1-1-2008

A Comparison of Filtering Approaches for Aircraft Engine Health Estimation

Daniel J. Simon
Cleveland State University, d.j.simon@csuohio.edu

Original Citation

Simon, D. (2008). A comparison of filtering approaches for aircraft engine health estimation. Aerospace Science and Technology, 12(4), 276-284.

A comparison of filtering approaches for aircraft engine health estimation

Dan Simon

Cleveland State University, Department of Electrical Engineering, Stilwell Hall Room 332, 2121 Euclid Avenue, Cleveland, OH 44115, USA

1. Introduction

The application considered in this paper is aircraft turbofan engine health parameter estimation [1]. Health parameters represent engine component efficiencies and flow capacities. The performance of a gas turbine engine deteriorates over time. This deterioration reduces the fuel economy of the engine. Airlines periodically collect engine data in order to evaluate the health of the engine and its components. The health evaluation is then used to determine maintenance schedules. Reliable health evaluations are used to anticipate future maintenance needs. This offers the benefits of improved safety and reduced operating costs. The money-saving potential of such health evaluations is substantial, but only if the evaluations are reliable. The data used to perform health evaluations are typically collected during flight and later transferred to ground-based computers for post-flight analysis. Data are collected each flight at the same engine operating points and corrected to account for variability in ambient conditions. Various algorithms have been proposed to monitor engine health, such as weighted least squares [2], expert systems [3], Kalman filters and neural networks [4], fuzzy logic [5], and genetic algorithms [6].

Kalman filter based approaches seem to be the most commonly used methods for aircraft engine health estimation, but up to this point in time a systematic comparison of these techniques has not been presented. This paper gives a comparison of the estimation accuracy and computational effort of various Kalman filter based approaches to aircraft engine health estimation. Note that smoothing does not provide any improvement over filtering because the health parameters that we estimate are modeled as constant biases [7].

We emphasize that in this paper we are confining the problem to the estimation of engine health parameters in the presence of degradation only. There are specific engine faults that can result in abrupt shifts in filter estimates, possibly even indicating an apparent improvement in some engine components. An actual engine performance monitoring system would need to include additional logic to detect and isolate such faults, as discussed in [8].

This paper is organized as follows. Section 2 presents a review of the LKF, the EKF, and the UKF. Section 3 discusses

E-mail address: d.j.simon@csuohio.edu.

the problem of turbofan health parameter estimation, along with the dynamic model that we use in our simulation experiments. Although the health parameters are not state variables of the model, the linearized dynamic model is augmented in such a way that a Kalman filter can estimate the health parameters as shown in previous publications [9,10]. Section 4 presents some simulation results based on a nonlinear turbofan model. We see in this section that the EKF and UKF both estimate engine health significantly better than the LKF. However, the EKF requires computational effort that is an order of magnitude higher than the LKF, and the UKF requires computational effort this is yet another order of magnitude higher than the EKF. Section 5 presents some concluding remarks and suggestions for further work.

2. State estimation for nonlinear systems

In this section we first summarize the standard Kalman filter equations. We then review three extensions of the standard Kalman filter to nonlinear systems: the LFK, the EKF, and the UKF. Details can be found in [7].

2.1. The Kalman filter

Consider the discrete linear time-invariant system given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + w(k), \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{1}$$

where k is the time index, x is the state vector, u is the known control input, y is the measurement, and $\{w(k)\}$ and $\{v(k)\}$ are noise input sequences. The problem is to find an estimate $\hat{x}(k+1)$ of $x(k+1)$ given the measurements $\{y(0), y(1), \ldots, y(k)\}$. We assume that the following standard conditions are satisfied.

$$\begin{aligned} &E\big[x(0)\big] = \bar{x}(0) \\ &E\big[w(k)\big] = E\big[v(k)\big] = 0 \\ &E\big[\big(x(0) - \bar{x}(0)\big)\big(x(0) - \bar{x}(0)\big)^T\big] = P^+(0) \\ &E\big[w(k)w^T(m)\big] = Q\delta_{km} \\ &E\big[v(k)v^T(m)\big] = R\delta_{km} \\ &E\big[w(k)v^T(m)\big] = 0 \end{aligned} \tag{2}$$

where $E[\cdot]$ is the expectation operator, $\bar{x}$ is the expected value of x, and δ_{km} is the delta function. The Kalman filter equations are given by

$$\begin{aligned} P(k) &= AP^+(k-1)A^T + Q \\ K(k) &= AP(k)C^T\big(CP(k)C^T + R\big)^{-1} \\ \hat{x}(k) &= F\hat{x}^+(k-1) + Bu(k) \\ \hat{x}^+(k) &= \hat{x}(k) + K(k)\big(y(k) - C\hat{x}(k)\big) \\ P^+(k) &= \big(I - K(k)C\big)P(k) \end{aligned} \tag{3}$$

where the filter is initialized with $\hat{x}^+(0) = \bar{x}(0)$, and $P^+(0)$ given above. $\hat{x}^+(k)$ is the a posteriori state estimate at time k, and $P^+(k)$ is its covariance. $\hat{x}(k)$ is the a priori state estimate at time k, and $P(k)$ is its covariance. The Kalman filter is widely used both for its practical success and for its attractive theoretical properties [7,11].

2.2. The linearized Kalman filter

Now suppose that we have the nonlinear system model

$$\begin{aligned} x(k+1) &= f\big(x(k), u(k), k\big) + w(k) \\ y(k) &= h\big(x(k), k\big) + v(k) \end{aligned} \tag{4}$$

where $f(\cdot)$, and $h(\cdot)$ are general nonlinear functions. We use Taylor series to expand these equations around a nominal control $\bar{u}(k)$, a nominal state $\bar{x}(k)$, and a nominal output $\bar{y}(k)$. This gives the following approximately correct linear system.

$$\begin{aligned} \Delta x(k+1) &= A\Delta x(k) + B\Delta u(k) + w(k) \\ \Delta y(k) &= C\Delta x(k) + v(k) \end{aligned} \tag{5}$$

The Δ quantities in the above equations are defined as deviations from the nominal trajectory: $\Delta x = x - \bar{x}$, $\Delta u = u - \bar{u}$, and $\Delta y = y - \bar{y}$. We assume that the control $u(k)$ is known perfectly so that $\Delta u(k) = 0$. The matrices on the right side of (5) are given as

$$\begin{aligned} A &= \frac{\partial f}{\partial x} \\ B &= \frac{\partial f}{\partial u} \\ C &= \frac{\partial f}{\partial u} \end{aligned} \tag{6}$$

and all partial derivatives are evaluated at the nominal state, control, and noise values. These matrices are called Jacobians. Now we can use a Kalman filter to estimate the deviation $\Delta x(k)$ of the state from its nominal value. The LKF is therefore given as

$$\begin{aligned} P(k) &= AP^+(k-1)A^T + Q \\ K(k) &= AP(k)C^T\big(CP(k)C^T + R\big)^{-1} \\ \Delta\hat{x}(k) &= A\Delta\hat{x}^+(k-1) + B\Delta u(k) \\ \Delta y(k) &= y(k) - \bar{y}(k) \\ \Delta\hat{x}^+(k) &= \Delta\hat{x}(k) + K(k)\big(\Delta y(k) - C\Delta\hat{x}(k)\big) \\ \hat{x}^+(k) &= \bar{x}(k) + \Delta\hat{x}^+(k) \\ P^+(k) &= \big(I - K(k)C\big)P(k) \end{aligned} \tag{7}$$

Jacobian calculations are performed as often as required in order to give the desired tradeoff between computational effort and filtering accuracy. The effort for a Jacobian calculation for a nonanalytic system depends on the specific system. First, a simulation with the states set equal to the current estimates needs to be run until the system reaches steady state. Then a series of short simulations needs to be run, one for each component of the control, state, and health parameter vectors. The perturbations (from nominal) in the state derivatives and outputs need to be measured for each individual control, state, and health parameter perturbation. The ratios of these perturbations are then used to obtain the A, B, and C matrices [12].

2.3. The extended Kalman filter

The LKF summarized in the preceding section is based on linearizing the nonlinear system around a nominal state trajectory. Since the Kalman filter estimates the state of the system, we can use the Kalman filter estimate as the nominal state trajectory. This is a boot strap approach. We linearize the nonlinear system around the Kalman filter estimate, and the Kalman filter estimate is based on the linearized system. This is the idea of the EKF. The EKF for the nonlinear system of (4) starts with the following time update equations.

$$P(k) = AP^{+}(k-1)A^{T} + Q$$
$$\hat{x}(k) = f\big(\hat{x}^{+}(k-1), u(k-1), k-1\big) \tag{8}$$

The Jacobian in the preceding equation is given as

$$A = \frac{\partial f(\hat{x}^{+}(k-1), u(k-1), k-1)}{\partial x} \tag{9}$$

Next the EKF performs the following measurement update equations.

$$K(k) = P(k)C^{T}\big(CP(k)C^{T} + R\big)^{-1}$$
$$\hat{x}^{+}(k) = \hat{x}(k) + K(k)\big[y(k) - h\big(\hat{x}(k), k\big)\big]$$
$$P^{+}(k) = \big(I - K(k)C\big)P(k) \tag{10}$$

where the Jacobian is given as

$$C = \frac{\partial h(\hat{x}(k), k)}{\partial x} \tag{11}$$

As with the LKF, the estimation accuracy and computational effort of the EKF increase with the frequency of the Jacobian calculations. Most of the computational effort is due to the system simulation that is required to obtain $f(\hat{x}, u, k)$ in (8), and $h(\hat{x}(k), k)$ in (10), and the system simulations that are required to obtain the Jacobians A and C.

2.4. The unscented Kalman filter

The EKF discussed in the previous section is the most widely applied state estimation algorithm for nonlinear systems. However, the EKF is notoriously difficult to tune and often gives unreliable estimates if the system or measurement nonlinearities are severe. This is because the EKF relies on linearization to propagate the mean and covariance of the state.

An unscented transformation is based on two fundamental principles [13]. First, it is easier to perform a nonlinear transformation on a single point rather than an entire pdf (probability distribution function). Second, it is not too hard to find a set of individual points in state space whose sample pdf approximates the true pdf of a state vector.

Taking these two ideas together, suppose that we know the mean $\bar{x}(k)$ and covariance $P(k)$ of a vector $x(k)$. The UKF (unscented Kalman filter) finds a set of deterministic vectors called *sigma points* whose sample mean and covariance are equal to $\bar{x}(k)$ and $P(k)$. We then apply our known nonlinear system function $f(x, u, k)$ to each deterministic vector to obtain transformed vectors. The sample mean and covariance of the transformed vectors will give a good estimate of the true mean and covariance of $x(k+1)$. The UKF can be summarized as follows [14,15].

1. We have an n-state discrete time nonlinear system given by

$$x(k+1) = f(x, u, k) + w(k)$$
$$y(k) = h(x, k) + v(k) \tag{12}$$

where $w(k)$ and $v(k)$ are zero mean, independent random noise processes with covariances Q and R respectively.

2. The UKF is initialized as follows.

$$\hat{x}^{+}(0) = E\big[x(0)\big]$$
$$P^{+}(0) = E\big[\big(x(0) - \hat{x}^{+}(0)\big)\big(x(0) - \hat{x}^{+}(0)\big)^{T}\big] \tag{13}$$

3. The following time update equations are used to propagate the state estimate and covariance from one measurement time to the next.
   (a) Choose the following sigma points

$$\hat{x}^{(i)}(k-1) = \hat{x}^{+}(k-1) + \tilde{x}^{(i)}, \quad i = 1, \ldots, 2n$$
$$\tilde{x}^{(i)} = \big(\sqrt{nP^{+}(k-1)}\big)_i, \quad i = 1, \ldots, n$$
$$\tilde{x}^{(n+i)} = -\big(\sqrt{nP^{+}(k-1)}\big)_i, \quad i = 1, \ldots, n \tag{14}$$

   $\sqrt{A}$ refers to the square root of the matrix A. That is, $(\sqrt{A})^{T}\sqrt{A} = A$, and $(\sqrt{A})_i$ refers to the ith row of $\sqrt{A}$. (The Cholesky factorization routine in Matlab can be used to find a matrix square root.)
   (b) Use the known nonlinear system equation to transform the sigma points into $\hat{x}^{(i)}(k)$ vectors as follows.

$$\hat{x}^{(i)}(k) = f\big(\hat{x}^{(i)}(k-1), u(k-1), k-1\big) \tag{15}$$

   (c) Combine the $\hat{x}^{(i)}(k)$ vectors to obtain the a priori state estimate.

$$\hat{x}(k) = \frac{1}{2n}\sum_{i=1}^{2n}\hat{x}^{(i)}(k) \tag{16}$$

   (d) Estimate the a priori estimation error covariance as follows.

$$P(k) = \frac{1}{2n}\sum_{i=1}^{2n}\big(\hat{x}^{(i)}(k) - \hat{x}(k)\big)\big(\hat{x}^{(i)}(k) - \hat{x}(k)\big)^{T} + Q \tag{17}$$

4. Implement the following measurement update equations.
   (a) Choose sigma points $\hat{x}^{(i)}(k)$ as follows.

$$\hat{x}^{(i)}(k) = \hat{x}(k) + \tilde{x}^{(i)}, \quad i = 1, \ldots, 2n$$
$$\tilde{x}^{(i)} = \big(\sqrt{nP(k)}\big)_i, \quad i = 1, \ldots, n$$
$$\tilde{x}^{(n+i)} = -\big(\sqrt{nP(k)}\big)_i, \quad i = 1, \ldots, n \tag{18}$$

   (b) Use the known nonlinear measurement equation to transform the sigma points into $\hat{y}^{(i)}(k)$ vectors (predicted measurements) as follows.

$$\hat{y}^{(i)}(k) = h\big(\hat{x}^{(i)}(k), k\big) \tag{19}$$

(c) Combine the $\hat{y}^{(i)}(k)$ vectors to obtain the predicted measurement.

$$\hat{y}(k) = \frac{1}{2n}\sum_{i=1}^{2n}\hat{y}^{(i)}(k) \tag{20}$$

(d) Estimate the covariance of the predicted measurement as follows.

$$P_y(k) = \frac{1}{2n}\sum_{i=1}^{2n}\big(\hat{y}^{(i)}(k) - \hat{y}(k)\big)\big(\hat{y}^{(i)}(k) - \hat{y}(k)\big)^T + R \tag{21}$$

(e) Estimate the covariance between $\hat{x}(k)$ and $\hat{y}(k)$ as follows.

$$P_{xy}(k) = \frac{1}{2n}\sum_{i=1}^{2n}\big(\hat{x}^{(i)}(k) - \hat{x}(k)\big)\big(\hat{y}^{(i)}(k) - \hat{y}(k)\big)^T \tag{22}$$

(f) The measurement updates are performed as follows.

$$\begin{aligned} K(k) &= P_{xy}(k)P_y^{-1}(k) \\ \hat{x}^+(k) &= \hat{x}(k) + K(k)\big(y(k) - \hat{y}(k)\big) \\ P^+(k) &= P(k) - K(k)P_y(k)K^T(k) \end{aligned} \tag{23}$$

It can be shown that the EKF estimate of the state matches the true mean of the state correctly up to the first order, but the UKF estimate is correct up to the third order. It can also be shown that both the EKF and the UKF approximate the covariance of the state estimate up to the third order. However, the error of the UKF approximation of the covariance is generally smaller than that of the EKF [14].

The UKF equations look more difficult than the LKF or the EKF. However, the UKF does not require any Jacobian calculations. Jacobian calculations of nonanalytic systems are often prone to numerical difficulties. The biggest computational difficulty of the UKF is the matrix square root that is required, and the system simulations of (15) and (19).

If computational effort is a primary consideration, then a smaller number of sigma points can be chosen. The above algorithm uses $2n$ sigma points, where n is the size of the state vector. The spherical UKF was developed with the goal of balancing computational savings and numerical stability [14,16] and uses only $(n+2)$ sigma points.

Another way to reduce the computational effort of the UKF is to skip the time update equation for the sigma points and simply use the most recent sigma points in the succeeding equations. This is an ad-hoc modification of the UKF that saves a lot of computational effort at the expense of the theoretical integrity of the filter. However, the measurement $y(k)$ is still used at each time step to update the a posteriori sigma points as shown in (23), so this approximation may not result in too much degradation of the filter performance. This is especially true for the case of health parameter estimation because the health parameters are modeled as constant biases (even though the rest of the state vector is still modeled as time varying).

3. Turbofan engine health monitoring

Fig. 1 shows a schematic representation of a turbofan engine [17]. A single inlet supplies airflow to the fan. Air leaving the fan separates into two streams: one stream passes through the engine core, and the other stream passes through the annular bypass duct. The fan is driven by the low pressure turbine. The air passing through the engine core moves through the compressor, which is driven by the high pressure turbine. Fuel is injected in the main combustor and burned to produce hot gas for driving the turbines. The two air streams combine in the augmentor duct, where additional fuel is added to further increase the air temperature. The air leaves the augmentor through the nozzle, which has a variable cross section area.

The simulation used in this paper is a software package called MAPSS (Modular Aero Propulsion System Simulation). In this section we summarize the model and the linearization process without going into the details that are provided elsewhere [17,18]. MAPSS is written using Matlab Simulink. The MAPSS engine model is based on a low frequency, transient,

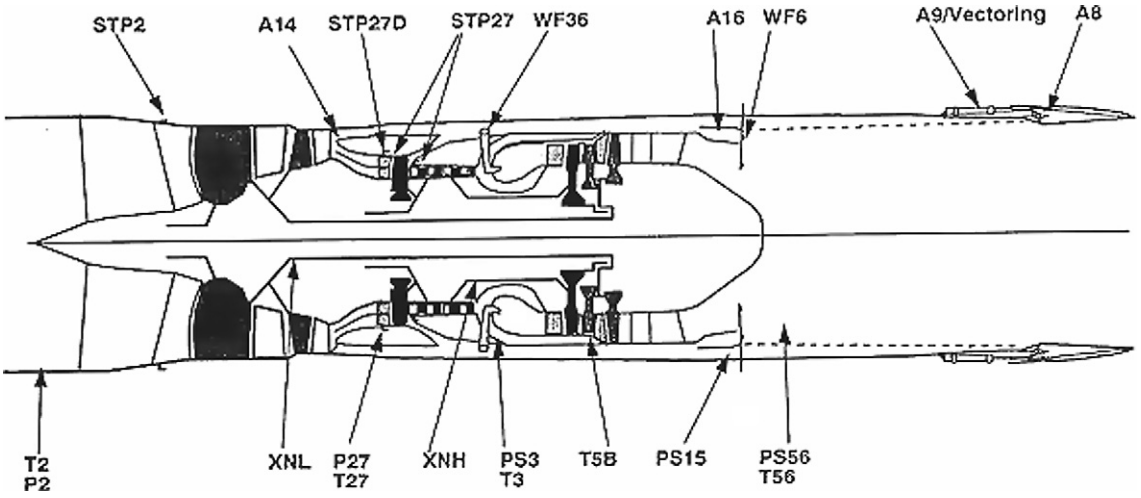

Fig. 1. Schematic representation of a turbofan engine.

Table 1
MAPSS turbofan model states and nominal values

State	Nominal value
LPT rotor speed	7264 RPM
HPT rotor speed	12 152 RPM
Average hot section metal temperature	1533° R

Table 2
MAPSS turbofan model controls and nominal values

Control	Nominal value
Main burner fuel flow	2454 lbm/hr
Variable nozzle area	343 in^2
Rear bypass door variable area	154 in^2

Table 3
MAPSS turbofan model health parameters and nominal values. Booster tip efficiency would normally be an additional health parameter, but it is not yet implemented in MAPSS

Health parameter	Normalized value
Fan airflow capacity	1
Fan efficiency	1
Booster tip airflow capacity	1
Booster hub airflow capacity	1
Booster hub efficiency	1
High pressure turbine airflow capacity	1
High pressure turbine efficiency	1
Low pressure turbine airflow capacity	1
Low pressure turbine efficiency	1

performance model of a high-pressure ratio, dual-spool, low-bypass, military-type, variable cycle, turbofan engine with a digital controller. The controller update rate is 50 Hz, and the component level model balances the mass/energy equations of the system at a rate of 2500 Hz. The three state variables used in MAPSS are low-pressure rotor speed, high-pressure rotor speed, and the average hot section metal temperature (measured from aft of the combustor to the high pressure turbine).

The discretized time invariant equations that model the turbofan engine can be summarized as follows.

$$\begin{aligned} x(k+1) &= f\big[x(k), u(k), p(k)\big] + w_x(k) \\ p(k+1) &= p(k) + w_p(k) \\ y(k) &= g\big[x(k), u(k), p(k)\big] + v(k) \end{aligned} \tag{24}$$

where k is the time index, x is the 3-element state vector, u is the 3-element control vector, p is the 9-element health parameter vector, and y is the 9-element measurement vector. The noise terms and health parameter degradations are not modeled in MAPSS but have been added to the model for the problem studied in this paper. The health parameters change slowly over time. Between measurement times their deviations can be approximated by the zero mean noise $w_p(k)$ (although in our study the health parameters only changed once per flight). The noise term $w_x(k)$ represents inaccuracies in the system model, and $v(k)$ represents measurement noise. A Kalman filter can be used with (24) to estimate the state vector x and the health parameter vector p. Since the system model is not available in analytical form, the Jacobian calculations need to be performed numerically. See [12] for Jacobian calculation details and trade-offs for the turbofan health estimation problem.

Table 4
MAPSS turbofan model measurements, nominal values, and signal-to-noise ratios. SNR is defined here as the nominal measurement value divided by one standard deviation of the measurement noise

Measurement	Nominal value	SNR
LPT exit pressure	19.33 psia	100
LPT exit temperature	1394° R	100
Percent low pressure spool rotor speed	63.47%	150
HPC inlet temperature	580.8° R	100
HPC exit temperature	965.1° R	200
Fan exit pressure	17.78 psia	200
Booster inlet pressure	20.19 psia	200
HPC exit pressure	85.06 psia	100
Core rotor speed	12 152 RPM	150

For systems with constant parameters appended to the state vector, the minimum number of observations required to achieve system observability is equal to the number of constant parameters [19,20]. Although the health parameters are not truly constant as seen in (24), they are modeled with infinite time constants and small amounts of artificial process noise. So since we want to estimate nine health parameters, we need at least nine measurements.

The states, controls, health parameters, and measurements are summarized in Tables 1–4, along with their values at the nominal operating point considered in this paper, which is a power lever angle of 21° at sea level static conditions (zero altitude and zero Mach). Table 4 also shows typical signal-to-noise ratios for the measurements, based on NASA experience and previously published data [21]. Sensor dynamics are assumed to be high enough bandwidth that they can be ignored in the dynamic equations. In Tables 1–4 we use the acronyms LPT for Low Pressure Turbine, HPT for High Pressure Turbine, LPC for Low Pressure Compressor, and HPC for High Pressure Compressor.

4. Simulation results

We simulated the filtering methods discussed in this paper using Matlab. We measured a steady state three second burst of open-loop engine data at 100 Hz during each flight. These routine data collections were performed over 50 flights at the single operating point shown in Tables 1, 2, and 4. The engine's health parameters were initialized to the values shown in Table 3 and then deteriorated a small amount once each flight (i.e., once every 300 time steps). The signal-to-noise ratios were determined on the basis of NASA experience and previously published data [21] and are shown in Table 4. In the Kalman filters we used a one-sigma state process noise equal to 0.005% of the nominal state values to allow the filter to be responsive to changes in the state variables. We also set the one sigma process noise for each component of the health parameter to a small percentage of the nominal parameter value. The values that we used were obtained by tuning. They were small enough to give reasonably smooth estimates, and large enough to allow the fil-

Table 5
Health parameter estimation errors (percent) and standard deviations of the LKF and EKF, averaged over all flights and all health parameters. The estimation error is measured as $|(p - \hat{p})/p_f|$, where p is the true health parameter value, $\hat{p}$ is the estimated health parameter value, and p_f is the health parameter value at the end of the simulation

	Number of Jacobian calculations					
	1	2	4	8	17	50
Linearized Kalman filter	5.7 ± 1.2	4.8 ± 0.9	3.5 ± 0.7	3.3 ± 0.8	3.7 ± 1.0	3.9 ± 1.0
Extended Kalman filter	2.9 ± 0.6	3.2 ± 1.1	2.9 ± 0.5	3.0 ± 0.9	2.5 ± 0.7	2.7 ± 0.7

ter to track slowly time-varying parameters. Although a number of approaches have been proposed for covariance tuning in the Kalman filter [7], our results were obtained with simple ad-hoc manual tuning.

For each simulation and each health parameter, we generated a random number $p_{im}(N)$ from a uniform distribution between 1% and 4%, where i is the health parameter number (between 1 and 9), m is the simulation run number, and N is the index of the final flight. We then simulated a linear-plus-exponential degradation of the health parameter such that the final health parameter value was $p_{im}(N)$. The initial health parameter estimation errors were zero. These health parameter degradation profiles were therefore random but were representative of turbofan performance data reported in the literature [22]. The health parameter degradation at flight k can be written as

$$p_{im}(k) = \frac{p_{im}(N)}{e^{-N/150} - 1 - N/600}\left(e^{-k/150} - 1 - k/600\right) \quad (25)$$

This gives a $p_{im}(k)$ profile that looks mostly exponential early in the engine's service cycle (small values of k) and looks mostly linear later in the engine's service cycle (large values of k).

4.1. Performance results

We ran between 20 and 60 Monte Carlo simulations for each filter, depending on how long it took the variance of the results to reach steady state (as graphically observed). Each simulation consisted of 50 flights, health parameter degradations with random magnitudes, and different random measurement noise. Tables 5 and 6, along with Figs. 2–4, show the average performance of the filters.

We can make some interesting observations from the tables and the figures. Table 5 and Fig. 2 shows that the performance of the LKF steadily improves as the frequency of the Jacobian calculations increases. This improvement continues until the Jacobian calculations are performed once every three flights (i.e., 17 Jacobian calculations over 50 flights). There does not appear to be any improvement if the frequency of the Jacobian calculations increases to more than once every three flights.

Table 5 and Fig. 3 show improvement in the performance of the EKF as the frequency of the Jacobian calculations increases. However, the improvement is not as large as with the LKF. This is because the LKF is more approximate than the EKF, so there is more room for improvement in the performance of the LKF.

Table 6 and Fig. 4 show that the performance of the UKF is independent of the number of sigma points, and independent of the number of sigma point updates per time step. Theoretically, the UKF with the full set of sigma points and two sigma point updates per time step should outperform the other UKFs. However, in practice we do not see any improvement. After using the near-minimum number $(n + 2)$ of spherical sigma points, the use of additional sigma points exceeds the point of diminishing returns due to the relatively mild nonlinearities of the turbofan health estimation problem.

Table 6
Health parameter estimation errors (percent) of the unscented Kalman filters, averaged over all flights and all health parameters. The estimation error is measured as $|(p - \hat{p})/p_f|$, where p is the true health parameter value, $\hat{p}$ is the estimated health parameter value, and p_f is the health parameter value at the end of the simulation

Number of sigma points	Number of sigma point updates per time step	
	1	2
$n + 2$	2.7 ± 0.5	2.4 ± 0.4
$2n$	2.7 ± 0.6	2.7 ± 0.5

We can compare the linearized, extended, and unscented Kalman filters by comparing Tables 5 and 6, and Figs. 2–4. We see that the EKF clearly outperforms the LKF, although the improvement in performance becomes less dramatic as the number of Jacobian calculations increases. The performance of the UKF appears to be about the same as that of the EKF.

Note that the Kalman filter works well only if the assumed system model matches reality fairly closely. The method presented in this paper, by itself, will not work well if there are large sensor biases or hard faults due to severe component failures. A mission-critical implementation of a Kalman filter should always include some sort of additional residual check to verify the validity of the Kalman filter results [23], particularly for the application of turbofan engine health estimation considered in this paper [1].

4.2. Computational effort

Now we consider the computational effort of the filters. A simulation of the MAPSS software discussed here requires about 30 s of computational effort on a 1.5 GHz PC with 256 MB of RAM. This simulates 4 s of aircraft engine dynamics, which is long enough to allow the engine to reach steady state after a small change in the health parameters. A Jacobian calculation requires about 90 s of computational effort. This involves the simulation of 4 s of aircraft engine dynamics, along with 16 short 0.1 s simulations to obtain the perturbations that are used to generate the Jacobians.

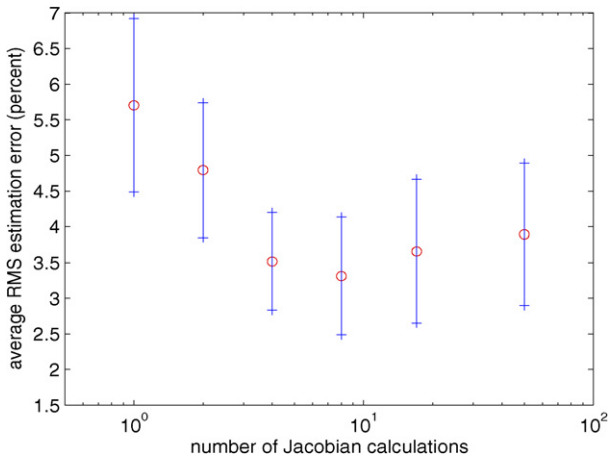

Fig. 2. Linearized Kalman filter estimation errors and standard deviations.

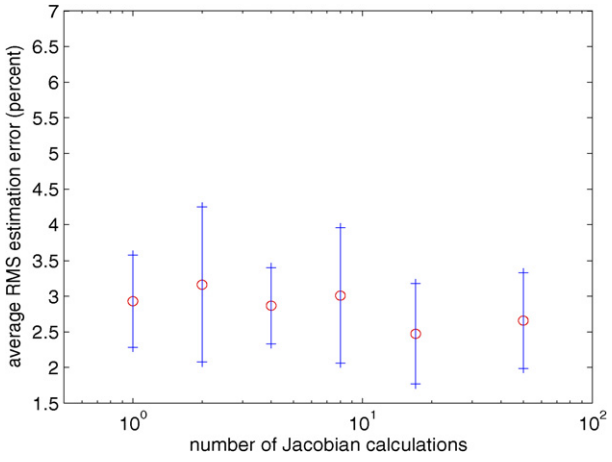

Fig. 3. Extended Kalman filter estimation errors and standard deviations.

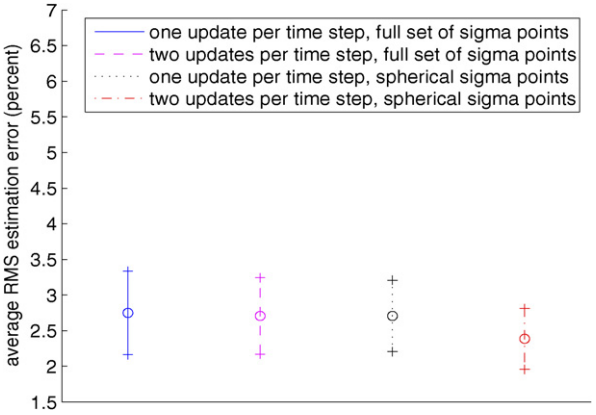

Fig. 4. Unscented Kalman filter estimation errors and standard deviations.

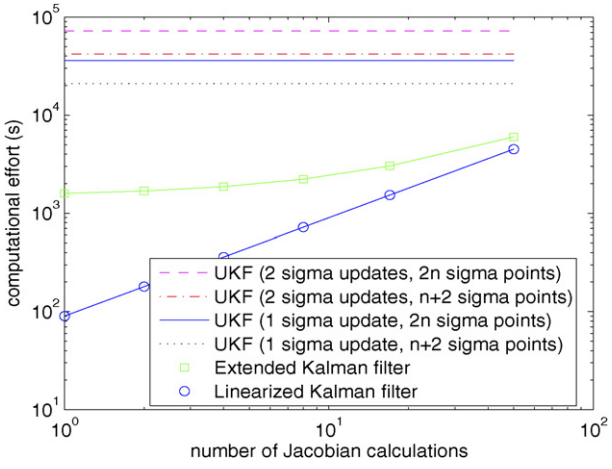

Fig. 5. Computational effort of the linearized, extended, and unscented Kalman filters. Note that the "number of Jacobian calculations" is not relevant for the UKF.

The computational effort of the LKF is dominated by the effort required for Jacobian calculations, and is therefore equal to the number of Jacobian calculations multiplied by 90 s. This can be written as

$$E_L = 90N_j \tag{26}$$

where N_j is the number of Jacobian calculations.

The EKF requires one system simulation plus Jacobian calculations. For our 50 flight simulation, the EKF therefore requires 50×30 s, plus the number of Jacobian calculations multiplied by 90 s. This can be written as

$$E_E = 1500 + 90N_j \tag{27}$$

The UKF can be implemented with $2n$ sigma points or $n+2$ sigma points, where $n = 12$ is the total number of states in the system (after augmentation of the nine health parameters to the three-element state vector). Furthermore, it can be implemented with either one or two sigma point updates per time step. For our 50 flight simulation with a 30 s simulation time, the UKF therefore requires

$$E_u = 1500N_u N_\sigma \tag{28}$$

where N_u is the number of sigma point updates per time step (either 1 or 2) and N_σ is the number of sigma points (either 14 or 24).

Fig. 5 summarizes the computational effort of the linearized, extended, and unscented Kalman filters. It is seen that the LKF has the lowest computational effort, and the effort grows linearly with the number of Jacobian calculations. The EKF has an effort that is an order of magnitude larger than the LKF, although the efforts of the two filters get closer as the number of Jacobian calculations increases and the extra simulations required by the EKF becomes less dominant. The UKF has a computational effort that is another order of magnitude larger than the EKF.

4.3. Discussion

We have seen that the EKF and UKF provide similar performance and both outperform the LKF. More frequent Jacobian calculations improve the performance of the LKF and EKF but also increase computational effort. Various approximations that are applied to the UKF to decrease computational effort do not degrade the performance of the UKF.

The EKF requires computational effort that is an order of magnitude higher than the LKF, and the UKF requires compu-

tational effort that is yet another order of magnitude higher than the EKF. Overall it appears that the EKF is the best choice for aircraft engine health parameter estimation, with Jacobian calculations about every three flights. This is our major conclusion and recommendation based on our simulation results.

Now we discuss the reasons for the observed results. First, we know from our system model that an aircraft engine is highly nonlinear. This indicates that the EKF and UKF should outperform the LKF. For highly nonlinear systems we would also expect the UKF to perform better than the EKF. However for the aircraft engine system the nonlinearities are not severe enough, nor are the health parameter deviations large enough, to cause the UKF to perform better than the EKF. This all indicates that the aircraft engine system is nonlinear, but not so nonlinear that extra computational effort (e.g., with a UKF or a computer intelligence based approach) is warranted. Once we get past the complexity of the EKF, we have reached a point of diminishing returns in our health estimation problem.

Similar conclusions can be reached relative to the frequency of Jacobian calculations. With the LKF more frequent Jacobian calculations gain better performance because of its relative inability to deal with nonlinearities. However with the EKF more frequent Jacobian calculations are probably not worth the effort because the EKF can handle the aircraft engine nonlinearities well enough already.

5. Conclusion

This paper has compared various Kalman filter based estimation approaches for the evaluation of aircraft engine health. The engine dynamics are nonlinear enough to warrant the use of an extended Kalman filter (EKF), but not so nonlinear as to justify the extra computational expense of an unscented Kalman filter (UKF). The nonlinearities are significant enough to justify EKF Jacobian calculations every three nights or so, but Jacobian calculations at a higher frequency are generally not worth the extra computational effort.

It is natural to consider the use of higher order linearization approaches to reduce the estimation errors that are due to nonlinearities. These approaches include the iterated EKF [24], the second order EKF [25], the Gaussian sum filter [26], the grid based filter [27, Chapter 6], and the more general particle filter [7]. However since the work presented in this paper indicates that the aircraft engine nonlinearities are mild enough that the UKF does not provide much better performance than the EKF, it is doubtful that these other higher order approaches will result in much improvement either.

Past work by the authors showed the advantages of constrained Kalman filtering for aircraft engine health estimation [18]. The present paper has not considered constrained Kalman filtering, but it would be interesting to see how the conclusions of this paper might change with the addition of state constraints.

Acknowledgements

This work was supported by the NASA Aviation Safety and Security Program at the NASA Glenn Research Center. The expertise of Donald L. Simon of the NASA Glenn Research Center, and the constructive criticism of the two anonymous journal reviewers, is gratefully acknowledged.

References

[1] D. Doel, TEMPER – A gas-path analysis tool for commercial jet engines, ASME Journal of Engineering for Gas Turbines and Power 116 (January 1994) 82–89.

[2] D. Doel, An assessment of weighted-least-squares-based gas path analysis, ASME Journal of Engineering for Gas Turbines and Power 116 (April 1994) 366–373.

[3] H. DePold, F. Gass, The application of expert systems and neural networks to gas turbine prognostics and diagnostics, ASME Journal of Engineering for Gas Turbines and Power 121 (October 1999) 607–612.

[4] A. Volponi, H. DePold, R. Ganguli, C. Daguang, The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study, ASME Journal of Engineering for Gas Turbine and Power 125 (October 2003) 917–924.

[5] R. Verma, N. Roy, R. Ganguli, Gas turbine diagnostics using a soft computing approach, Applied Mathematics and Computation 172 (January 2006) 1342–1363.

[6] T. Kobayashi, D.L. Simon, A hybrid neural network-genetic algorithm technique for aircraft engine performance diagnostics, in: 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, July 2001.

[7] D. Simon, Optimal State Estimation, John Wiley & Sons, New York, 2006.

[8] R. Verma, R. Ganguli, Denoising Jet Engine Gas Path Measurements Using Nonlinear Filters, IEEE/ASME Transactions on Mechatronics 10 (August 2005) 461–464.

[9] B. Friedland, Treatment of bias in recursive filtering, IEEE Transactions on Automatic Control AC14 (August 1969) 359–367.

[10] H. Lambert, A simulation study of turbofan engine deterioration estimation using Kalman filtering techniques, NASA TM 104233, June 1991.

[11] B. Anderson, J. Moore, Optimal Filtering, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[12] B. Endurthi, Linearization and health estimation of a turbofan engine, Masters Thesis, Cleveland State University, Cleveland Ohio, December 2004. Available online at http://academic.csuohio.edu/embedded/Publications/Thesis/Bharath_thesis.pdf.

[13] S. Julier, J. Uhlmann, H. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, IEEE Transactions on Automatic Control 45 (March 2000) 477–482.

[14] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, in: Proceedings of the IEEE 92, March 2004, pp. 401–422.

[15] E. Wan, R. van der Merwe, The unscented Kalman filter, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, John Wiley & Sons, New York, 2001.

[16] S. Julier, The spherical simplex unscented transformation, in: American Control Conference, 2003, pp. 2430–2434.

[17] K. Parker, K. Melcher, The modular aero-propulsion systems simulation (MAPSS) users' guide, NASA TM 2004-212968, March 2004.

[18] D. Simon, D.L. Simon, Kalman filtering with inequality constraints for turbofan engine health estimation, IEE Proceedings – Control Theory and Applications 153 (May 2006) 371–378.

[19] M. Espana, On the estimation algorithm for adaptive performance optimization of turbofan engines, in: AIAA/SAE/ASME/ASEE 29th Joint Propulsion Conference, June 1993.

[20] M. Espana, G. Gilyard, On the estimation algorithm for adaptive performance optimization of turbofan engines, NASA TM 4551, January 1994.

[21] W. Merrill, Identification of multivariable high-performance turbofan engine dynamics from closed-loop data, AIAA Journal of Guidance, Control, and Dynamics 7 (November 1984) 677–683.

[22] O. Sasahara, JT9D engine/module performance deterioration results from back to back testing, in: International Symposium on Air Breathing Engines, 1985, pp. 528–535.
[23] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.
[24] M. Athans, R. Wishner, A. Bertolini, Suboptimal state estimation for continuous-time nonlinear systems from discrete measurements, IEEE Transactions on Automatic Control AC-13 (October 1968) 504–514.
[25] R. Henriksen, The truncated second-order nonlinear filter revisited, IEEE Transactions on Automatic Control AC-27 (February 1982) 247–251.
[26] D. Alspach, H. Sorenson, Nonlinear Bayesian estimation using Gaussian sum approximations, IEEE Transactions on Automatic Control AC-17 (August 1972) 439–448.
[27] J. Spall (Ed.), Bayesian Analysis of Time Series and Dynamic Models, Marcel Dekker, New York, 1988.