Pergamon

Automatica, Vol. 30, No. 8, pp. 1333–1338, 1994

0005-1098/94 $7.00 + 0.00

Brief Paper

Adaptive Fading Kalman Filter with an Application*

QIJUN XIA,† MING RAO,† YIQUN YING† and XUEMIN SHEN†

Key Words—Kalman filter; state estimation; adaptive estimation; discrete system; industrial processes.

Abstract—A new adaptive state estimation algorithm, namely adaptive fading Kalman filter (AFKF), is proposed to solve the divergence problem of Kalman filter. A criterion function is constructed to measure the optimality of Kalman filter. The forgetting factor in AFKF is adaptively adjusted by minimizing the defined criterion function using measured outputs. The algorithm remains convergent and tends to be optimal in the presence of model errors. It has been successfully applied to the headbox of a paper-making machine for state estimation.

1. *Introduction*

CONTROL SYSTEM design very often involves the estimation of unmeasurable states. Kalman and Bucy (1961) introduced an effective algorithm to realize the optimum filter for Gaussian processes. The recursive computation nature of the algorithm has attracted much attention. This well-known Kalman filtering technique has been widely employed in inertial navigation (Tze-Kwarfung and Grimble, 1983), target tracking (Chang and Tabaczynski, 1984) and industrial processes (Bialkowski, 1983).

In spite of its successful use, Kalman filter still has some drawbacks. Inaccuracy in system models may seriously degrade the performance of the filter. Particularly, the usefulness of the filter may be nullified by the "divergence" phenomenon (Fitzgerald, 1971). The linear model of a real system is usually obtained as a result of either purposeful approximation and simplification or lack of knowledge about the true characteristics of the system, which is always erroneous. The convergence problem is hence becoming a main research subject of Kalman filter.

Shellenbarger (1966) considered the problem of unknown process noise covariances and computed maximum likelihood estimates of the unknown variable from the residuals. Ohap and Stubberud (1976) provided an adaptive algorithm to determine the optimal gain matrix for discrete time systems with stationary ergodic white noise. Masreliez and Martin (1977) and Tsai and Kurz (1983) compensated model errors by noises, and thus suggested the noise distributions be non-Gaussian. They proposed a robust Kalman filter based on the m-interval polynomial approximation (MIPA). Another approach to the divergence problem is to limit effective filter memory length. Fagin (1964) and Sorenson and Sacks (1971) pointed out that a given linear model is adequate for a certain duration of time, but may be inadequate over long time intervals. Thus, they suggested limiting the memory of the Kalman filter by using exponential fading of past data via forgetting factors. On the other hand, it is found that it is advantageous to vary the time constant of the exponentially weighted filter when there are unpredictable jumps and drifts. Ydstie and Co (1985) proposed a variable forgetting factor (VFF) algorithm in which forgetting factors were determined based on "memory length". Rapid fading occurs when data give poor fit with the model, and slow fading for a good fit.

The above techniques have successfully improved the convergence of Kalman filter. However, there are still further needs to improve the optimality of filter. This paper deals with the optimality and convergence of Kalman filters in the presence of both model parameter and noise covariance errors. New algorithms are developed to adaptively adjust the forgetting factors according to the optimality condition of the Kalman filter. Thus the filter remains convergent and tends to be optimal in the cases where there exist model errors. The algorithms are efficient and have very moderate computation burden, and are thus convenient to be implemented for industrial applications.

2. *Problem formulation*

Consider a linear, discrete time, stochastic multivariable system

$$x(k+1) = \Phi(k+1, k)x(k) + G(k)w(k) \quad (1)$$

$$y(k) = H(k)x(k) + v(k), \quad (2)$$

where $x(k)$ is the $n \times 1$ state vector, $y(k)$ is the $m \times 1$ measurement vector, $\Phi(k+1, k)$ and $H(k)$ are state transition matrix and observation matrix, respectively. $w(k)$ and $v(k)$ denote sequences of uncorrelated Gaussian random vectors with zero means, the covariance matrices of which are

$$E[w(k)w^T(j)] = Q(k)\delta_{kj} \quad (3)$$

$$E[v(k)v^T(j)] = R(k)\delta_{kj}. \quad (4)$$

The initial state $x(0)$ is specified as a random Gaussian vector

$$E[x(0)] = \bar{x}(0), \quad E[(x(0)-\bar{x}(0))(x(0)-\bar{x}(0))^T] = P(0). \quad (5)$$

If the system is completely observable, the equations describing the optimal estimator (the normal Kalman filter) are (Maybeck, 1982)

$$\hat{x}(k \mid k-1) = \Phi(k, k-1)\hat{x}(k-1) \quad (6)$$

$$\hat{x}(k) = \hat{x}(k \mid k-1) + K(k)[y(k) - H(k)\hat{x}(k \mid k-1)], \quad (7)$$

where

$$K(k) = P(k \mid k-1)H^T(k) \times [H(k)P(k \mid k-1)H^T(k) + R(k)]^{-1} \quad (8)$$

$$P(k+1 \mid k) = \Phi(k+1, k)P(k)\Phi^T(k+1, k) + G(k)Q(k)G^T(k) \quad (9)$$

$$P(k) = [I - K(k)H(k)]P(k \mid k-1). \quad (10)$$

* Received 23 March 1992; revised 31 August 1992; revised 17 March 1993; revised 24 August 1993; received in final form 14 September 1993. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor R. Boel under the direction of Editor Tamer Başar. Corresponding author M. Rao. Tel. +1 403 492 7433; Fax +1 403 492 2881.

† Department of Chemical Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2G6.

The normal Kalman filter provides the best (minimum variance, unbiased) estimation $\hat{x}(k \mid k-1)$ of the state $x(k)$ with the given observations $y(k-1)$, $y(k-2), \ldots, y(0)$ when the linear model for the system dynamics and measurement relation are perfect. Unfortunately, when the model is developed on the basis of an erroneous model, the filter can "learn the wrong state too well" (Synder, 1973). Because the filter estimation depends highly upon the past data, and the system model degrades the measurement information from the distant past, the heavy reliance on the past data may cause state estimation to diverge. In order to overcome this problem, the filter should be capable of eliminating the effect of older data from a current state estimate if these data are no longer meaningful due to the erroneous model. Fagin (1964) initiated a method to limit the memory of the Kalman filter by using exponential fading of past data via forgetting factor $\lambda(k)$. The equations describing the fading Kalman filter are identical to those of the normal Kalman filter in equations (6)–(10) except the forgetting factor $\lambda(k)$ in the time propagation error covariance equation

$$P(k+1 \mid k) = \lambda(k+1)\Phi(k+1, k)P(k)\Phi^T(k+1, k) + G(k)Q(k)G^T(k) \quad (11)$$

with $\lambda(k) \geq 1$. As a result, the influence of the most recent measured data in state estimation is overweighted and thus divergence is avoided.

The performance of the exponential fading Kalman filter fully depends on the selection of the forgetting factor. Therefore, how to generate optimal forgetting factor $\lambda(k)$ is the key problem in AFKF. In the following section, we present three algorithms for choosing optimal forgetting factor $\lambda(k)$ to improve the convergence and optimality of Kalman filter.

3. *Main results*

In developing the algorithms, we employ an important property of the optimal filter, that is, the residual $z(k)$ defined in the following equation is a white noise sequence when optimal filtering gain is used

$$z(k) = y(k) - H(k)\hat{x}(k \mid k-1). \quad (12)$$

For an arbitrary gain $K(k)$, it can be shown that the covariance of the residual is

$$C_0(k) = E[z(k)z^T(k)] = H(k)P(k \mid k-1)H^T(k) + R(k) \quad (13)$$

and the auto-covariance of the residual is

$$\begin{aligned} C_j(k) &= E[z(k+j)z^T(k)] \\ &= H(k+j)\Phi(k+j, k+j-1) \\ &\quad \times [I - K(k+j-1)H(k+j-1)] \cdots \Phi(k+2, k+1) \\ &\quad \times [I - K(k+1)H(k+1)]\Phi(k+1, k) \\ &\quad \times [P(k \mid k-1)H^T(k) - K(k)C_0(k)] \\ &\qquad \forall j = 1, 2, 3, \ldots. \end{aligned} \quad (14)$$

Substituting equations (8) and (13) into equation (14), $C_j(k)$ is identical to zero. This confirms that the sequence of residuals is uncorrelated when the optimal gain is used.

In practical situations, the real covariance of the residual $C_0(k)$ will be different from a theoretical one given in equations (8)–(10) and (13) because of the errors in model parameters and noise covariances. Thus, $C_j(k)$ may not be identical to zero. From equation (14), we know that if a forgetting factor can be chosen so that the last term of $C_j(k)$, which is the only common term of $C_j(k)$ for all $j = 1, 2, \ldots$, be zero

$$P(k \mid k-1)H^T(k) - K(k)C_0(k) = 0, \quad (15)$$

then $K(k)$ is optimal. In other words, if the gain is optimal, equation (15) holds. This forms the basis for the adaptive filtering algorithms developed below.

It should be noted that $C_0(k)$ in equation (15) is computed from measured data, rather than from equations (8)–(10) and (13).

Defining

$$S(k) = P(k \mid k-1)H^T(k) - K(k)C_0(k) \quad (16)$$

the optimality of the Kalman filter can be judged by a scalar function defined by

$$f(\lambda; k) = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij}^2(k), \quad (17)$$

where $S_{ij}(k)$ is the (i, j)th element of $S(k)$. The smaller the $f(k)$ is, the closer the filter is to the optimum. The absolute minimum of $f(k)$ means the most closely optimal estimate. Hence the forgetting factor $\lambda(k)$ should be chosen to minimize $f(k)$.

It should be pointed out that $C_j(k)$ depends also on the other terms in equation (14). However, these terms only include the gains in the future, $K(k+j-1)$ for $j = 2, 3, \ldots$. $C_1(k)$ depends just on the term $S(k)$. It is reasonable to consider equation (15) as a performance criterion, that is, to adjust the current gain matrix to improve the performance of the filter.

Since the measurement matrix $H(k)$ is involved in optimality condition (15) and in the relation between $K(k)$ and $P(k \mid k-1)$ [equation (8)], we assume that $H(k)$ is perfect. This assumption is reasonable in most real world processes since measurement relations are usually much easier to obtain than the system dynamics.

Algorithm 1. (*Steepest descent AFKF algorithm.*) Given system equations (1)–(5), the optimal forgetting factor can be obtained through iterative computation of the equation

$$\lambda^{l+1}(k) = \lambda^l(k) - \varphi\frac{\partial f^l(\lambda; k)}{\partial \lambda^l(k)} \quad \forall l = 0, 1, 2, \ldots, \quad (18)$$

with initial conditions

$$\lambda^0(1) = 1, \quad \lambda^0(k) = \lambda(k-1), \quad (19)$$

where k is the time series and l is the iteration times in a time instant. $0 < \varphi < 1$ is the step length in the gradient method.

At the pth iteration, if the following condition holds

$$|\lambda^p(k) - \lambda^{p-1}(k)| < \varepsilon \quad (20)$$

stop iteration and take

$$\lambda(k) = \max\{1, \lambda^p(k)\}. \quad (21)$$

The gradient term in equation (18) is presented as

$$\frac{\partial f^l(\lambda; k)}{\partial \lambda^l(k)} = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij}^l(k)\left(\frac{\partial S^l(k)}{\partial \lambda^l(k)}\right)_{ij}, \quad (22)$$

where

$$S^l(k) = P^l(k \mid k-1)H^T(k) - K^l(k)C_0(k) \quad (23)$$

$$\begin{aligned} \frac{\partial S^l(k)}{\partial \lambda^l(k)} &= \Phi(k, k-1)P(k-1)\Phi^T(k, k-1)H^T(k) \\ &\quad \times \{I - [T^l(k)]^{-1}C_0(k)\} + K^l(k)H(k) \\ &\quad \times \Phi(k, k-1)P(k-1)\Phi^T(k, k-1)H^T(k) \\ &\quad \times \{I + [T^l(k)]^{-1}C_0(k)\} \end{aligned} \quad (24)$$

and

$$P^l(k+1 \mid k) = \lambda^l(k+1)\Phi(k+1, k)P(k)\Phi^T(k+1, k) + G(k)Q(k)G^T(k) \quad (25)$$

$$K^l(k) = P^l(k \mid k-1)H^T(k)[T^l(k)]^{-1} \quad (26)$$

$$T^l(k) = H(k)P^l(k \mid k-1)H^T(k) + R(k) \quad (27)$$

$$P^l(k) = [I - K^l(k)H(k)]P^l(k \mid k-1). \quad (28)$$

The value of $C_0(k)$ can be estimated using recursive equations

$$C_0(k) = G_1(k)/G_2(k) \quad (29)$$

$$G_1(k) = G_1(k-1)/\lambda(k-1) + z(k)z^T(k) \quad (30)$$

$$G_2(k) = G_2(k-1)/\lambda(k-1) + 1 \quad (31)$$

with initial conditions

$$G_1(0) = 0, \quad G_2(0) = 0.$$

Proof. The criterion $f(\lambda; k)$ given in equation (17) is a nonlinear function of $\lambda(k)$. The problem of searching optimal forgetting factor $\lambda(k)$ is equivalent to the problem of searching the absolute minimum of nonlinear function $f(k)$. Applying the steepest descent method, the forgetting factor can be calculated by

$$\lambda(k+1) = \lambda(k) - \varphi \frac{\partial f(\lambda; k)}{\partial \lambda(k)}. \tag{32}$$

In order to find the gradient term in equation (32), taking derivations in equations (11) and (13) generates the following equations

$$\frac{\partial P(k \mid k-1)}{\partial \lambda(k)} = \Phi(k, k-1)P(k-1)\Phi^T(k, k-1) \tag{33}$$

$$\frac{\partial C_0(k)}{\partial \lambda(k)} = H(k)\frac{\partial P(k \mid k-1)}{\partial \lambda(k)}H^T(k). \tag{34}$$

Substituting equation (8) into (16) gives

$$S(k) = P(k \mid k-1)H^T(k)[I - T^{-1}(k)C_0(k)], \tag{35}$$

where $T(k)$ is defined in equation (27). Using equations (33) and (34) and differentiating equation (35) with respect to $\lambda(k)$, equation (24) (without the superscript l) is obtained.

From equation (17), equation (22) (without the superscript l) is obvious.

When more than one correction on $\lambda(k)$ is needed in a time instant, it is necessary to add superscript l in related equations to indicate the number of iteration. Applying the above results and adding superscript l in equations (8), (10), (11), (34) and (35) forms equations (22)–(28).

To estimate $C_0(k)$ by using on-line measured data, an unbiased consistent estimate for $C_0(k)$ based on k successive residuals is presented as

$$\hat{C}_0(k) = \frac{1}{k-1}\sum_{i=1}^{k} z(i)z^T(i). \tag{36}$$

Then a fading estimation formula for $C_0(k)$ can be given to overweight the most recent measured data

$$C_0(k) = \frac{\sum_{i=1}^{k-1} \sigma_{i,k} z(i)z^T(i) + z(k)z^T(k)}{\sum_{i=1}^{k-1} \sigma_{i,k} + 1}, \tag{37}$$

where

$$\sigma_{i,k} = \prod_{j=i}^{k-1} \frac{1}{\lambda(j)}. \tag{38}$$

Since $\lambda(j) \geq 1$ and thus $\sigma_{i,k} \geq \sigma_{i-1,k}$, it is obvious that the most recent data are overweighted. For the convenience of real-time application, we define

$$G_1(k) = \sum_{i=1}^{k-1} \sigma_{i,k} z(i)z^T(i) + z(k)z^T(k) \tag{39}$$

$$G_2(k) = \sum_{i=1}^{k-1} \sigma_{i,k} + 1. \tag{40}$$

It is easy to show that $G_1(k)$ and $G_2(k)$ can be computed recursively from (30) and (31). This completes the proof.

To improve the convergency and efficiency of the iteration process, the Armijo algorithm (Polak, 1971) can be applied to choose step size.

Defining

$$C(\lambda^0; k) = \{\lambda \mid f(\lambda; k) \leq f(\lambda^0; k)\} \tag{41}$$

and

$$\theta(\varphi; \lambda; k) = [f(\lambda + \varphi h(\lambda); k) - f(\lambda; k)] - \langle \nabla f(\lambda; k), h(\lambda; k) \rangle, \tag{42}$$

where

$$\nabla f(\lambda; k) = -\frac{\partial f(\lambda; k)}{\partial \lambda(k)},$$

and $h(\lambda; k)$ is given below. The Armijo algorithm can be stated as follows:

Step 1. Select $\lambda^0(k)$ such that the set $C(\lambda^0; k)$ is bounded; select $\alpha \in (0, 1)$, $\beta \in (0, 1)$ and $\rho > 0$.

Step 2. Set $l = 0$.

Step 3. Compute $h(\lambda^l; k) = -D(\lambda^l; k)\nabla f(\lambda^l; k)$.

Step 4. If $|h(\lambda; k)| \leq \epsilon$ (ϵ is a small positive scalar), set $\lambda(k) = \max\{1, \lambda^l(k)\}$ and stop; otherwise, go to next step.

Step 5. Set $\mu = \rho$.

Step 6. Compute $\theta(\mu; \lambda^l; k)$.

Step 7. If $\theta(\mu; \lambda^l; k) \leq 0$, set $\varphi^l = \mu$ and go to next step; otherwise set $\mu = \beta\mu$ and go to Step 6.

Step 8. Set $\lambda^{l+1}(k) = \lambda^l(k) - \varphi^l h(\lambda^l; k)$; set $l = l + 1$ and go to Step 3.

In this paper, we select $D(\lambda^l; k) = 1$, $\lambda^0(1) = 1$, $\lambda^0(k) = \lambda(k-1)$, $\alpha = \frac{1}{2}$, $\beta \in (0.5, 0.8)$ and $\rho = 1$.

Algorithm 1 fails to give an explicit formula for the calculation of $\lambda(k)$. Since iterative computation is involved, it may be difficult to apply this algorithm to real-time processes. In the remainder of this section, two one-step algorithms are developed.

Algorithm 2. (*One-step AFKF algorithm.*) Given system state equations (1)–(5) with the following Assumptions 1 and 2:

Assumption 1. $Q(k)$, $R(k)$ and $P(0)$ are all positive definite.

Assumption 2. The measurement matrix $H(k)$ is full-ranked. The optimal forgetting factor can be computed by

$$\lambda(k) = \max\left\{1, \frac{1}{m}\operatorname{trace}[N(k)M^{-1}(k)]\right\}, \tag{43}$$

where

$$M(k) = H(k)\Phi(k, k-1)P(k-1)\Phi^T(k, k-1)H^T(k) \tag{44}$$

$$N(k) = C_0(k) - H(k)G(k-1)Q(k-1) \times G^T(k-1)H^T(k) - R(k). \tag{45}$$

Proof. Substituting equation (8) into the optimality condition (15) gives

$$P(k \mid k-1)H^T(k)\{I - [H(k)P(k \mid k-1)H^T(k) + R(k)]^{-1}C_0(k)\} = 0. \tag{46}$$

Since $P(k \mid k-1)$ is nonsingular and $H(k)$ is assumed to be full-ranked, it is obvious that equation (46) implies the following relation

$$[H(k)P(k \mid k-1)H^T(k) + R(k)]^{-1}C_0(k) = I \tag{47}$$

or

$$H(k)P(k \mid k-1)H^T(k) = C_0(k) - R(k). \tag{48}$$

Equation (48) means that, with Assumptions 1 and 2, the optimality condition (15) is equivalent to (13). Substituting equation (11) into (48) and reorganizing it generates

$$\lambda(k)H(k)\Phi(k, k-1)P(k-1)\Phi^T(k, k-1)H^T(k) = C_0(k) - H(k)G(k-1)Q(k-1)G^T(k-1)H^T(k) - R(k). \tag{49}$$

Using equations (44) and (45), equation (49) is simplified as

$$\lambda(k)M(k) = N(k) \tag{50}$$

or

$$\lambda(k)I = N(k)M^{-1}(k), \quad \lambda(k) \geq 1. \tag{51}$$

Taking trace in both sides of the equation, we get (43), and this completes the proof.

In equation (43), the inversion of matrix $M(k)$ is involved, which will complicate the computation. To avoid inversion manipulation, trace is directly taken in both sides of equation (50) and this gives Algorithm 3.

Algorithm 3. (*Simplified one-step AFKF algorithm.*) Using system state equations and conditions given in Algorithm 2, the optimal forgetting factor can be computed by the following equation

$$\lambda(k) = \max\{1, \operatorname{trace}[N(k)]/\operatorname{trace}[M(k)]\}, \tag{52}$$

where matrices $M(k)$ and $N(k)$ are defined by equations (44) and (45).

In choosing the preferable algorithm, the trade-off between performance and real-time computational effort should be considered. In fact, simulation and industrial application results show that the performance of Algorithm 3 is quite satisfactory.

From equations (43)–(45) and (52), the physical meaning of optimal forgetting factor is clear: for unknown drifts and process changes, the adaptive fading algorithm compensates the increasing estimation errors by choosing larger forgetting factor.

The uniformly asymptotical stability of AFKF is obvious by applying the results obtained in Deyst and Price (1968) and Sorenson and Sacks (1971).

Remarks.

(1) When the filtering model is exactly correct, $C_0(k)$ is given as equation (13). Substituting equation (13) into (44)–(45), we have $M(k) = N(k)$, which thus results in $\lambda(k) = 1$. This implies that in the case where exact filtering model is used, AFKF functions in the same way as the normal Kalman filter and provides the optimal (minimum variance, unbiased) estimation. When the model errors cause the covariance of the residual to deviate from the theoretical one, AFKF compensates the model errors by adjusting the forgetting factor according to the optimality condition. In this way the filter achieves better performance.

(2) The matrix $S(k)$ has $n \times m$ elements. Since only one factor $\lambda(k)$ can be used to null out terms in $S(k)$, the filter may not be exactly optimal. However, since $\lambda(k)$ is adjusted according to the optimality condition of the filter, the optimality and convergence of the Kalman filter are surely improved.

4. *Simulation studies*

The comparison of performance between normal Kalman filter and AFKF has been undertaken in the cases of model coefficient errors and unknown drifts.

For simplicity, one-dimensional random state model is considered. For the scalar system, Algorithms 2 and 3 give the same results.

Case 1. *Model coefficient errors.*

The system equations of scalar discrete-time random state $x(k)$ are represented as

$$x(k+1) = 0.4x(k) + w(k) \tag{53}$$

$$y(k) = x(k) + v(k), \tag{54}$$

where $w(k) \sim N(0, 0.1^2)$, $v(k) \sim N(0, 0.5^2)$ and $x(0) \sim N(2, 0.2^2)$, and the erroneous filtering model is

$$x(k+1) = 0.8x(k) + w(k) \tag{55}$$

$$y(k) = x(k) + v(k). \tag{56}$$

The simulation results are presented in Fig. 1.

Case 2. *Unknown drifts.*

Consider a random state described by equations

$$x(k+1) = 0.5x(k) + 0.4 + w(k) \tag{57}$$

$$y(k) = x(k) + v(k), \tag{58}$$

where $w(k) \sim N(0, 0.05^2)$, $v(k) \sim N(0, 0.05^2)$ and $x(0) \sim N(2, 0.05^2)$, and the erroneous filtering model is

$$x(k+1) = 0.5x(k) + w(k) \tag{59}$$

$$y(k) = x(k) + v(k). \tag{60}$$

The simulation results are presented in Fig. 2.

From Figs 1 and 2, it is found that the simulation results agree with the theoretical ones. During the initial period, a larger forgetting factor is generated due to the poor fit of the model with the actual process. Normal Kalman filter starts to degrade, but AFKF still performs well through adaptively adjusting the forgetting factors. After a while, the system closes to its new steady state, and the difference between the actual state and the estimate becomes smaller. As a result, the forgetting factor returns to its normal value—unity. Figure 3 shows the time evolution of the forgetting factors for Case 1. Sorenson and Sacks (1971) indicated that if the model discrepancy is very significant, the fading factor will be very large in order to recover the filter from divergence. The forgetting factors for Case 2 have a similar behavior. As predicted theoretically, AFKF has satisfactory dynamic performance, and the residuals of state estimate are almost eliminated. It can also be found that the performance of Algorithms 1–3 are nearly the same.

5. *Application to paper machine headbox*

We have applied the AFKF algorithm to a real world paper machine in a paper Mill in the P.R.C., which produces super-thin condenser paper. The headbox section, as shown in Fig. 4, is an essential part of the paper machine. Since the paper machine has slow machine rate (70 m min^{-1}), its headbox is not pressurized, but open to atmosphere. The main purpose of the headbox is to distribute the water/fiber suspension onto the wire as evenly as possible: thick stock from the machine chest is diluted by white water to form thin

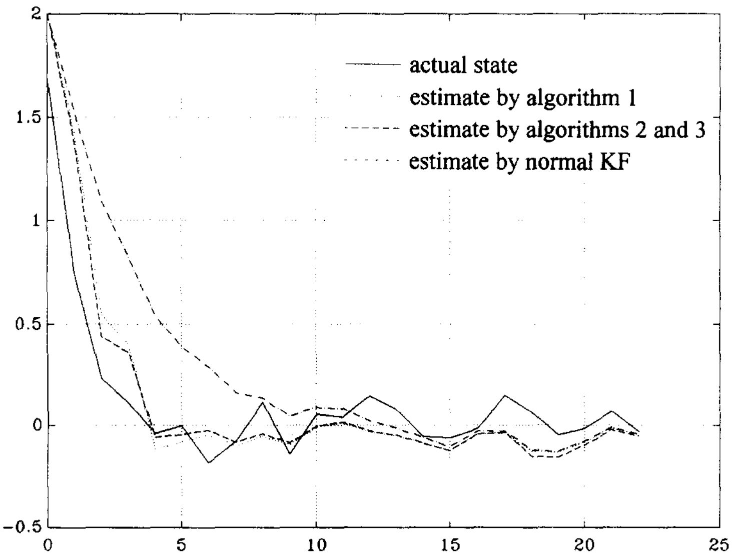

FIG. 1. Simulation results with system model errors.

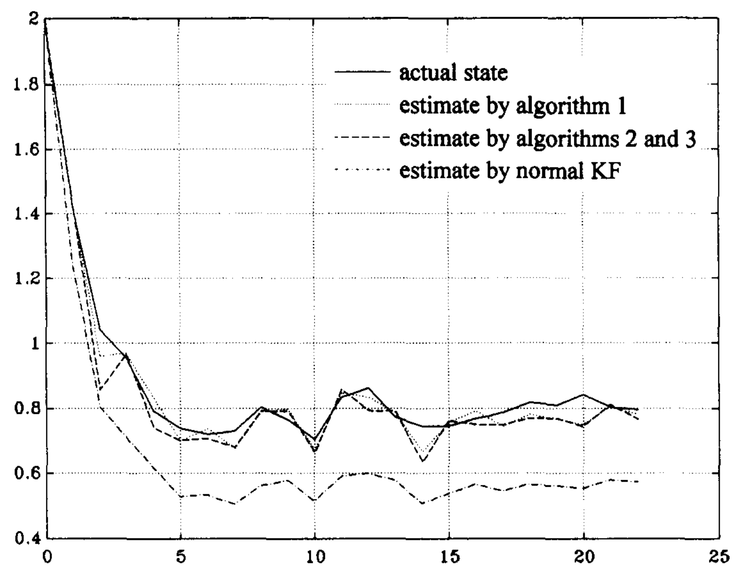

FIG. 2. Simulation results with unknown drifts.

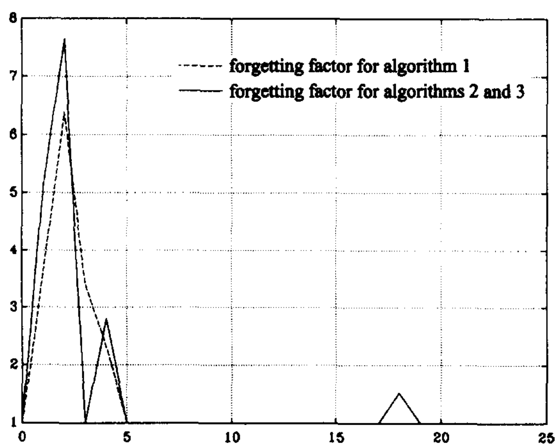

FIG. 3. Time evolution of the forgetting factor in Case 1.

stock which then flows onto the wire. The flow-rate of both thick stock and white water is controlled.

The dry basis weight of paper sheet on reel is an important quality property, which varies with the flow-rate and the consistency of stock onto the wire. However, the measurement sensor for basis weight is not available in the paper machine. The purpose of using AFKF is to obtain an estimate of dry basis weight and other states to implement control algorithm.

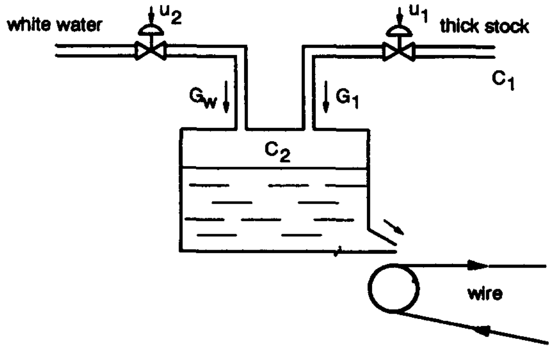

FIG. 4. Principle diagram of headbox section.

The model of the headbox section of the paper machine can be described by (Xia, 1989)

$$G_1(k+1) = 0.8667G_1(k) - 0.0344u_1(k-1) \tag{61}$$

$$G_w(k+1) = 0.8667G_w(k) - 0.6877u_2(k-1) \tag{62}$$

$$C_2(k+1) = 0.9099C_2(k) + 0.1069G_1(k) - 0.0503G_w(k) + 0.0788C_1(k-2) \tag{63}$$

$$B_w(k) = 4.089G_1(k-3) + 1.4910G_w(k-3) + 1.7533C_2(k-3), \tag{64}$$

where C_1, C_2, G_1, G_w and B_w are the consistency of thick stock, consistency of thin stock, flow-rate of thick stock, flow-rate of white water, and dry basis weight, respectively. u_1 and u_2 are the changes of the openings of thick stock and white water control valves. The sampling interval is 20 s, and G_1, C_1 and C_2 can be measured on-line.

Denoting

$$x(k) = [G_1(k) \quad G_w(k) \quad C_2(k)]^T,$$

$$y(k) = [G_1(k) \quad C_2(k)]^T, \quad u(k) = [u_1(k-1) \quad u_2(k-1)]^T$$

and $d(k) = C_1(k-2)$ as state vector, output vector, control vector and disturbance vector of the system, respectively, and adding input noise $w(k)$ and measurement noise $v(k)$ to the model, then the state space model of headbox section is written as

$$x(k+1) = \begin{pmatrix} 0.8667 & 0 & 0 \\ 0 & 0.8667 & 0 \\ 0.1069 & -0.0503 & 0.9099 \end{pmatrix} x(k) + \begin{pmatrix} -0.0344 & 0 \\ 0 & -0.6877 \\ 0 & 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ 0 \\ 0.0788 \end{pmatrix} d(k) + w(k) \tag{65}$$

$$y(k) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x(k) + v(k). \tag{66}$$

Algorithm 3 is used to estimate the dry basis weight of paper sheet on reel. Figure 5 shows the change of actual and estimated basis weights when the opening of thick stock valve has a step change. Since there is no on-line measurement for basis weight, the actual basis weight is obtained by manually testing the paper sheet samples picked

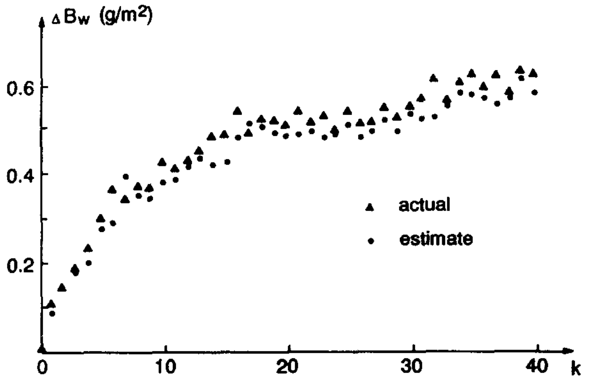

FIG. 5. Basis weight with step change in thick stock valve.

up from the dry end of the paper machine. The filter chooses covariances $Q(k) = \text{diag}(0.4^2, 0.4^2, 1)$, $R(k) = \text{diag}(0.3^2, 0.4^2)$, $P(0) = \text{diag}(0.8^2, 0.8^2, 0.8^2)$. It is seen that the actual and estimated basis weights match very well. AFKF has also been applied to a basis weight and moisture control system for the paper machine. The computer control system has been in operation reliably and satisfactorily.

6. *Conclusions*

In this paper, an adaptive fading Kalman filter algorithm for state estimation is proposed. The algorithm improves both optimality and convergence. The filter uses the variable exponential weighting approach to compensate the model errors and unknown drifts. Since there is only one adjustable factor in the algorithm, complete optimality may not be ensured. In the case where higher degree of optimality is required, matrix forgetting factor should be considered instead of scalar factor in order to provide different rates of fading for different filter channels.

References

Bialkowski, W. L. (1983). Applications of Kalman filters to the regulation of dead time processes. *IEEE Trans. Automatic Control*, **AC-28**, 400–406.

Chang, C.-B. and J. A. Tabaczynski (1984). Application of state estimation to target tracking. *IEEE Trans. Automatic Control*, **AC-29**, 98–109.

Deyst, J. J. Jr and C. F. Price (1968). Conditions for asymptotic stability of the discrete minimum-variance linear estimator. *IEEE Trans. Automatic Control*, **AC-13**, 702–705.

Fagin, S. L. (1964). Recursive linear regression theory, optimal filter theory, and error analysis optimal system. *IEEE Int. Convention Record*, **12**, 216–240.

Fitzgerald, R. L. (1971). Divergence of the Kalman filter. *IEEE Trans. Automatic Control*, **AC-16**, 736–747.

Kalman, R. E. and R. S. Bucy (1961). New results in linear filtering and prediction theory. *Trans. ASME J. Basic Engng*, **83**, 95–108.

Masreliez, C. J. and R. D. Martin (1977). Robust Bayesian estimation for the linear model and robustifying the Kalman filter. *IEEE Trans. Automatic Control*, **AC-22**, 361–371.

Maybeck, P. S. (1982). *Stochastic Models, Estimation, and Control*, Vol. 1–3. Academic Press, NY.

Ohap, R. H. and A. R. Stubberud (1976). Adaptive minimum variance estimation in discrete time linear systems. In C. T. Leondes (Ed.), *Control and Dynamic Systems*, Vol. 2, pp. 583–642. Academic Press, NY.

Polak, E. (1971). *Computational Methods in Optimization: a Unified Approach*. Series in Mathematics in Science and Engineering, Vol. 77. Academic Press, NY.

Shellenbarger, J. C. (1966). Estimation of covariance parameter for an adaptive Kalman filter. *Proc. National Electronics Conference*, pp. 698–702.

Sorenson, H.-W. and J. E. Sacks (1971). Recursive fading memory filtering. *Information Sciences*, **3**, 101–119.

Synder, D. L. (1973). Information processing for observed jump processes. *Information and Control*, **22**, 69–75.

Tsai, C. and L. Kurz (1983). An adaptive robustizing approach to Kalman filtering. *Automatica*, **19**, 279–288.

Tze-Kwarfung, P. and M. J. Grimble (1983). Dynamic ship positioning using a self-tuning Kalman filter. *IEEE Trans. Automatic Control*, **AC-28**, 339–350.

Xia, Q. J. (1989). Reliable control theory and application in industrial processes. Ph.D. dissertation, Zhejiang University, Hangzhou, P.R.C.

Ydstie, B. E. and T. Co (1985). Recursive estimation with adaptive divergence control. *IEEE Proc., Pt.D*, **132**, 124–130.