: Effect on f(x) = $\alpha_1 \exp(\alpha_2 x)$ due to 10% variation in parameters α_1 and α_2. (Saha & Goebel, 2011)

The magnitude of the derivative indicates the degree to which the parameter affects the output of **f**, as shown in Figure 5. This directs the choice of the parameters to estimate online. If the *posterior error* given by:

$$e_k^i = x_k^i - \sum_{i=1}^{N} w_k^i x_k^i \,. \qquad (12)$$

is positive then the parameters that have a positive local partial derivative need to be reduced and those with a negative one need to be increased. The opposite holds true if the error is negative. The amount by which the parameters need to be reduced or increased also depends on the magnitude of the local partial derivative. The higher the magnitude, the smaller the steps needed in order to prevent instability while approaching the true value. This notion can be formalized in the following way (the particle index **i** has been dropped for the sake of generality):

$$\alpha_{j,k} = \alpha_{j,k-1} + C_{j,k} + \omega_{j,k-1}; \quad \omega_{j,k-1} \sim \mathcal{N}\left(0, \sigma_j^2\right). \qquad (13)$$

$$\begin{aligned} C_{j,k} &\propto -e_k \,, \\ &\propto \left.\frac{\partial \mathrm{f}}{\partial \alpha_{j,k}}\right|_{\mathbf{x}_k} , \\ &= -K \cdot \frac{e_k}{\left.\partial \mathrm{f} / \partial \alpha\right|_{j,k}\Big|_{\mathbf{x}_k}} \,. \end{aligned} \qquad (14)$$

Note that in this model adaptation scenario, the noise variance parameter is not added to the state vector since the search process is directed and not random as discussed previously.

Battery Model

For the purposes of the electric UAV BHM, the battery model design space is explored at a high level of abstraction with respect to the underlying physics. To predict the end-of-discharge (EOD), it is required to model the SOC of the battery. For the empirical charge depletion model under consideration here, the output voltage $E(t_k)$ of the cell is expressed in terms of the effects of the changes in the internal parameters (Saha et al. 2011):

$$E(t_k) = E^o - \Delta E_{IR}(t_k) - \Delta E_{AP}(t_k) - \Delta E_{CP}(t_k) \qquad (15)$$

where, E° is the Gibb's free energy of the cell, ΔE_{IR} is the Ohmic drop, ΔE_{AP} is the drop due to activation polarizaton and ΔE_{CP} denotes the voltage drop due to concentration polarization. These individual effects are modeled as:

$$\Delta E_{IRC}(t_k) = \Delta I_k \alpha_6 (1 - \exp(-\alpha_7 (t_k - t_{\Delta I_k}))) - \alpha_1 t_k \qquad (13)$$

$$\Delta E_{AP}(t_k) = \alpha_{2,k} \ln(1 + \alpha_{3,k} t_k) \qquad (14)$$

$$\Delta E_{CP}(t_k) = \alpha_{4,k} \exp(\alpha_{5,k} I_k t_k) \qquad (15)$$

where, ΔI_k is the step change in current at time $t_{\Delta I_k}$, and the α's represent the set of model parameters to be estimated.

The PF representation of the battery state for predicting EOD, is given by:

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k},\ j = 1, \ldots,. 7 \qquad (16)$$

$$x_k = E(t_k) + \omega_k \qquad (17)$$

$$\mathbf{x}_k = [x_k\ \alpha_{j,k}]^T,\ j = 1, \ldots,. 7 \qquad (18)$$

$$z_k = E(t_k) + \nu_k \qquad (19)$$

where all but the parameters α_3 and α_5 are learnt from training data, while α_3 and α_5 are estimated by the PF online.

UAV Application

Electric aviation concepts are receiving increaling attention because of their potential benefits for emissions, fuel savings, and possibly noise reduction. While electric propulsion is used in cincept manned aircraft, electric UAV are more common due to their lighter weight, and lower loss consequences (compared to manned flight). However, like ground vehicles, battery powered electric propulsion in aircraft suffer from uncertainties in estimating the remaining charge and hence most flight plans are highly conservative in nature. Different flight regimes like takeoff/landing and cruise have different power requirements and a dead stick condition (battery shut off in flight) can have catastrophic consequences. To tackle this issue the battery health management algorithm was designed and implemented on an embedded hardware platform and integrated it into an UAV airframe to provide real-time on-board battery life predictions (Saha et al. 2012).

The particular UAV platform is a COTS 33% scale model of the Zivko Edge 540T as shown in Figure . The UAV is powered by dual tandem mounted electric out-runner motors capable of moving the aircraft up to 85 knots using a 26 inch propeller. The gas engine in the original specification was replaced by two electric out runner motors which are mounted in tandem to power a single drive shaft. The motors are powered by a set of 4 Li-Poly rechargeable batteries, each rated at 6000 mAh. The tandem motors are controlled by separate motor controllers.

Figure 6: Edge 540 UAV

A 12 channel JR radio system is used to control the airplane. The system communicates in the 2.4 GHz band using a proprietary DSM2 protocol. Control surfaces are manipulated by seven actuators.

The airplane is equipped with a number of sensors to collect structure, propulsion, and navigation. The health of the structure is monitored using a series of strain gauges and accelerometers. Navigation data consist of GPS location, ground speed, altitude, true heading, and magnetic heading. Power plant data (such as motor RPMs, currents, battery voltages and temperatures) help to assess adequacy of the thrust from the motors and 26 inch propeller, the relevant parameters being.

Data from sensors are logged by 2 separate data systems instantiated in a combination of RCATS and PC104. The RCATS is a turn-key system with proprietary software that creates an ASCII log of connected sensors. It provides telemetry data to a laptop receiver which displays the data for callout to the pilot. It measures flight related parameters such as motor RPM, motor temperature, airspeed, z-axis acceleration at 10 Hz and interleaves GPS position and altitude data at 1 Hz. The PC104 stack consists of a CPU board, a DC/DC converter, an IO card, and a signal conditioning card for strain gauges. It runs MathWork's xPC Target operating system. The data are acquired using a Simulink model that is compiled to an xPC target OS. It records strain, accelerometer, battery temperature, and motor current at 200 Hz. It also outputs a 0.5 Hz sine wave to the RCATS system for synchronizing data.

The BHM system is designed to be a relatively low cost analog-to-digital data acquisition system. The design philosophy behind the first BHM system is to use COTS solutions which would have a light weight compact footprint. Figure 7 shows the system installed onboard the Edge 540 airframe.

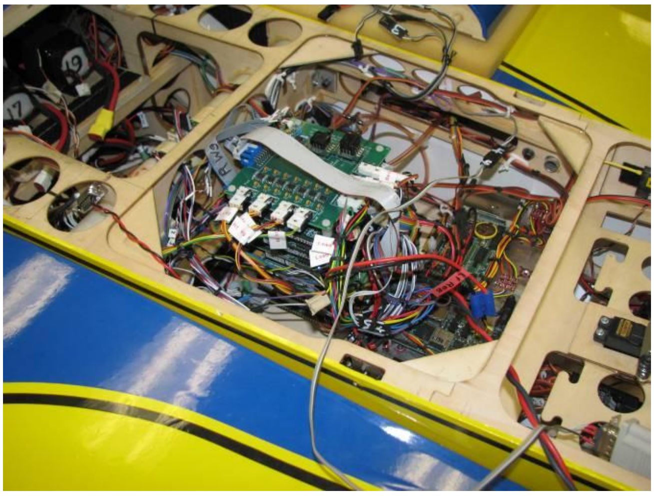

Figure 7 – The BHM system installed onboard

The BHM system itself utilizes several small COTS boards to convert TTL signal voltages into PC RS-232 signal voltages.

Implementation Results

Testing on the Edge 540 UAV platform was initially carried out with the airframe restrained on the ground. The propeller was run through various load regimes indicative of the intended flight profile (takeoff, climb, multiple cruise, turn, and glide segments, descent and landing) by varying the propeller rotational speed. Figure 4 shows the voltages during a typical profile. It is desired to predict when the battery will run out of charge, i.e., the EOD event indicated by the end of the voltage plots after landing.

In order to evaluate the prognostic algorithm multiple predictions were made at the time instants 13, 15, 17 and 19 minutes. It is not desired to completely discharge the batteries in flight since there needs to be some time for the UAV pilot to land the aircraft with some safety margin on the remaining battery life.

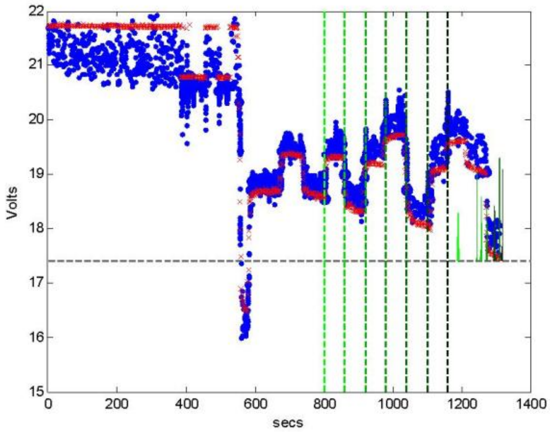

Figure 5: Predictions (vertical lines) during ground test (Saha et al. 2012)

In order to validate the learned prognostic model several dozen actual flight tests were conducted using the UAV with randomized flight profiles. The prediction performance was accurate to within 2 minutes, i.e. $|t_{EOD} - \bar{t}_{RUL,p}| < 2$ mins, over multiple flights of durations between 15 to 25 minutes. Figure 6 shows the profile of one such flight.

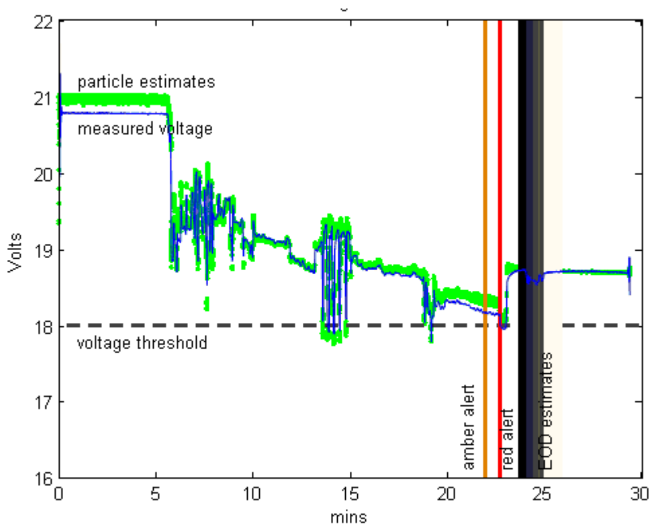

Figure 6: Battery voltage prediction using Particle Filter during flight test (Saha et al. 2012)

The blue line indicates the measured voltage while the green dots indicate the state values of the PF. The grey lines denote the $\mu_{RUL,p}$ values plotted every second, while the amber and red lines represent early alerts for the pilot to land the plane before the dead stick condition.

Conclusion

This chapter lays out a technique for predicting component degradation as exemplified on a battery used for propulsion of an electric UAV. The approach chosen here is model-based where the form of the model has been linked to the internal processes of the battery and validated using experimental data. The model was used in a PF framework to make predictions of EOD. By profiling the power required for different flight regimes like cruise segments, banked turns and landings, one can estimate the mission completion probability by calculating the RUL pdf. Since the prediction result is in the form of a pdf, it is easy to integrate the BHM routine into a higher level decisioning algorithm that can provide advance warning about when to land the aircraft.

Generally, a similar process can be followed for other components where a damage progression model is built for the component at hand. Naturally, the model would be different, if, say, a bearing, an electronics component, or a structural element were the object of interest.

The next step in Prognostics Health Management is to integrate the health information, in this case remaining life, into a decision making unit that reacts appropriately. Depending on the prognostic horizon, the reaction could be to autonomously change controller settings, reconfigure system resources to ensure primary mission goals (and perhaps extend the life of the stressed component), invoke a replanning or rescheduling routine, or provide the information for later processing in a maintenance setting.

Acknowledgement

This work was a performed as a cross-center collaboration between NASA Ames and Langley Research Centers (ARC and LaRC) and Dryden Flight Research Center (DFRC). The authors would like to especially thank Sixto L. Vazquez, Edward F. Hogge, Thomas H. Strom and Boyd L. Hill at LaRC, and Edwin Koshimoto at DFRC for their contributions. The funding for this work was provided by the NASA System-Wide Safety and Assurance Technologies (SSAT) project under the Aviation Safety Program of the Aeronautics Research Mission Directorate (ARMD).

References

[1] Arulampalam S, Maskell S, Gordon NJ, Clapp T (2002) A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. In: IEEE Trans on Signal Processing 50(2) pp 174-188

[2] Bockris, JO'M, Reddy AKN, Gamboa-Aldeco A (2000) Modern Electrochemistry 2A: Fundamentals of Electrodics, 2nd Edition, Kluwer Academic/Plenum Publishers

[3] Huggins R (2008) Advanced Batteries: Materials Science Aspects, 1st ed., Springer

[4] Saha B, Goebel K (2009) Modeling Li-ion Battery Capacity Depletion in a Particle Filtering Framework. In: Proc Annual Conf Prognostics and Health Management Society

[5] Saha B, Quach C, Goebel K (2011) Exploring the Model Design Space for Battery Health Management. In: Proc of the Annual Conf Prognostics and Health Management Society

[6] Revor M, Bechhoefer E (2004) Rotor track and balance cost benefit analysis and impact on operational availability. In: Proc American Helicopter Society 60th Annual Forum

[7] Marble S, Tow D (2006) Bearing health monitoring and life extension in satellite momentum/reaction wheels. In: Proc IEEE Aerospace Conf

[8] Saha B, Goebel K, PollS, Christophersen J (2007) An integrated approach to battery health monitoring using bayesian regression and state estimation. In: Proc IEEE Autotestcon, pp 646:653

[9] Gu J, Barker D, Pecht M (2007) Prognostics implementation of electronics under vibration loading. In: Microelectronics Reliability, vol. 47.

[10] Luchinsky D, Osipov V, Smelyanskiy V (2008) Model based IVHM system for the solid rocket booster. In: Proc IEEE Aerospace Conf

[11] Hess A, Calvello G, Dabney T (2004) PHM a key enabler for the JSF autonomic logistics support concept. In: Proc IEEE Aerospace Conf

[12] Barton P (2007) Prognostics for combat systems of the future. In: IEEE Inst & Meas Mag, vol. 10, Aug 2007

[13] Osipov V, Luchinsky D, Smelyanskiy V (2007) In-flight failure decision and prognostics for the solid rocket booster. In: AIAA 43rd AIAA/ASME/SAE/ ASEE Joint Propulsion Conf

[14] Yu WK, Harris T (2001) A new stress-based fatigue life model for ball bearings. In: Tribology Trans, Vol. 44, pp 11:18

[15] Paris P, Erdogan F (1963) A critical analysis of crack propagation laws. In: Trans. The ASME, J. Basic Eng. pp 528:534

[16] Saha B, Goebel K, Poll S, Christophersen J (2009) Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework. In: IEEE Trans Inst. & Meas., vol.58, no.2, pp 291-296

[17] Cacuci DG (2003) Sensitivity and Uncertainty Analysis: Theory, Vol I, Chapman & Hall

[18] Helton JC, Johnson JD, Salaberry JC, Storlie JB (2006) Survey of sampling based methods for uncertainty and sensitivity analysis. In: Reliability Engineering and System Safety, vol. 91, pp 1175:1209

[19] Saltelli A, Tarantola S, Campolongo F, Ratto M (2004). Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, John Wiley & Sons

[20] Bockris O'M, Reddy AKN (1973) Modern Electrochemistry, vol. 2, Plenum Publishing Corporation, New York, pp 845:1136

[21] Saha B, Quach P, Goebel K (2012) Optimizing Battery Life for Electric UAVs using a Bayesian Framework. In: Proc 2012 IEEE Aerospace Conf

[22] Valenti M, Bethke B, How D, de Farias D, Vian J (2007) Embedding Health Management into Mission Tasking for UAV Teams. In: Proc American Control Conf

[23] Li Y, Kurfess, T, Liang S (2000) Stochastic Prognostics for Rolling Element Bearings. In: Mechanical Systems and Signal Processing, Vol. 14 (5), pp 747:762