Calculating Remaining Useful Life In An Embedded System

Eric Bechhoefer[1], and Rune Schlanbusch[2]

[1]*GPMS Inc., Cornwall, VT 05753*
eric@gpsm-vt.com

[2]*Teknova AS, Grimstad, Norway*
rune.schlanbusch.2008@ieee.org

ABSTRACT

This paper presents a prognostic algorithm with low computational requirements that was implemented on an embedded system. The health of components such as shafts, bearings, and gears, is estimated based on a paradigm where condition indicators (CIs) are mapped into a health indicator (HI) for detecting and identifying faults based on vibration data. For estimating the time evolution of the HI along with remaining useful life (RUL) and its evolution, an alpha-beta and alpha-beta-gamma trackers are prosed. The estimator assumes that the plant noise is converging to a steady state over time. The advantage of the proposed filters is considerably less memory usage and computational load than a Kalman filter. The efficiency of the proposed method is demonstrated with a known fault case based on real-world data. The demonstration shows that the state observer is capable of tracking the evolution of the HI and estimates the RUL with sufficient robustness and performs favorably to previous results based on the Kalman filter. The method is well suited to computing on lower cost smart sensors.

1. INTRODUCTION

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. Condition monitoring is the process of determining the condition of machinery using vibration, acoustic emissions, lubrication analysis, motor current, model-based current systems, etc. Condition monitoring processes generally measure features of the component under analysis, then uses algorithms or heuristics to generate condition indicators (CIs). Remaining useful life (RUL) is the time until it is appropriate to perform maintenance. There are several applications where

an embedded calculation of RUL can be used for logistic support and asset planning. This intern will allow for the system to be operated at lower risk for unscheduled maintenance.

As noted, the process of condition monitoring generally measures some features related to wear or damage of a component. The features, CIs, are statically representation of the condition of the component under analysis (Vercer, 2005). For a rotating shaft, a CI might be the measure of the first, second and third order harmonics. The first harmonic (Shaft Order 1, or SO1) is used to measure imbalance. The second harmonic (SO2) indicates a bent shaft, while the third harmonics (SO3) is indicative of a coupling failure. Sometimes both SO2 and SO3 are excited by a coupling failure. Using the additional *a priori* configuration items, a decision can be made as to when to recommend maintenance based on the measured CIs. This *a priori* configuration describes a threshold. Threshold setting is used, again, to allow CIs to recommend a maintenance event.

For many components, no single CIs can determine the condition of the component. As noted, for a shaft, there are at least three CIs, which can be used to determine the shaft condition. Further, as measured acceleration is the second derivative of displacement, the CIs threshold for one shaft likely will be different for a shaft with a different mass or rotation rate (as acceleration is a function of shaft rate, squared). As such, for systems that only use CIs, a great deal of expertise is needed to interpret these CIs.

The *a priori* configuration, if appropriately modeled, can give a representation of the component's health using a health indicator (HI, see Bechhoefer, He, Dempsey, 2011). The HI is a mapping from 1 to *n* CIs that scales the CIs into a common threshold nomenclature. Typically, the HI is modeled such that the HI ranges from 0 to a positive value, where the probability of exceeding an HI of 0.5 is the probability of false alarm (PFA). A warning alert is generated when the HI is greater than or equal to 0.75. An

HI greater than 0.75 indicates that maintenance should be planned, by estimating the RUL until the HI is 1.0. An HI greater then 1 indicates an alarm situation, where the continued operation will result in a lower reliability of the system, and potential collateral damage to other components within the system.

This HI nomenclature defines that an alarm alert is generated when the HI is greater than or equal to 1.0. Note that this nomenclature does not define a probability of failure for the component within a system, or that the component fails when the HI is 1.0. Instead, it suggests a change in operator/maintainer behavior to a proactive maintenance policy: perform maintenance prior to the generation of collateral or cascading faults. For example, by performing maintenance on a bearing prior to the bearing shedding extensive material, costly gearbox replacement can be avoided.

1.1. Component Health: Failure, Reliability, and RUL

In application, an HI greater than 1.0 is indicative of a component fault where it would be appropriate to do maintenance. That said, it is difficult to determine when actual failure might occur. For example, a cracked inner race ring on a bearing is a fault. It is clearly appropriate to do maintenance. Yet this bearing might run for 10's if not 100's of hours prior to seizing. The crack has reduced the reliability of the system.

For aircraft or other critical systems, the design reliability is typically "six-nines," e.g., the probability of failure of the part under design loads is less than 10^{-6} per hour. For the damaged part, the reliability may be reduced to three-nines or a probability of failure of 10^{-3}. The appropriateness to repair the faulty component, then, can be seen as an action to restore the designed reliability of the system as a whole.

The estimation of RUL requires five pieces of estimation:

- An estimate of the current component health.
- An estimate of when it is appropriate to do maintenance.
- An estimate of the future component load.
- A model of the component degradation process, which takes the current component health, the estimated future load, and calculating the time/cycles to when it is appropriate to do maintenance.
- Some measure of the confidence in the RUL.

The confidence in the RUL estimate is likely more useful than the RUL itself. A low confidence model suggests to the maintainer that the RUL should be ignored, while a high confidence model should give motivation to implementing a maintenance action. Further, various alerting or reporting processing can be implemented based on RUL confidence.

In the extreme, the RUL is not reported if it is of low confidence.

2. EMBEDDED RUL APPLICATIONS

There are a large number of applications where an embedded calculation of RUL will be of great use for logistic support and asset planning.

Consider a satellite system, were the attitude control uses a reaction wheel. The type of control device is useful because it uses no fuel or rockets for reaction. Such devices are particularly useful for aligning an antenna or a camera. The reaction wheel, being a type of flywheel, is a rotating device, which uses bearings. An embedded RUL estimation of the bearing life could be of great use to the satellite mission planner, as it indicates when the spacecraft needs to be replaced. This knowledge is vital for a mission-critical system where continuous coverage is essential, such as for communication or weather satellite.

In another scenario, consider a boiler circulator pump (BCP) in a power plant. Such pumps are critical for the power plant operation. A failure of a BCP typically causes the power plant to be taken offline, resulting in fines and higher operating cost for the power plant operator. The installation of a condition monitoring system on a BCP can indicate shaft imbalances, wear on the pump bearings, and detect faults within the motor, such as rotor bars, shorts, eccentricity, etc.

Any of the faults as mentioned earlier can cause the BCP to fail, resulting in the power plant being taken offline. An embedded RUL calculation could be used to trigger maintenance on the BCP opportunistically (e.g., when the plant is offline for scheduled maintenance) or allowing the operator time to bring other power plants online to take over the power generation. Further, an embedded RUL capability would allow plant maintenance personnel in their daily routine of visual inspection, to see the RUL displayed on the BCP. This is advantageous as often part of the expense of a condition-monitoring program is the installation cost (e.g., wiring to a remote server) and the software maintenance cost of hosting the condition monitoring on a server for display.

While helicopter designs are inherently safe, including an embedded RUL capability improves reliability, and therefore, safety. As an example, there have been several helicopter mishaps (CHC, 2015, Bond, 2009, Cougar, 2009, and CHC, 2016) of aircraft equipped with health and usage monitoring systems (HUMS), which is a general term for condition monitoring system on helicopters. The lack of an embedded RUL to notify the crew of impending failure resulted in the loss of life to crew and passengers in two of the references, and the loss of an asset in (CHC 2016). While an on-board condition monitoring system is a valuable safety tool, if the data is not processed and viewed

routinely, or if the fault propagates in a time period less than turnaround time of server/desktop based condition-monitoring systems, the system provides little value to improve safety.

3. THRESHOLD SETTING AND THE HI CALCULATION

All CIs have a probability distribution functions (PDFs). Any operation on the CI to form a health index (HI) is then a function of distributions. The HI function is defined as The norm of n CIs (which is normalized energy). The function is valid if and only if the distributions (CIs) are independent and identical.

As an example, for Gaussian distribution, subtracting the mean and dividing by the standard deviation will give identical Z distributions. The issue of ensuring independence is much more difficult. In general, the correlation between CIs is non-zero. For instance, Table 1 shows the correlation coefficients for 6 CIs used for gear fault analysis, many of which are non-zero (Bechhoefer, He & Dempsey, 2011).

Table 1: Correlation Coefficients for the Six CIs Used in the Study

ρ_{ij}	CI 1	CI 2	CI 3	CI 4	CI 5	CI 6
CI 1	1	0.84	0.79	0.66	-0.47	0.74
CI 2		1	0.46	0.27	-0.59	0.36
CI 3			1	0.96	-0.03	0.97
CI 4				1	0.11	0.98
CI 5					1	0.05
CI 6						1

This correlation between CIs implies that for a given function of distributions to have a threshold that operationally meets the design PFA, the CIs must be whitened (de-correlated). A whitening process can be found using Cholesky decomposition. The Cholesky decomposition of a Hermitian, positive definite matrix results in

$$A = LL^*, \qquad (1)$$

Where **L** is a lower triangular, and **L*** is its conjugate transpose. The inverse covariance is a positive definite Hermitian matrix. It then follows that if: $LL^* = \Sigma^{-1}$, then **Y** = **L** x **CI**^{T}. The vector **CI** is the correlated CIs used for the HI calculation, and **Y** is 1 to n independent CI with unit variance (one CI representing the trivial case). The Cholesky decomposition, in effect, creates the square root of the inverse covariance. This, in turn, is analogous to dividing the CI by its standard deviation (the trivial case of one CI). It can then be shown that **Y** = **L** x **CI**^{T} creates the

necessary independent and identical distributions required to calculate the critical values for a function of distributions.

As an example of the importance of correlation on, consider a simple HI function HI = CI_1 + CI_2. The CIs will be normally distributed with mean 0 and standard deviation of 1. The standard deviation of this HI is:

$$\sigma_{HI} = \sqrt{\sigma_{CI1}^2 + \sigma_{CI2}^2 + 2\rho_{CI1,CI2}\sigma_{ci1}\sigma_{ci2}} \qquad (2)$$

Where $\rho_{CI1, CI2}$ is the correlation between CI_1 and CI_2. If one assumes $\rho_{CI1,CI2}$ is 0.0, then $\sigma_{HI} = \sqrt{2}$. For a PFA of 10^{-6}, the threshold is then 6.722. Consider the case in which the observed correlation is closer to 1 (e.g., $\rho_{CI1, CI2}$ is 1.0), then the observed $\sigma_{HI} = 2$. For a threshold of 6.722, the operational PFA is 4 x 10^{-4}. This is 390 times greater than the designed PFA. This illustrates the effect of correlation on the threshold setting.

The CIs used for condition monitoring have Rayleigh like PDFs (i.e., heavily tailed, Byington et al., 2003). For magnitude based CIs, it can be shown that for the nominal case, the CI PDF is Rayleigh. For Gear CIs, and Bearing CIs (which have magnitudes that are biased by RMS), a transform is used by making the CI more Rayleigh like. The transform "left shifts" the CI such that the 0.05 cumulative distribution function (CDF) is assigned to 0.0.

Consequently, the HI function is designed using the Rayleigh distribution. The PDF for the Rayleigh distribution uses a single parameter, β, defining the mean

$$\mu = \beta*(\pi/2)^{0.5}, \qquad (3)$$

and variance

$$\sigma^2 = (2 - \pi/2) * \beta^2. \qquad (4)$$

The PDF of the Rayleigh is

$$pdf = x/\beta^2 exp(x/2\beta^2). \qquad (5)$$

Note that when applying these equations to the whitening process, we have $\sigma^2 = 1$, and thus the β value for each CI will then be

$$\beta = \sigma / (2 - \pi/2)^{1/2} = 1.5264. \qquad (6)$$

As noted, the HI function is the norm of n CIs. This represents the normalized energy of the CIs. If the CIs are IID, it can be shown that the function defines a Nakagami PDF. The statistics for the Nakagami are $\eta = n$, and $\omega = 1/(2-\pi/2)*2*n$, where n is the number IID CIs used in the HI calculation (Bechhoefer, Bernhard, 2007).

3.1. Procedure for Calculating Threshold

Guidance for the appropriate CIs can be found in the Aeronautical Design Handbook (McCall, 2013). For Bearing or Gear components, calculate the offset.

1. The offset is calculated using the empirical CDF.

2. Sort the CI values. The offset is the CDF value that is indexed from round(50*number of aircraft *0.05)

For Shafts, the offset index is zero. The CIs used for Shaft will generally be SO1, SO2, and SO3.

The CIs used for roller element Bearing will typically be Cage, Ball, Inner and Outer Race Energy.

The CIs used for Gear will generally be Residual RMS, Residual CF, Energy Operator RMS, FM0, AM RMS, FM RMS (See McCall 2013). Gears are complicated, and CI performance can be affected by externalities, such as changes in gear-to-gear contact angle, which is effect by torque/twist on the drive train (FM0, in particular, is sensitive to this). This may require selecting other CIs, or reduce the number of CIs used in the HI calculation.

For fleet statistics, under nominal load (> 30% torque) randomly pull 50 acquisitions for a minimum of 5 aircraft.

Thresholds can be set for fewer aircraft, at the risk of increasing the fleet level PFA

Typically, One should plan on three threshold releases.

- The prototype release. This is used to capture raw data, find gearing resonance, check kinematics

- The initial release, ideally with five aircraft

- The final release with 20+ aircraft.

In general, an operator should develop a condition monitoring plan. Development of the configuration, e.g., thresholding is not static. As feedback from depot level repairs/inspections is gathered, the thresholds should be updated to reflect operator/maintainer appropriateness for maintenance. In general, the development of the *a priori* configuration requires:

- Calculate the mean value and covariance for the component.

- Calculate the inverse of the covariance.

- Calculate the Nakagami critical value and on η and ω

- Set PFA: 10^{-6}.

- Critical Value T = Nakagami Inverse CDF for 1-PFA, η, ω.

- Calculate **L**, the Cholesky Decomposition value of the inverse covariance.

- Calculate the Scale values: $S = L * (0.5/T)^2$. This is an upper triangular matrix, which is stored as an array as configuration, along with the offset values.

The procedure defines the *a priori* data needed by the embedded system to calculate health in situ.

4. CALCULATION OF RUL IN AN EMBEDDED SYSTEM.

Because of the limited resources available, both regarding computation speed and memory (RAM), a fast and efficient method is needed to calculate the RUL of a component. The goal is to calculate and store an RUL state vector on the embedded system, which incorporates all of the data needed to update the next estimate of RUL. This is done in near real time. Such a system would be very efficient, as there is no need to process the entire history of the component health to estimate the RUL.

Paris's Law (Paris, 1961) relates the stress intensity factor to sub-critical crack growth under a stress regime. It is a popular fatigue crack growth model for homogeneous material. The basic model is:

$$\frac{da}{dN} = D(\Delta K)^m \qquad (7)$$

where

- *da/dN* is the rate of change in the half crack length per cycle

- *D* is a material constant

- *ΔK* is the range of strain, and

- *m* is the crack growth exponent.

The range of strain, ΔK, is defined as $2\sigma\alpha(\pi a)^{1/2}$, where

- σ is the gross strain, generally unknown but proportional to torque,

- α is some geometric correction factor, again generally unknown, and

- *a* is the half crack length, which is proportional to component health.

These variables are usually specific for some given material. Simplifying the model by assuming a crack growth exponent as 2 (which is typical for steel), and collapsing the geometric correction factor (an unknown constant) into *D*, then:

$$da/dN = D4\sigma^2\pi a \qquad (8)$$

Where *N* is the number of cycles. For constant rate machines, such as a helicopter gearbox, *N* is proportional to time. Taking the inverse, one now has:

$$\frac{dN}{da} = (D4\sigma^2 \pi a)^{-1} \qquad (9)$$

Integrating gives the number of cycles (e.g., time) from the current state a_0, to the remaining useful life state, a_f:

$$N = dN/da \times a_0 \times (\ln(a_f) - \ln(a_0)) \qquad (10)$$

If it is assumed that the current health is proportional to the crack length a_o, and, as noted early that the RUL is the estimated time from the current HI to an HI of 1 (e.g., $a_f = 1$), then the RUL is:

$$RUL = -dHI/dt \times HI \times \ln(HI) \qquad (11)$$

4.1. Model validation and confidence in the RUL.

Clearly, the operator or maintainer needs a level of confidence in the RUL. Conceptually, confidence relates to model validation. As such, a process needs to be implemented that in an automated way, validates the RUL model performance.

Consider that the RUL value for some given time is 100 hours. One hour into the future, the new RUL value is calculated as 100 hours. In this case, the rate of change of the model is 0, and subjectively, the confidence in the model is low as the model has not accounted for consuming one hour of life.

Consider another case, where one hour into the future, the RUL value is 90. Hence, the rate of change in the RUL is -10. Again, this is a poor model. The confidence in the RUL is low because only one hour of life has been consumed, but the model is suggesting it should be 10 hours.

Finally, consider a case where the model initially estimates an RUL of 100. Say that one hour in the future, the RUL is measured as 99. The rate of change in the RUL is -1.0, e.g. dRUL/dt = -1. Further, given one additional hour in the future, the RUL measurement is 98. Again the rate of change in the RUL is -1. As important, the rate of change of the rate of change (e.g., the second derivative) is zero. This gives insight into model validation and confidence. The model is indicating that the rate of change is consuming one hour of life, each hour. Conceptually, this is a valid model.

The estimation of dHI/dt, dRUL/dt, and d^2RUL/dt^2 are a typical class of problems solved using a state observer. A state observer is a model that provides an estimate of the internal state of a given system. While a Kalman filter is one type of state observer, it may not be the most appropriate solution for an embedded system.

In an embedded system using a microcontroller, there are limited computation recourses and constraints on RAM. It is well known in a Kalman filter that, along with estimating the state (e.g., HI, dHI/dt), an estimate of the *a posteriori* covariance is needed (Bar-Shalom, 1992). This will increase the computation and memory load on an embedded system. For a complex installation, the additional resources may exceed the embedded system resources.

The Kalman filter can be dynamic in the sense that the measurement noise, plant noise, and *a posteriori* covariance can be updated based on real time measurement. In most implementation the state gains quickly approach a steady state. If the assumption of stationarity is made (i.e.,

measurement noise, plant noise, and update rate are constant), then the performance of a Kalman filter can be obtained using an α-β tracker, (an "alpha-beta" tracker). Using an α-β tracker, two states the filtered estimate or HI, and dHI/dt are derived, or an α-β-γ tracker, (an "alpha-beta-gamma" tracker), where the three states represent RUL, dRUL/dt, d^2RUL/dt^2.

Given the assumption of stationarity, in effect, the α-β and α-β-γ, trackers are treated as steady-state Kalman filters. As such, assuming the limit as time moves toward infinity, the filter coefficients for the α-β tracker (used for HI, and dHI/dt) can be calculated as (Bar-Shalom, 1992)-

$$\lambda = \frac{\sigma_w dt^2}{\sigma_v}, \qquad (12)$$

$$r = \frac{4 + \lambda - \sqrt{8\lambda + \lambda^2}}{4}, \qquad (13)$$

Where the process variance is σ_w^2, and plant noise variance is σ_v^2. The filter gains are

$$\alpha_1 = 1 - r^2, \qquad (14)$$

$$\beta_1 = 2(2 - \alpha) - 4\sqrt{1 - \alpha}. \qquad (15)$$

The coefficients for the α-β-γ tracker (needed for RUL, dRUL/dt, and d^2RUL/dt^2) are

$$\lambda = \frac{\sigma_w dt^2}{\sigma_v}, \qquad (16)$$

$$b = \frac{\lambda}{2} - 3, \qquad (17)$$

$$c = \frac{\lambda}{2} + 3, \qquad (18)$$

$$p = c - \frac{b^2}{3}, \qquad (19)$$

$$q = \frac{2b^3}{27} - \frac{bc}{3} - 1, \qquad (20)$$

$$v = \sqrt{q^2 + \frac{4q^3}{27}}, \qquad (21)$$

$$z = -\sqrt[3]{q + \frac{v}{2}}, \qquad (22)$$

$$s = z - \frac{p}{3z} - \frac{b}{3}, \qquad (23)$$

so the filter gains are

$$\alpha_2 = 1 - s^2, \qquad (24)$$

$$\beta_2 = 2(1 - s)^2, \qquad (25)$$

$$\gamma_2 = \frac{\beta^2}{2\alpha}. \qquad (26)$$

Procedurally, if the component state file is empty, initialize the filtered HI (fHI), its derivative (dHI) to zero, the filtered RUL (fRUL) and its first and second derivative (dRUL, d2RUL) to zero, or read the component state from the

component state file. Then, if the first RUL estimation is from startup

- Initialize the α_1 and β_1 coefficients based on the measurement noise, the plant noise, and the update rate, dt, or retrieve from memory.

- Initialize the α_2, β_2, γ_2 coefficients based on the measurement noise, the plant noise and the update rate, dt.

It is assumed that the coefficients α_1 β_1, α_2, β_2, γ_2 are initialized at startup and persisted throughout the operation.

```
For each HI update:

  Recover fHI,dHI, fRUL, dRUL, d2RUL for
  the component state file.

  fHI = fHI + dHI * dt;
  rk = HI - fHI;        //rk is innovation
  fHI = fHI + alpha1*rk;
  dHI = dHI + (beta1*rk)/dt;
```

Calculate the RUL

```
  rul =  -1 / dHI * fHI * log(fHI);
```

Note that in application, it is assumed that parts/components do not "heal" or improve over time. Hence dHI is always greater than zero. In general, the minimum value of dHI is 1/mean time between failure or 1/design life.

Filter the RUL

```
  fRUL = fRUL + dRUL*dt + d2RUL*dt*dt/2;
  rk = RUL - fRUL;
  fRUL = fRUL + alpha2 * rk;
  dRUL = dRUL + beta2 * rk / dt;
  d2RUL = d2RUL+gamma2 * rk / (2*dt*dt);
 //update the component state file.
end
```

Further note that this series of calculation is run for each component, for each acquisition. After each component update, the component file (i.e., the component state file) is updated with HI, fHI, dHI, RUL, fRUL, dRUL and d2RUL. Additionally, similar to a CI file, an HI/RUL file entry is appended with the current component state. The computational load for each update is then 11 multiplications, nine additions, and a call to log.

In this way, the component health and RUL can be viewed, along with component CI values, on a user interface or browser user interface, during the mission, or after the mission, or after periodic download.

5. EMBEDDED RUL FOR A HIGH-SPEED BEARING

Example: A bused smart sensor system was installed on a 2.1 MW wind turbine. A six-second acquisition was made every 10 minutes, with periodic (24 hour) data download. The HI methodology for thresholding, as outlined, was utilized and maintenance was performed when the HI was greater than 1.0 (the inspection was made when the HI was at a value of 1.1, Figure 1, which indicated an inner race crack that spanned the width of the race).

Figure 1 Damaged High-Speed Bearing, HI 1.1

Figure 2 depicts approximately 55 days of data. Note that the fault starts to propagate at approximately time -700, which corresponded to high loads from a winter storm.
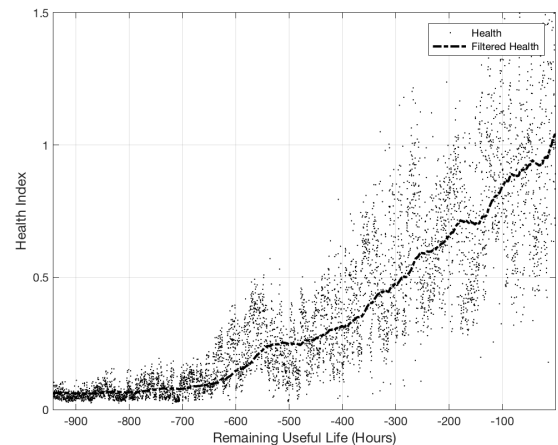

Figure 2 Measured and Filtered HI for a High-Speed Bearing

The state dHI/dt is given in Figure 3.

The RUL estimate and filtered RUL estimate are given in Figure 4. Note that before -700 hours, the RUL is effectively infinite. Once the fault starts to propagate, the RUL reduces quickly. Additionally note that the filtered RUL is subject to phase lag, which is a phenomenon of all recursive filters (seen from hour –600 to -400). This can be mitigated using a forward/backward, zero phase filter, such as a Kalman Smoother, at the expense of increased memory and computational burden.
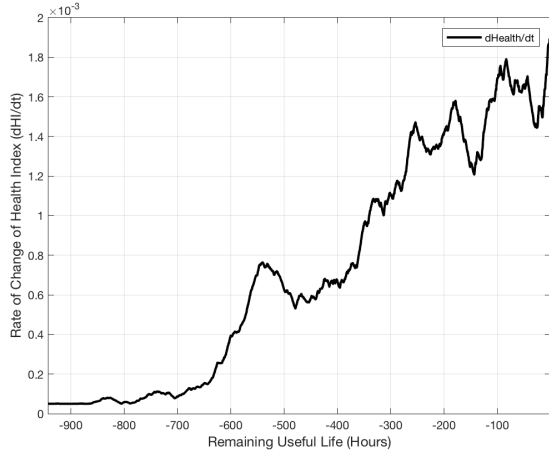
Figure 3 Estimated dHI/dt for the High-Speed Bearing

The computational burden of using a Kalman Smother or particle filter method may exceed the resources available on an embedded system. In this application, the effect of the phase lag, as will be shown, is minimal, justifying the use of the α-β-γ tracker.
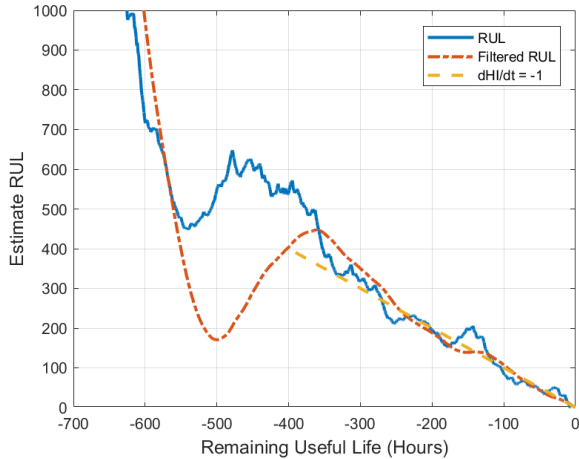

Figure 4 Linear "Life," RUL and Filter RUL of the Bearing

Note that the RUL and Filtered RUL converge to the linear "Life" at approximately 300 hours before the component bearing is removed. Figure 5 gives the first derivative of RUL. The upper subplot rectangle is from -400 to 0 hours, while the lower subplot shows the zoomed dRUL/dt, with a value approximately -1 dRUL/dt.

The second derivative of RUL is given in Figure 6. The upper subplot rectangle is from -400 to 0 hours, while the lower subplot shows the zoomed d^2RUL/dt^2, with a value approximately -.002 d^2RUL/dt. Figures 5 and 6 suggest that the model does not converge to high confidence/validated, until after the phase lag becomes nominal in the RUL (i.e., filtered RUL is consistent with raw RUL).
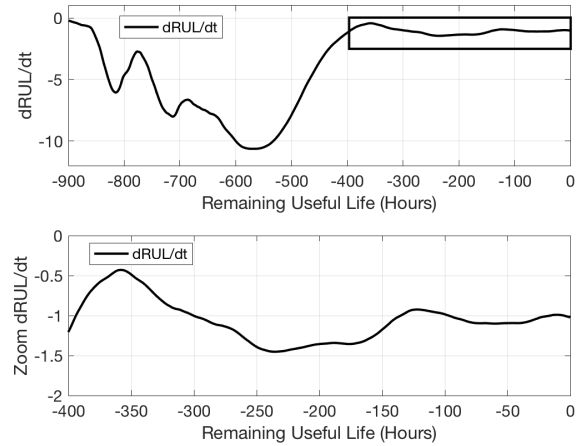

Figure 5 dRUL/dt

Ideally, the second derivative of RUL should be zero. The fact that it is not likely indicates that m, the crack growth exponent in (7) is not exactly 2.
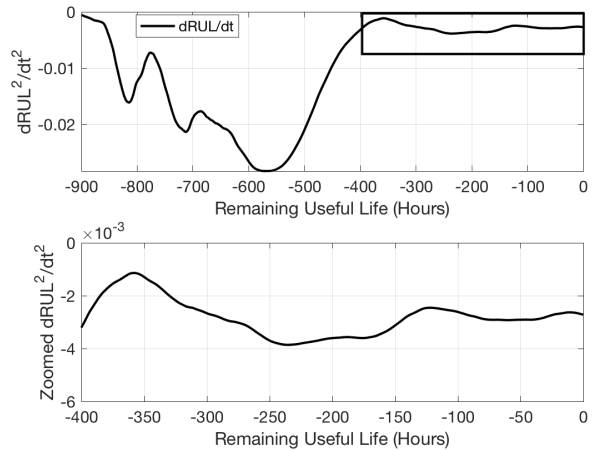

Figure 6 d^2RUL/dt^2

6. RULES FOR AUTOMATED REPORTING

Because an operators', pilots', or maintainers' time is constrained, only relevant and pertinent data should be alerted to draw their attention. As such, while the system may monitor many components, with the health and RUL being often updated, only data that is of concern (e.g., components in warning or alarm) and/or with valid RUL, should be displayed or alerted. A number of rules can then be developed which can automate this reporting. Some suggested rules would be:

- A valid/high confidence RUL is when the dRUL/dt is between 0 and -2 (near 1) and when the d^2RUL/dt^2 is small, with an absolute value less than 0.01.

- Display a warning alert when the HI is greater than 0.75 and the RUL is valid and less than some nominal alerting time, say 100 hours.

- Display an alarm alert if the HI is greater than 1.

This alert can be extinguished if acknowledged by the operator/maintainer/pilot, as per effective human factors engineering requirements.

As seen in Figures 5 and 6, the dRUL/dt and d^2RUL/dt^2 meet the criterion for a valid model approximately 450 hours before the HI of 1. That said, the warning alert does not trigger until 138 hours prior the HI of 1.0, as seen in Figure 7.

Note that given (10) and (11), other analyses are now possible. While the RUL is calculated based on the mean load measured over the prior sample period, the actual RUL is a distribution function, which is based on projected load (greater or lesser) and the measurement error of the HI. In Figure 7 confidence limits (high load health RUL of 105 hours, and low load health RUL 174 hours) are displayed, as based on the +/- one standard deviation of the estimated HI, and a +/- 10% in mean load.
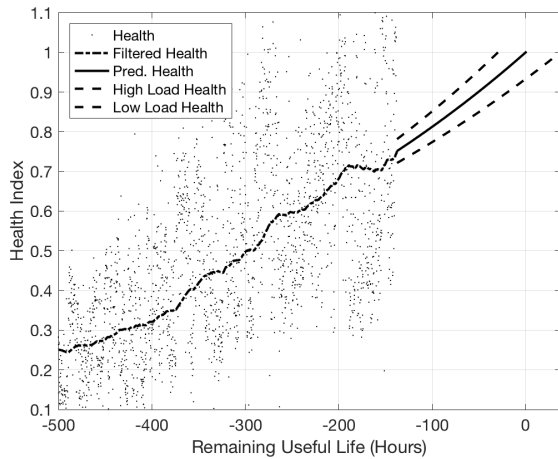

Figure 7 Warning and alert for valid RUL model with confidence limits

It can be shown that from any estimated RUL, the rate of change in the *HI* can be calculated as

$$\frac{dHI}{dt} = \frac{-log(HI)}{RUL\pi}. \qquad (27)$$

Then the estimate for any future HI_t is simply

$$HI_t = exp\left(t\pi\frac{dHI}{dt} + log(HI)\right) \qquad (28)$$

To estimate the confidence interval, it is a matter of accounting for the distribution of the *HI* and the change in load. For example, given that the one standard deviation for the HI is calculated as 0.03, and for a scenario where the

load is 10% higher than average (e.g., an aggressive flight regime), the future HI_t would be:

$$HI_t = exp\left(t\pi\frac{dHI}{dt}1.10 + log(HI + 0.03)\right) \qquad (29)$$

7. CONCLUSION

As has been demonstrated, simple α-β and α-β-γ trackers implementation on an embedded system allow estimation of the remaining useful life. The advantages of embedded RUL calculation is that the operator/maintainer of the asset would have decision support tools that allow them to protect the asset under analysis better. For a helicopter, this would allow the pilot, in the most extreme case, to land before mission completeness to avert a mishap. For other applications, it facilitates a proactive maintenance policy that ensures the reliability of the system.

The computation load of the analysis is 11 multiplications, nine additions, and a call to log, for each update. Given that for most systems, an acquisition is usually taken once every one to ten minutes, this can be implemented on even the most modest embedded system.

As of equal importance, methods for validating and reporting the RUL were demonstrated. Further, even for a computationally lightweight embedded system, the quality of the resulting analysis compares favorably with Kalman smoother or particle filter technique.

REFERENCES

Bar-Shalom, Multitarget-Multisensor Tracking: Application and Advances, Artech House, 1992

Bechhoefer, E., Bernhard, A., "A Generalized Process for Optimal Threshold Setting in HUMS" IEEE Aerospace Conference, MT, USA, March 2007.

Bechhoefer, E., He, D., Dempsey, P., "Gear Health Threshold Setting Based On a Probability of False Alarm," Conference of the Prognostics and Health Management Society, 2011.

Bond Helicopter Eurocopter AS332 crash, 2009, https://en.wikipedia.org/wiki/2009_Bond_Helicopters_ Eurocopter_AS332_crash

Byington, C., Safa-Bakhsh, R., Watson, M., Kalgren, P., (2003), Metrics Evaluation and Tool Development for Health and Usage Monitoring System Technology, HUMS 2003 Conference, DSTO-GD-0348

CHC Helicopter Service Flight 241, 2015, https://en.wikipedia.org/wiki/CHC_Helikopter_Service _Flight_241

CHC Helicopter, S-92a, Dec 28, 2016, https://aviation-safety.net/wikibase/wiki.php?id=192590

Cougar Helicopter Flight 91, 2009, https://en.wikipedia.org/wiki/Cougar_Helicopters_Fligh t_91

McCall, M, Stephan, D, Lackery, L, *ADS-79D Aeronautical Design Standard Handbook*, US Army Aviation and Missile Research, Development and Engineering Center, 2013

Paris, P., Gomez, M. and Anderson, W. A rational analytic theory of fatigue. The Trend in Engineering, 1961, 13: p. 9-14.

Vercer, P., Kreidl, M., Smit, R, "Condition Indicators for Gearbox Condition Monitoirng Systems", Acta Polytechnica Vol. 45, No. 6/2005

BIOGRAPHIES

Eric Bechhoefer received his B.S. in Biology from the University of Michigan, his M.S. in Operations Research from the Naval Postgraduate School, and a Ph.D. in General Engineering from Kennedy Western University. He is a former Naval Aviator who has worked extensively on condition-based maintenance, rotor track and balance, vibration analysis of rotating machinery and fault detection in electronic systems. Dr. Bechhoefer is a board member of the Prognostics Health Management Society and a member of the IEEE Reliability Society.

Rune Schlanbusch received his MSc in Space Technology from Narvik University College (NUC), Norway in 2007, and a Ph.D. in Engineering Cybernetics from NTNU, Norway in 2012. He currently holds positions as Senior Researcher at Teknova, Norway. His principal research interests include nonlinear control theory and stability analysis, Multiphysics modeling and simulation, condition monitoring and vibration and acoustic data analysis, intelligent sensor technologies and drone technology. He is currently a member of the IEEE since 2008, Control Systems Society and Robotics & Automation Society.