An Extension to the Kalman filter for an Improved Detection of Unknown Behavior

Emmanuel Benazera and Sriram Narasimhan

Abstract— **The use of Kalman filter (KF) interferes with fault detection algorithms based on the residual between estimated and measured variables, since the measured values are used to update the estimates. This feedback results in the estimates being pulled closer to the measured values, influencing the residuals in the process. Here we present a fault detection scheme for systems that are being tracked by a KF. Our approach combines an open-loop prediction over an adaptive window and an information-based measure of the deviation of the Kalman estimate from the prediction to improve fault detection.**

I. Introduction

A. Kalman Filter

Consider a discrete-time controlled process that is governed by a linear stochastic difference equation (1) and a measurement (2):

$$x(t_i) = Ax(t_{i-1}) + Bu(t_i) + w(t_i) \tag{1}$$

$$z(t_i) = Hx(t_i) + v(t_i) \tag{2}$$

$w(t_i)$, $v(t_i)$ represent the process and measurement noise respectively and are assumed to be independent, white and Gaussian with probability distributions $\mathcal{N}(0,Q)$, $\mathcal{N}(0,R)$ respectively. Given the noise in the process and measurements, the KF [1] computes an unbiased estimate $\hat{x}$ of the state x by providing an optimal solution of the least-squares method. This is achieved by recursively minimizing the *a posteriori* estimate error covariance $P(t_i) = E[e(t_i)e^T(t_i)]$ where $e(t_i) = x(t_i) - \hat{x}(t_i)$ is the *a posteriori* error between the true state $x(t_i)$ and the *a posteriori* state estimate $\hat{x}(t_i)$. First the state and error variance estimates are projected forward from time t_{i-1} to time t_i through the following equations:

$$\hat{x}(t_i^-) = A\hat{x}(t_{i-1}) + Bu(t_i) \tag{3}$$

$$P(t_i^-) = AP(t_{i-1})A^T + Q \tag{4}$$

where t_i^- indicates *a priori* values. An adaptive gain factor K minimizes (in the least-square sense) the error covariance. Noisy measurements of the process are then used to compute the *a posteriori* state estimate. Finally the *a posteriori* covariance estimate is computed. These three steps are summarized as:

$$K(t_i) = P(t_i^-)H^T(HP(t_i^-)H^T + R)^{-1} \tag{5}$$

$$\hat{x}(t_i) = \hat{x}(t_i^-) + K(t_i)(z(t_i) - H\hat{x}(t_i^-)) \tag{6}$$

$$P(t_i) = (I - K(t_i)H)P(t_i^-) \tag{7}$$

The KF has been the subject of extensive research and applications ([2]).

B. Fault Detection and the Kalman Filter

We argue that in several situations the KF is in cross-purposes with the fault detection. First, the KF is designed to filter any deviations in the measurements and predictions by using the measurement updates. As a result the magnitude of the residual $\epsilon(t_i) = z(t_i) - H\hat{x}(t_i^-)$ is reduced, affecting the fault detection capability. Second, when the measurement noise is high the error covariance is so large that even a large residual falls well within its bounds. Furthermore, since the gain factor K is not dependent on the input matrix B, the covariance minimization is not affected by any faults on the input ([3]).

II. Preliminaries

A. n-step predictor

We define the n-step predictor of the state $\tilde{x}_n(t_i)$ to be the n-step open-loop estimate of the state. $\tilde{x}_n(t_i)$ is computed recursively by taking the KF state estimate at time t_{i-n} and then projecting it forward for n steps using equation (3). The covariance is also projected forward using equation (4).

$$\tilde{x}_n(t_i) = A^n\hat{x}(t_{i-n}) + \sum_{j=1}^{n} A^{n-j}Bu(t_{i-n+j}) \tag{8}$$

$$\tilde{P}_n(t_i) = A^nP(t_{i-n})(A^T)^n + \sum_{j=0}^{n-1} A^jQ(A^T)^j \tag{9}$$

Let $\tilde{X}_n(t_i) \sim \mathcal{N}(\tilde{x}_n(t_i), \tilde{P}_n(t_i))$ be the random variable corresponding to the n-step prediction. We define $D_n(t_i) = \tilde{X}_n(t_i) - \hat{X}(t_i)$, and $D_n(t_i) \sim \mathcal{N}(E[D_n(t_i)], \mathrm{cov}(D_n(t_i)))$:

$$\begin{aligned} E[D_n(t_i)] &= d_n(t_i) = \tilde{x}_n(t_i) - \hat{x}(t_i) \\ &= Ad_{n-1}(t_{i-1}) + K(t_i)(H(Bu(t_i) \\ &\quad +A\hat{x}(t_{i-1})) - z(t_i)) \end{aligned} \tag{10}$$

If $n = 1$, then $\tilde{x}_1(t_{i-1}) = \hat{x}(t_{i-1})$ and $d_1(t_i) = -K(t_i)\epsilon(t_i)$ as $d_0(t_i) = 0$.

E. Benazera is with RIACS, S. Narasimhan is with QSS Group Inc. / NASA Ames Research Center,Moffett Field, CA, USA. ebenazer@email.arc.nasa.gov, sriram@email.arc.nasa.gov

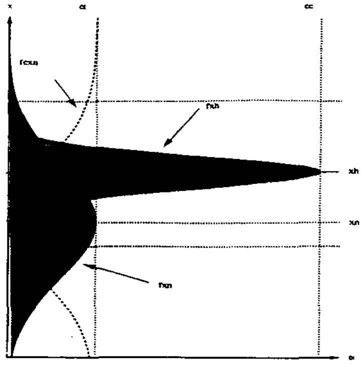

Fig. 1. A posteriori n-steps prediction likelihood (based on KL divergence).

III. Fault Detection

When the system is behaving nominally (without faults) we expect the measurements, estimates and the n-step prediction to be close to each other. We examine the quality of the produced state estimate given the measures and the open-loop estimate (with limited measure influence). The probability of the estimated state $\hat{X}(t_i)$ at any time t_i given the measurements $z(t_i)$ and the n-step prediction $\tilde{X}_n(t_i)$ is given by:

$$
\begin{aligned}
p(\hat{X}(t_i) \mid z(t_i), \tilde{X}_n(t_i)) &= \frac{p(z(t_i), \tilde{X}_n(t_i) \mid \hat{X}(t_i)) p(\hat{X}(t_i))}{p(z(t_i), \tilde{X}_n(t_i))} \\
&\approx p(z(t_i) \mid \tilde{X}_n(t_i), \hat{X}(t_i)) p(\tilde{X}_n(t_i) \mid \hat{X}(t_i)) p(\hat{X}(t_i))
\end{aligned} \tag{11}
$$

For each value of x, the higher the probability returned by this distribution, the higher we expect the system to be nominal.

A. Likelihood Indicators

To make decisions, we use a likelihood (L) indicator:

$$
\begin{aligned}
&L(N \mid z(t_i), \tilde{X}_n(t_i)) \\
&\quad \triangleq L(z(t_i) \mid \tilde{X}_n(t_i)) L(\tilde{X}_n(t_i) \mid \hat{X}(t_i)) p_N
\end{aligned} \tag{12}
$$

where $L(z(t_i) \mid \tilde{X}_n(t_i))$ is the *a priori* n-steps measurement likelihood, $L(\tilde{X}_n(t_i) \mid \hat{X}(t_i))$ is the *a posteriori* n-steps prediction likelihood, and p_N is the fixed probability that no fault occurs at each time step. The *a priori* n-steps measurement likelihood $L(z(t_i) \mid \tilde{X}_n(t_i))$ is based on the distance between $z(t_i)$ and $\tilde{z}_n(t_i) = H\tilde{x}_n(t_i)$ given the covariance $\tilde{S}_n(t_i) = H\tilde{P}_n(t_i)H^T$. This distance is expected to be small under nominal behavior, and to increase when a fault occurs. We have:

$$
\begin{aligned}
L(z(t_i) \mid & \tilde{X}_n(t_i)) \sim \mathcal{N}(\tilde{z}_n(t_i), \tilde{S}_n(t_i)) \\
= & f_{\tilde{X}_n(t_i)}(z(t_i))
\end{aligned}
$$

where $f_{\hat{X}(t_i)}(x) = C(P(t_i))\exp(-\frac{1}{2}(x - \hat{x}(t_i))^T P^{-1}(t_i)(x - \hat{x}(t_i)))$ and $C(P(t_i)) = 1/(2\pi^{n_x/2}|P(t_i)|^{1/2})$. Due to the potentially large variance $\tilde{S}_n(t_i)$, $L(z(t_i) \mid \tilde{X}_n(t_i))$ may not be sufficient for quick detection.

The *a posteriori* n-steps prediction likelihood $L(\tilde{X}_n(t_i) \mid \hat{X}(t_i))$ assesses the distance between $\tilde{X}_n(t_i)$ and $\hat{X}(t_i)$. We examine the Kullback-Leibler (KL) divergence [4] between $\tilde{X}_n(t_i)$ and $\hat{X}(t_i)$ which measures how different the two distributions are[1]. $KL(\tilde{X}_n(t_i), \hat{X}(t_i))$ can be understood as the average number of bits that are wasted by encoding events from the predicted distribution (over n-steps) with a code based on the estimated distribution. Therefore, the less bits are wasted, the more it is likely the system behavior is nominal. We thus note $L(\tilde{X}_n(t_i), \hat{X}(t_i)) = KL(C(\tilde{P}_n(t_i)) - \tilde{X}_n(t_i), \hat{X}(t_i))$. This number is typically infinite as the surface under $C(\tilde{P}_n(t_i)) - f_{\tilde{X}_n(t_i)}$ is infinite (see figure 1). We thus study the number of wasted bits over 99.7% of $\hat{X}$'s variance instead to get a good approximation (through Monte-Carlo (MC) simulation). Based on this, the fault indicator (F) mirrors the nominal one:

$$
\begin{aligned}
&L(F \mid z(t_i), \tilde{X}_n(t_i)) \\
&\triangleq (C(\tilde{P}_n(t_i)) - L(z(t_i) \mid \tilde{X}_n(t_i)))(KL(\tilde{X}_n(t_i), \hat{X}(t_i)))p_F
\end{aligned} \tag{13}
$$

where $p_F = 1 - p_N$.

B. Fault decision

Considering the two classes N and F and their respective conditional likelihoods $L(N \mid z(t_i), \tilde{X}_n(t_i))$, $L(F \mid z(t_i), \tilde{X}_n(t_i))$, two decision functions are built, that discriminate between the two classes given $z(t_i)$ and $\tilde{X}_n(t_i)$:

$$
\begin{aligned}
g_N(t_i) \triangleq & \log(L(z(t_i) \mid \tilde{X}_n(t_i))) + L(\tilde{X}_n(t_i) \mid \hat{X}(t_i)) \\
& + \log(p_N)
\end{aligned}
$$

and

$$
\begin{aligned}
g_F(t_i) \triangleq & \log(C(\tilde{P}_n(t_i)) - L(z(t_i) \mid \tilde{X}_n(t_i))) \\
& + KL(\tilde{X}_n(t_i), \hat{X}(t_i)) + \log(p_F)
\end{aligned} \tag{14}
$$

The overall decision function is then based on the sign of

$$
\delta_n(z(t_i), \tilde{X}_n(t_i), \hat{X}(t_i)) = g_N(t_i) - g_F(t_i) \tag{15}
$$

and is given by: a fault occured if $\delta_n(z(t_i), \tilde{X}_n(t_i), \hat{X}(t_i)) < 0$.

C. Determining n dynamically

One key factor in the effectiveness of our fault detector is the value for n. Here, we propose to dynamically adapt n. We study the changes in the decision line (15) as a result of unit change in n: this comes to comparing the decision lines for an n and n+1-step predictors. We note $\delta_{n+1,n} = \delta_{n+1} - \delta_n$. This short paper precludes the writing of the complete developments, so we give the reader a brief outline of our methods: we study the deritative values of $\delta_{n+1,n}$ w.r.t. $z(t_i)$ and $\tilde{x}_n(t_i)$, then the orientation of these two vectors of derivatives with respect to each other in the observation space: if they are negatively oriented, n stays unchanged,

[1] Note that it is not real distance, as it is not symmetric.

otherwise the sign of $\delta_{n+1,n}$ decides for n increment. For this we need to project the derivative with respect to $\tilde{x}_n(t_i)$ to the observation space. θ being the angle between the vectors in the observation space, we have:

$$\cos\theta = \frac{(\frac{\partial}{\partial z(t_i)}\delta_{n+1,n}(t_i))^T(H\frac{\partial}{\partial \tilde{x}_n(t_i)}\delta_{n+1,n}(t_i))}{\|\frac{\partial}{\partial z(t_i)}\delta_{n+1,n}(t_i)\|.\|H\frac{\partial}{\partial \tilde{x}_n(t_i)}\delta_{n+1,n}(t_i)\|}$$

where $\|.\|$ denotes the l_2 norm. The adaptation strategy for n is then given by:

$$\begin{cases} \pi \leq \theta \leq 2\pi & n(t_{i+1}) = n(t_i) \\ \text{else if } \delta_{n+1,n} > 0 & n(t_{i+1}) = n(t_i) - 1 \\ \text{else } \delta_{n+1,n} \leq 0 & n(t_{i+1}) = n(t_i) + 1 \end{cases} \tag{16}$$

D. Implementation

Algorithm 1 presents the filter loop at time step t_i. It is initialized with $\hat{x}(0) = \tilde{x}_n(0) = x_0$, $P(0) = \tilde{P}_n(0) = P_0$ and $n = n_{min}$. The implementation requires storing or recomputing several values and matrices: $\epsilon(t_{i-n})$, $K(t_{i-n})$, $P(t^-_{i-n})$. This is consistent with modern diagnosis engines that work on a fixed temporal window [5], although increasing the computational complexity of the KF. The following results help in mitigating the computational effort:

$$d_{n+1}(t_i) - d_n(t_i) = -A^n K(t_{i-n})\epsilon(t_{i-n}) \tag{17}$$

$$\tilde{P}_{n+1}(t_i) - \tilde{P}_n(t_i) = A^n K(t_{i-n}) H P(t^-_{i-n})(A^n)^T \tag{18}$$

These relations appear on steps 2 and 3:

```
1: Standard Kalman filter prediction and update.
2: $d_n(t_i)$ computation:
       $d_{n-1}(t_{i-1}) = d_n(t_{i-1}) + A^{n-1}K(t_{i-n+1})\epsilon(t_{i-n+1})$
       $d_n(t_i) = Ad_{n-1}(t_{i-1}) + K(t_i)(HBu(t_i) + HA\hat{x}(t_{i-1}) - z(t_i))$
3: $n$-steps prediction:
       $\tilde{x}_n(t_i) = d_n(t_i) + \hat{x}(t_i)$
       $\tilde{P}_{n-1}(t_{i-1}) = \tilde{P}_n(t_{i-1}) - A^{n-1}K(t_{i-n+1}) HP(t^-_{i-n+1})(A^{n-1})^T$
       $\tilde{P}_n(t_i) = A\tilde{P}_{n-1}(t_{i-1})A^T + Q$
4: Fault detection: $sign(\delta_n(z(t_i), \tilde{X}_n(t_i), \hat{X}(t_i)))$.
5: Adapt $n$ according to $\theta$ and $\delta_{n+1,n}(z(t_i), \tilde{X}_n(t_i), \tilde{X}_{n+1}(t_i), \hat{X}(t_i))$.
```

Algorithm 1: KF with n-steps open-loop and fault detection

IV. Application

The application that motivated this research was that of a hybrid diagnosis engine that combines a Rao-blackwellized particle filter (RBPF) [6] with a logical approach to the diagnosis [7]. An efficient fault detector was needed to articulate the tracking and the consistency-based engine for logical diagnosis, i.e. for deciding when to trigger the latter, or returning to the former.

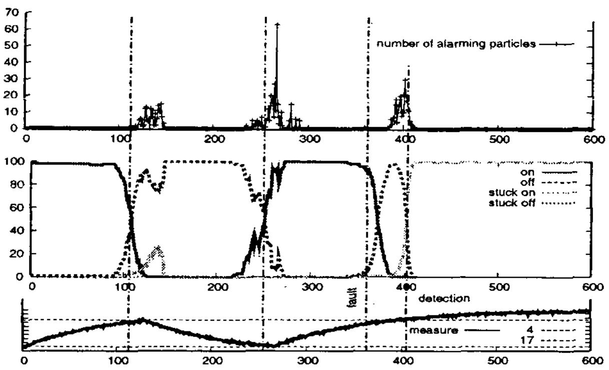

Fig. 2. Bottom graph: a simulated thermostat fails turning off around step 370. Middle graph: identified modes (percentages). The RBPF with embedded fault detector alarms on early mode changes and lowers particles weight that identify the wrong mode. Top graph: the percentage of alarming particles (over 100 particles). The bumps correspond to the system nominal and faulty mode changes.

As a preliminary test, we plugged the fault detector into the RBPF and tracked a simulated noisy thermostat. The RBPF tracks multi-modal linear systems with Gaussian noise. The belief state is a mixture of Gaussians whose statistics are propagated with a KF. The particle weight is computed as the observation probability $p(z(t_i) \mid \hat{X}(t^-_i))$. Our strategy uses the fault detector to assert the quality the estimate and lowers the weight of particles that are not in the correct mode. Figure 2 shows a run on a faulty thermostat ($n \leq 50$): the number of alarming particles rises at each mode change. Our version of the filter detects wrong modes and faults almost instantly. Identification however depends on the modes sampling.

Unfortunately, on large multi-dimensional continuous spaces, the computational weight of the detector is very heavy due to the MC calls for the KL computation. Moreover, results are deceiving on systems with uncertain parameters (high process noise) and precise sensing (low observation noise). For these reasons, we are not using this detector in our current diagnosis engines.

References

[1] E. Kalman, Rudolph, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[2] "http://www.cs.unc.edu/ welch/kalman/."

[3] P. D. Hanlon and P. S. Maybeck, "Characterization of kalman filter residuals in the presence of mismodeling," *IEEE Transactions on Aerospace and Electronic systems*, vol. 36, no. 1, pp. 114–131, january 2000.

[4] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[5] P. Nayak and J. Kurien, "Back to the future for consistency-based trajectory tracking," in *Proceedings of AAAI-2000, Austin, Texas*, 2000.

[6] N. de Freitas, "Rao-blackwellised particle filtering for fault diagnosis," in *IEEE Aerospace*, 2002.

[7] "Combining particle filters and consistency-based approaches for monitoring and diagnosis of stochastic hybrid systems," in *Proc. 15th International Workshop On Principles of Diagnosis, DX04*, 2004.