ON-LINE PERFORMANCE MONITORING AND ENGINE DIAGNOSTIC USING ROBUST KALMAN FILTERING TECHNIQUES

Pierre Dewallef
Turbomachinery Group
Aerospace, Mechanics and Materials Department
Université de Liège
chemin des chevreuils, 1
4000 Liège - Belgium
Email: p.dewallef@ulg.ac.be

Olivier Léonard
Turbomachinery Group
Aerospace, Mechanics and Materials Department
Université de Liège
chemin des chevreuils, 1
4000 Liège - Belgium
Email: o.leonard@ulg.ac.be

ABSTRACT
In this contribution, an on-line engine performance monitoring is carried out through an engine health parameter estimation based on several gas path measurements. This health parameter estimation makes use of the analytical redundancy of an engine model and therefore implies the knowledge of the engine state. As the latter is a priori not known the second task is therefore an engine state variable estimation. State variables here designate working conditions such as inlet temperature, pressure, Mach number, rotational speeds, . . .

Estimation of the state variables constitutes a general application of the Extended Kalman Filter theory, while the health parameter estimation is a classical recurrent regression problem. Recent advances in stochastic methods [1] show that both problems can be solved by two Kalman filters working jointly. Such filters are usually named Dual Kalman Filters.

The present contribution aims at using a dual Kalman filter modified to provide robustness. This procedure should be able to cope with as much as 20 to 30% of faulty data. The resulting online method is applied to a turbofan model developed in the frame of the OBIDICOTE 1 project. Several tests are carried out to check the performance monitoring capability and the robustness that can be achieved.

INTRODUCTION
Nowadays turbine engine tests are processed in such a way that most of the measurements are performed during steady states. This is aimed to make the data processing as simple as possible, and corresponds to on-wing situations since the engine spends as much as 90% of the time in steady state cruise conditions.

In steady conditions the state of the engine can be measured or assessed with a relatively low uncertainty when compared to the gas path measurements. This uncertainty is therefore usually neglected when the health parameter estimation is performed. However even if steady state conditions are achieved sensor biases can spoil the results of the engine performance monitoring. This addresses the problem of the combined estimation of both engine state (the operating point) and engine health parameters when several “approximately known” operating conditions have to be dealt with.

Besides, waiting for steady state conditions to be achieved at the test bench is time consuming and costly and a method for performance assessment during transients would be of great interest, keeping in mind that the state variables of the engine cannot be measured accurately during transients due to heat transfers and sensor delays.

The foregoing considerations lead us to develop an engine performance monitoring method based on dual Kalman filtering for transient conditions and to treat steady state identification as a sub-case of transient identification.

1 A Brite/Euram project concerning On-Board Identification, Diagnosis and Control of Gas Turbine Engines
In this paper a short introduction of the linear Kalman filter is followed by the development of a robust form of this algorithm and the extension of the robust linear Kalman filter to non-linear models. The health parameter estimation through Kalman filtering techniques and the dual estimation problem is then addressed and finally some tests are carried out to underline strengths and weaknesses of robust dual Kalman filtering for performance monitoring and sensor fault detection.

**PROBLEM STATEMENT**

The “prediction-correction” structure of the Kalman filter represented in figure 1 is the basis of a measurement validation and diagnostics procedure.

![General "prediction-correction" structure of the Kalman filter](image)

**State variables** \( x_i \) is the minimal set of data which is sufficient to uniquely describe the unforced dynamical behaviour of the system (e.g. rotational speeds). The environment variables, defined as the set of external disturbances which depend only on their own past history but not on the present and past values of all other quantities observed on the system (e.g. flight Mach number, altitude, ambient temperature and pressure) have been embedded into the state variables. The time update equations also include the transition model of the environment variables.

**Command variables** \( u_i \) is the set of variables specified by the operator (e.g. fuel flow).

**Model parameters** \( w_i \) is the set of “supposed” constant parameters that characterize the model.

**Measurements** \( y_i \) is the set of observed variables that are not directly manipulated.

All the above variables but the command variables are statistical variables. In other words, a probability density function is attached to each of these variables representing their probability of occurrence. On the other hand the command variables are set by the operator with a probability of one. Tilded variables are estimated variables while untilded variables are related to raw variables.

Given any initial conditions on the state and environment variables, the time update equations determine the next state variables by a direct transition function or by integration of differential equations modelling the gas turbine. Based on these new state variables a simulation of the measurements can be realized through a measurement update equation system, providing predicted measurements \( \hat{y}_i \) based on the gas turbine model. Both time update and measurement update equations are supposed to have a known structure parameterized by the set of parameters \( w_i \). Predicted measurements are then compared to raw measurements to update the state variables according to the Kalman gain \( K \).

The turbofan used as an application is a two spool, mixed flow turbofan engine with 417 kg/s mass flow rate, 12500 daN gross thrust at take-off. The physical model of this engine was developed in the frame of the OBIDICOTE project founded by the European Community.

Even if this model is developed for flight conditions, only test bench conditions are considered herein as this is the context of our project. As a consequence flight Mach number and altitude are zero.

The model variables are summarized in figure 2, state variables as well as command variables are

| \( u_1 \) | \( W_{fuel} \) | Fuel flow |
| \( x_1 \) | \( ZT_{amb} \) | Ambient temperature |
| \( x_2 \) | \( ZP_{amb} \) | Ambient pressure |
| \( x_3 \) | \( NLP \) | Low pressure spool speed |
| \( x_4 \) | \( NHP \) | High pressure spool speed |
| \( x_5 \) | \( T_b^3 \) | high pressure compressor (HPC) blade temperature |
| \( x_6 \) | \( T_c^3 \) | high pressure compressor (HPC) casing temperature |
| \( x_7 \) | \( T_b^b \) | combustor casing temperature |
| \( x_8 \) | \( T^b_42 \) | high pressure turbine (HPT) blade temperature |
| \( x_9 \) | \( T_c^b_42 \) | high pressure turbine (HPT) casing temperature |

Some of these state variables are measured and are used to initialize the iterations performed by the filter. Others are not measured and need to be guessed from a priori knowledge. This is the case for casing and blade temperatures of HPC, HPT and combustor. The filter has to determine these state variables before being efficient.

The state update equations of the turbofan model are integrated in time using a five step Runge-Kutta algorithm. Health parameters are included into the model in order to simulate defective components. These health parameters are multiplying factor on efficiencies (SE12, SE26, SE3, SE42, SE49) in all the components of the engine. The original OBIDICOTE model also includes mass flow rate coefficients but they are not considered here because of a lack of information about their link with the efficiency coefficients (this aspect is discussed in the application).
FEATURES OF KALMAN FILTERS

Let us consider a linear update system of equations for the state variables and the measurements such as

\[
\begin{align}
x_{t+1} &= F_t x_t + B_t u_t + \mu_t \\
y_t &= H_t x_t + \epsilon_t.
\end{align}
\]

The linear Kalman filter performs the following recursion where the noises \(\epsilon_t\) and \(\mu_t\) follow an independent zero mean normal distribution with covariance matrix \(R_t\) and \(Q_t\) respectively and where process and measurement equations (i.e. the model of the engine) are linear. The procedure is detailed in algorithm 1.

1. The state variables \(\tilde{x}_t\) are predicted based on the preceding estimated state \(\hat{x}_{t-1}\) and the command variables \(u_t\). The state covariance matrix \(P_{xx,t-1}\) is projected in the direction \(F_t\) to give the a priori covariance matrix \(P_{xx,t}\).

2. The measurements \(\tilde{y}_t\) are predicted based on \(\tilde{x}_t\) and the measurement covariance matrix \(P_{yy,t}\) as well as the state-measurement covariance matrix \(P_{xy,t}\) are computed.

3. The state variables \(\hat{x}_t\) are calculated based on the prediction \(\tilde{x}_t\), the predicted measurements \(\tilde{y}_t\) and the raw measurements \(y_t\). The a posteriori state covariance matrix \(P_{xx,t}\) is then computed.

This solution is recursive as each updated estimate of the state is computed from the previous estimate and the new input data. Consequently only the previous estimate requires storage: there is no need to store the entire past observed data. Moreover the Kalman filter is computationally more efficient than calculating the estimation directly from the entire past observed data at each step of the filtering process. For more information see [1,2].

MAKING THE KALMAN FILTER ROBUST

The main drawback of the normal law is that it cannot cope with data far from the most likely value such as sensor faults. This robustness aspect is the main concern of our contribution.

During the last forty years, many researchers developed theories to robustify statistical estimators. Tukey, Hampel and Huber [3] defined the bases of robust statistics. One of the Huber’s main results is a probability density function for a location parameter \(\theta\) which assumes that measurement noise may be approximated by a gaussian law (normal noise) \(p_\theta(\epsilon)\) contaminated by another distribution \(p_\delta(\epsilon)\). This function is known as the \(\delta\) contaminated distribution \(p_\delta(\epsilon)\) and has already been applied by the authors [4]. If \(\epsilon(\theta) = \gamma - \tilde{y}(\theta)\) is the measurement noise and \(p(\epsilon)\) its probability distribution, the score function \(\rho(\epsilon) = -\log(p(\epsilon))\) and the gain function \(\psi(\epsilon) = \frac{d}{d\theta}\rho(\epsilon)(\frac{d\epsilon}{d\theta})^{-1}\) are defined as

\[
\begin{align}
\rho_\delta(\epsilon) &= (\delta - 1)p_\epsilon(\epsilon) + \delta p_\delta(\epsilon) \\
\psi_\delta(\epsilon) &= \frac{1}{\sigma^2} \max(-k\sigma, \min(k\sigma, \epsilon))
\end{align}
\]

The contamination level \(\delta\) controls the robustness of the estimation. \(\sigma\) is the standard deviation of the measurement noise and \(k\) is a threshold depending on \(\delta\) [3].

Later others such as Tsyokin [5] applied robust statistics to regression problems. Recursive methods based on stochastic approximations were adapted to robust filter by Tsyokin, Mar-

Algorithm 1 Linear Kalman filter

Require: \(x_0\) and \(P_0\) \{Initialization\}

while new measurements are performed do
\{State estimate propagation\}
\(\hat{x}_t = F_t \hat{x}_{t-1} + B_t u_t\)
\{Error covariance projection in the direction \(F_t\)\}
\(P_{xx,t} = F_t P_{xx,t-1} F_t^T + Q_t\)
\{Kalman gain matrix computation\}
\(P_{yy,t} = P_{yy,t} - K_t P_{yy,t} P_{yy,t}^T\)
\{Measurement update propagation\}
\(\tilde{y}_t = H_t \tilde{x}_t\)
\{State estimate update\}
\(\hat{x}_t = \tilde{x}_t + K_t (y_t - \tilde{y}_t)\)
\{Error covariance update\}
\(P_{xx,t} = (I - K_t H_t) P_{xx,t}\)
end while
A robust form of Kalman filter may be defined by finding the state \( x_t \) that maximizes the posterior state probability which can be found by solving the following equation:

\[
\frac{\partial}{\partial x_t} \left( p(y_t|x_t)p(x_t|y^{-1}w) \right) = 0
\]

Equation (6) can be solved using a weighted least squares approach that robustifies the filter [3]. The weighting matrix \( S_t \) is defined as the diagonal matrix

\[
S_t(i,i) = R_t(i,i) \frac{\psi(r_t(i))}{r_t(i)} \text{ with } r_t = y_t - H_t x_t.
\]

Such a weighted least square method has already been used in [9] for a steady-state, batch (the data are treated by batch), robust engine performance monitoring. Using \( S_t \), equation (6) yields

\[
0 = (P_{xx,t})^{-1}(x_t - \hat{x}_t) + H_t^T S_t^{-1}(y_t - H_t x_t) \Leftrightarrow
\]

\[
x_t = \hat{x}_t + (P_{xx,t})^{-1} + H_t^T S_t^{-1} R_t H_t \left[ H_t^T S_t R_t H_t + S_t^{-1} R_t \right]^{-1} H_t^T S_t R_t H_t \hat{x}_t
\]

with \( K_t \) and \( r_t^{-} \) defined by

\[
K_t = P_{xx,t} H_t^T \left[ H_t P_{xx,t} H_t^T + S_t^{-1} R_t \right]^{-1}
\]

\[
r_t^{-} = y_t - H_t x_t
\]

The \textit{a posteriori} state covariance matrix \( P_{xx,t} \) may be assessed by using \( K_t \), defined by (11) and by noting that \( E[\varepsilon_t \varepsilon_t^T] = \sigma^2 \frac{I(p)}{I(p)} \) where \( I(p) \) is the Fisher information for the Huber probability function.

Figure 3. Shape of the different score and gain functions with \( \sigma = 1, \delta = 5\% \) and \( k = 1.399 \)

Tabulated values can be found in [3]. \( \frac{1}{I(p)} \) equals 1 for a gaussian law and increases as the contamination factor \( \delta \) increases.

\[
P_{xx,t} = P_{xx,t}^- - K_t \left[ H_t^T P_{xx,t}^- H_t + R_t \left( 2S_t^{-1} - \frac{1}{I(p)} I \right) \right] K_t^T
\]

In the application to follow \( \delta = 5\% \) that corresponds to \( k = 1.399 \) and \( \frac{1}{I(p)} = 1.256 \).

From relation (7), we make use of \( r_t \) which is available after the update. A possible solution is to iterate until convergence is reached but this procedure will destroy the time deterministic property of the filter. A better approach is to use \( r_t^{-} \) instead of \( r_t \). As \( x_t^{-} \) converges to \( x_t \) (the model performance improves during learning), \( r_t^{-} \) converges to \( r_t \). The weighting matrix is then defined by:

\[
S(i,i) = P_{yy,t}(i,i) \frac{\psi(r_t^{-}(i))}{r_t^{-}(i)}
\]

The gain matrix \( K_t \) as well as the covariance update matrix \( P_{xx,t} \) are still estimated using (11) and (13) to maintain asymptotic properties of the estimation. It has to be noted that using relation (14) can cause filter divergence if the initial model (initial parameter values and/or uncertainty) are too far from the actual ones. This aspect is discussed in more details in the application.

**EXTENSION TO NON-LINEAR SYSTEMS**

As modern turbofan models are non-linear, the state variables update and the measurements prediction equations become

\[
x_{t+1} = F(x_t, u_{t+1}, w_t) + \mu_t
\]

\[
y_t = H(x_t, w_t) + \varepsilon_t
\]

Several Kalman filters have been extended to non-linear systems. The so-called Extended Kalman filter uses a linearization of the system at the current state estimate, while the Unscented Kalman filters developed by Julier and Uhlmann is based on the following intuition: \textit{with a fixed number of parameters it should be easier to approximate a gaussian distribution than an arbitrary non-linear function/ transformation}. The unscented transformation is a method for calculating statistics which undergo a non-linear transformation. The main reason of using the unscented Kalman filter instead of the extended one resides in its improved covariance prediction accuracy. The unscented transformation is accurate to the fourth order while the linearization projection of the extended Kalman filter is only of the second order [10].
Consider a random variable $x$ of dimension $L$ through a non-linear function $y = f(x)$. Assume $x$ has mean $\bar{x}$ and covariance $P_x$, to calculate the statistics of $y$, we form a matrix $X$ of $2L + 1$ vectors $X_i$ perturbed according to the following:

\begin{align}
X_0 &= \bar{x}, \\
X_i &= \bar{x} + (\gamma \sqrt{P_x})_i, \text{ for } i = 1, \ldots, L \\
X_i &= \bar{x} - (\gamma \sqrt{P_x})_i, \text{ for } i = L + 1, \ldots, 2L
\end{align}

$\gamma = \sqrt{L + \lambda}$ and $(\sqrt{P_x})_i$ is the $i^{th}$ column of the matrix square root (e.g., lower triangular Cholesky factorisation). These vectors are propagated through the non-linear function:

\[ y_i = f(X_i), \text{ for } i = 0, \ldots, 2L \]

and the mean and covariance for $y$ are approximated using a weighted sample mean and covariance of the posterior sigma points:

\[ \bar{y} \simeq \sum_{i=0}^{2L} z_i^{(m)} y_i \quad \text{and} \quad P_y \simeq \sum_{i=0}^{2L} z_i^{(c)} [(y_i - \bar{y})(y_i - \bar{y})^T] \]

with $z_i$ given by

\begin{align}
z_0^{(m)} &= \frac{\lambda}{\sqrt{L + \lambda}}, \\
z_0^{(c)} &= \frac{\lambda}{\sqrt{L + \lambda}} + 1 - \alpha^2 + \beta, \\
z_i^{(c)} &= z_i^{(m)} = \frac{1}{2(\sqrt{L + \lambda})}, \forall i = 1, \ldots, 2L
\end{align}

where $\lambda = \alpha^2(L + \kappa) - L$ is a scaling parameter and $\beta$ is used to incorporate prior knowledge about the probability distribution. We refer the interested reader to [1] for detailed information about the value of these parameters. The unscented Kalman filter making use of this non-linear transformation is summarized in algorithm 2.

The recursion $P_{x,t} = P_{x,t} - K_z P_{y,t} K_z^T$ does not guarantee that the matrix $P_{x,t}$ remains positive-definite. To avoid such problems, the square root unscented Kalman filter (SR-UKF) is used. This filter has been modified to provide robustness. One interesting aspect is that making the SR-UKF robust does not affect much the computational effort.

**Algorithm 2** Unscented Kalman filter: state update

**Require:** $x_0$ and $P_0$ {Initialization}

**while** new measurements are performed **do**

{Calculate sigma points}

\[ X_{t-1} = [\bar{x}_{t-1} \quad \gamma \sqrt{P_{x,t-1}} \quad \gamma \sqrt{P_{x,t-1}}] \]

{State estimate and covariance propagation}

\[ X_t^- = F(X_{t-1}, u_t) \Rightarrow \bar{x}_t = \sum_{i=0}^{2L} z_i^{(m)} X_i^- \]

{Calculate sigma points and measurement update}

\[ X_t = [\bar{x}_t \quad \gamma \sqrt{P_{x,t}} \quad \bar{x}_t - \gamma \sqrt{P_{x,t}}] \]

{Kalman gain computation}

\[ Y_t = H(X_t) \Rightarrow \bar{y}_t = \sum_{i=0}^{2L} z_i^{(m)} Y_i \]

{State and error covariance prediction}

\[ \bar{x}_t = \bar{x}_{t-1} + K_t r \]

\[ P_{x,t} = P_{x,t} - K_t P_{y,t} K_t^T \]

**end while**

**DUAL ESTIMATION**

**Parameter estimation** or system identification involves determining the non-linear mapping

\[ d_t = G(x_t, w) \]

where $x_t$ are the input, $d_t$ the output and the non-linear function $G$ is parameterized by model parameters $w$. Model parameters $w$ are determined by minimizing any particular error function. While gradient based algorithms exist, the Kalman filter may be efficiently used to estimate the parameters $w$ by writing a new state space representation

\[ w_{t+1} = w_t + \eta_t \]

\[ d_t = G(x_t, w) + e_t \]

This parameter estimation is performed for both the time update and the measurement equations. To learn the state dynamics, simply make the substitutions $G \rightarrow F, d_t \rightarrow x_{t+1}$ and $e_t \rightarrow \mu_t$. To learn the measurement update, make the substitutions $G \rightarrow H, d_t \rightarrow y_t$ and $e_t \rightarrow \epsilon_t$. For both cases the state $x_t$ must be available and free of noise [11, 12].
Algorithm 3 Unscented Kalman filter: parameter update

Require: \( \tilde{w}_0 \) and \( P_{ww;0} \). {Initialization}
while new measurements are performed do

{Parameter estimate and covariance propagation}
\[
\tilde{w}_t = \tilde{w}_{t-1},
\]
\[
P_{ww;t} = P_{ww;t-1} + N_t
\]
{Calculate sigma points and measurement update}
\[
\mathcal{W}_t = \begin{bmatrix} \tilde{w}_t \tilde{w}_t^T + \gamma \sqrt{P_{ww;t}} & \tilde{w}_t^T - \gamma \sqrt{P_{ww;t}} \end{bmatrix}
\]
\[
\mathcal{Y}_t = H[\mathcal{W}_t] \Rightarrow \tilde{y}_t = \sum_{i=0}^{2L} \xi_i^{(m)} \mathcal{Y}_t
\]
{Kalman gain computation}
\[
P_{yy} = \sum_{i=0}^{2L} \xi_i^{(c)} [\mathcal{Y}_t - \tilde{y}_t] [\mathcal{Y}_t - \tilde{y}_t]^T + R_t
\]
\[
r = (y_t - \tilde{y}_t)
\]
\[
P_{yy;t} = \sum_{i=0}^{2L} \xi_i^{(c)} [\mathcal{Y}_t - \tilde{y}_t] [\mathcal{Y}_t - \tilde{y}_t]^T + R_t S_t^{-1}
\]
\[
P_{wy;t} = \sum_{i=0}^{2L} \xi_i^{(c)} [\mathcal{Y}_t - \tilde{y}_t] [\mathcal{Y}_t - \tilde{y}_t]^T
\]
\[
K_t = L_t^{-1} P_{wy;t}
\]
{State and error covariance prediction}
\[
\tilde{w}_t = \tilde{w}_{t-1} + K_t r
\]
\[
P_{ww;t} = P_{ww;t-1} - K_t P_{yy;t} K_t^T
\]
end while

If \( \eta_t \) is a zero mean normal noise with covariance matrix \( N_t \), the algorithm 3 updates the parameter \( \tilde{w}_t \).

The covariance projection \( P_{ww;t} = P_{ww;t-1} + N_t \) can be replaced by
\[
P_{ww;t} = P_{ww;t-1} + (\lambda_{RLS}^{-1} - 1) P_{ww;t-1} = \lambda_{RLS}^{-1} P_{ww;t-1}, \quad (28)
\]
where \( \lambda_{RLS} \) is a "forgetting" factor controlling the time constant of an exponential window over the data. The discrete time step being \( dt \), the time constant \( \Delta t \) may be computed as
\[
\Delta t = -\frac{dt}{\ln \lambda_{RLS}}, \quad (29)
\]
The algorithm 3 is equivalent to a batch algorithm (all the data processed at the same time) if \( \lambda_{RLS} = 1 \) and when \( \lambda_{RLS} \) decreases, the model becomes more adaptive.

**Dual estimation** combines the state estimation and the parameter estimation. Now the task is to estimate the state \( \tilde{x}_t \) and the model parameters \( \tilde{w}_t \) from noisy information. Essentially, two filters are run concurrently. At every time step a state estimation filter estimates the state assuming \( \tilde{w}_t \) while the parameter estimation filter estimates the weight using the current state estimate \( \tilde{x}_t \). Figure 4 describes the configuration that has been used in the application to follow.

### TEST CASES

The general structure of the original model is represented in figure 1. The Kalman gain corrects the state variables which are integrated to the next step. The parameter estimation is the same as in figure 4. Algorithm 2 is used for the state estimation and algorithm 3 is used for the health parameter estimation. The acquisition frequency used in the test cases is 10Hz which corresponds to frequencies used in commercial test benches. Thrust and the mass flow rate measurements have been added to the usual OBDICOTE variables since they are available in a test bench.

**Engine state tracking and measurement validation**

The first application focuses on engine state tracking that achieves robust measurement validation. The health parameter estimation is not considered (no engine fault has been simulated).

Figures 5 and 6 summarize the results of a classical gaussian unscented Kalman filter facing a constant sensor fault on \( N_{LP} \) (+50 rpm). It is seen by the divergence of \( T^b_4 \) that gaussian noise cannot cope with sensor faults. \( T^b_4 \) diverges because it is the least observable state variable. This is due to the fact that no feedback measurement is available in the combustion chamber and that \( T^b_4 \) is not strongly correlated with the other available measurements.

Conversely the robust filter does not show any divergence even in the case of a multiple sensor fault (Figure 7 and 8), the filter converges to the true values in a few seconds and is not perturbed by the sensor faults. The robust filter has a high breakdown point and therefore is still efficient in numerous sensor faults situation. Figure 8 compares the detected sensor fault with the effective one, and shows that the filter efficiently isolate any sensor fault even with drifting faults (on \( T_{13} \) and Thrust).
Engine diagnostic at test bench

This test is representative of an engine going through a test bench for maintenance. A high pressure turbine degradation has been simulated by a 2% drop of its efficiency (SE42=0.98). The fault is present from the beginning of the test and does not vary much during the test. Therefore, the model must not be adaptive and all data are equally weighted. This is done by setting $\lambda_{RLS}$ to one in the parameter estimation filter. Initial health parameter values are set to 0.99 with an uncertainty of 1%, representing the prior knowledge about the parameters.

After ten minutes the fault is clearly located and the engine state is accurately tracked (figures 9 and 10).

Adaptive model and fault tracking

The following application deals with a component fault evolving in time, which is maybe more representative of performance monitoring. The model has to adapt itself to this varying fault. This is done by setting the exponential window to a constant that is representative of the time behavior of the fault. In this application, the fault is supposed to vary in half an hour (due to CPU time limitations) and the constant $\lambda_{RLS}$ is then set to 0.9999 according to (29). As this application is closer to an engine in operation the mass flow rate and the thrust measurements are disabled. A sensor fault on $T_{13}$ is also added to check the robustness of the algorithm. Results are summarized in figures 11 and 12 show that the model adapts itself to the fault and maintain an efficient state tracking.

The procedure is robust as it can cope with a constant faulty measurement on $T_{13}$ that concerns directly the component degradation. However as more adaptivity means that less measurements are used and therefore measurement redundancy is lowered, it must be emphasized that the more the model is adaptive the less it is able to cope with sensor faults, resulting in a trade off between adaptivity and robustness.
ADDITIONAL COMMENTS

The three applications mentioned in this contribution need a few more comments. First the set of health parameter doesn’t include the mass flow rate coefficients. Adding those coefficients would decrease the measurement redundancy and therefore the stability of the identification. This is mainly due to the fact that the health parameters are assumed independent by setting the initial covariance matrix $P_{\text{w},0}$ to a diagonal matrix.

On the other hand, adding off-diagonal terms into this matrix would add some linkage between the variation of those parameters. For example, as the efficiency decreases the mass flow rate should also decrease. This should avoid spurious variation of the health parameters. Such information may come from a priori experience or from a learning phase consisting in confronting known degradation situations to the model. This is a key aspect that has to be deeply studied to guarantee a stable and reliable identification. It has to be noted that this remark is also applicable to steady state identification problems.

Another important point is that state identification accuracy is lower for steady states than for transients. In steady states indeed the measured data do not carry much information about the engine dynamics. Accordingly some instabilities of the filter are observed that can cause filter divergence. A procedure that would update the engine state only if the data carry enough information such as the state variance estimation is still needed.
CONCLUSIONS

The robust form of the unscented Kalman filter seems to be efficient for the dual estimation of both the state and health parameters of a gas turbine engine during transient sequences. The robustness property of this filter is very useful when sensor faults have to be dealt with: identified parameters are made less sensitive to sensor faults and therefore sensor fault isolation is made easier.

While some aspects concerning parameter identifiability, state observability during steady states and health parameters cross correlation (off diagonal terms of the parameter covariance matrix) have to be more deeply understood, the results of the presented method are very encouraging.

The recurrent, time deterministic structure of the Robust Kalman filter makes it very suitable for on-line engine performance monitoring or adaptive controller. Indeed if the data processing is performed after the test, a huge database must be recorded to describe the transients and no provision is made for possible faulty sensors. The present methodology is aimed to perform an on-line monitoring which minimizes storage and requires limited computational efforts. The bottleneck of the method is that physical models are much too slow to achieve real time health monitoring. A solution would be to substitute the physical model by a neural network or any other automatic learning technique. This approach has already been successfully applied by the authors to a steady state turbojet application [13,14].

ACKNOWLEDGEMENTS

The authors wish to thank Snecma for providing the OBIDI-COTE model. This work is funded by the Walloon Region of Belgium in the frame of its Program FIRST and is supported by Techspace Aero, Snecma Group.

REFERENCES