MODELING, IDENTIFICATION AND CONTROL, 1994, VOL. 15, NO. 2, 109–121
doi:10.4173/mic.1994.2.4

Recursive prediction error methods for online estimation in nonlinear state-space models

DAG LJUNGQUIST† and JENS G. BALCHEN‡

Keywords: *Recursive estimation; line-search methods; recursive prediction error methods; system identification.*

Several recursive algorithms for online, combined state and parameter estimation in nonlinear state-space models are discussed in this paper. Well-known algorithms such as the extended Kalman filter and alternative formulations of the recursive prediction error method are included, as well as a new method based on a line-search strategy. A comparison of the algorithms illustrates that they are very similar although the differences can be important for the online tracking capabilities and robustness. Simulation experiments on a simple nonlinear process show that the performance under certain conditions can be improved by including a line-search strategy.

1. Introduction

Modeling from fundamental physical laws and principles is of growing importance to the process industry, and the established knowledge has reduced the time needed to develop a 'first principles' model to be used for control purposes. Online estimation in the resulting model, usually a nonlinear state-space model, is needed when the model is used in an advanced instrumentation system or when the model is used online in the controller.

The basic algorithms presented in this paper are the Extended Kalman Filter (EKF), see for example Anderson and Moore (1979), and a recursive Gauss–Newton algorithm which computes an approximate of the maximum likelihood estimate (Ljung and Söderström 1983). The latter algorithm is often referred to as *the* recursive prediction error method (RPEM), although the term 'recursive prediction error methods' actually refers to a *family* of methods. The EKF used for combined state and parameter estimation will be denoted as the Augmented Kalman Filter (AKF) since the vector of state variables is augmented to include the unknown parameters. Both the AKF and the RPEM are widely used and thoroughly discussed in the literature, and with the exception of higher order methods there exists no recursive algorithm which is proven generally to have better convergence and robustness properties.

Due to the notational differences, the close relationship between the AKF and the RPEM is not always apparent. The similarity is to some extent illustrated in Ljung and Söderström (1983) (Appendix 3.C) and Ljung (1979). Moreover, Ljung ((1987), p. 314) points out that the AKF itself is a recursive prediction error method. The comparison

Received 8 December 1993.

† Hydro Aluminium a.s., Årdal Metallverk, N-5870 Øvre Årdal, Norway.

‡ Department of Engineering Cybernetics, Norwegian Institute of Technology, N-7034 Trondheim, Norway.

given here is based on Ljungquist (1990) where a unifying notation is introduced to emphasize the similarity.

Line-search techniques are widely used to solve optimization problems and dual offline estimation problems. A family of online estimation algorithms which may be denoted as moving-horizon batch methods (see Ljungquist (1990)) makes use of line-search techniques as well. To the authors' knowledge, however, line-search techniques were not used in recursive estimation algorithms until a strategy was proposed by Ljungquist (1990) to improve the robustness and the transient performance.

First, a problem formulation is given. Then results the of an AKF-RPEM comparison are summarized together with the RPEM in a unified notation. Next a recursive line-search method (RLM) is presented. Simulation results from a simple continuous stirred-tank reactor (CSTR) conclude the paper.

2. Problem formulation

Throughout this paper it is assumed that the system under consideration can be described by the following model formulation

$$\begin{aligned} x(k+1) &= f(x(k), \theta(k), u(k) + v_1(k) \quad & v_1 \sim N(0, V_1(k)) \\ y(k) &= g(x(k), \theta(k), u(k)) + w(k) \quad & w \sim N(0, W(k)) \\ & & E(v_1(k)w(l)^{\mathrm{T}}) = Z(k)\delta_{kl} \end{aligned} \tag{1}$$

where

x	vector of state variables, dimension n
k	discrete time
$f(.)$	vector of nonlinear functions, dimension n
u	vector of control variables, dimension r
θ	vector of time-varying parameters, dimension p
v_1	vector of process noise, dimension n
y	vector of measurements, dimension m
$g(.)$	vector of nonlinear measurement functions, dimension m
w	vector of measurement noise, dimension m

Moreover, $v_1 \sim N(0, V_1)$ is a shorthand notation for a normally distributed noise process with the properties $E[v_1(k)] = 0$ and $E[v_1(k)v_1^{\mathrm{T}}(l)] = V_1\delta_{kl}$, where δ_{kl} is the Kronecker delta. Additional notation is given at the end of the paper. The time-varying parameters are modeled as random walk variables, that is

$$\theta(k+1) = \theta(k) + v_2 \quad v_2 \sim N(0, V_2(k)) \tag{2}$$

The nonlinearities of the functions $f(.)$ and $g(.)$ are assumed to be such that approximations of the sensitivities with respect to the states and the parameters can be computed.

The goal of the estimation algorithm is to adjust the model, state and parameter vectors in an optimal way. In order to do this, 'optimality' has to be defined by means of a criterion. Since different criteria are used to derive the different algorithms discussed here, it may seem difficult to make a fair comparison. However, the structures of the algorithms are very similar to each other. The comparison can therefore be based on the importance of the differences to the transient responses and to the robustness/convergence properties.

3. An AKF–RPEM comparison

It is probably not an overstatement to assert that the Kalman filter represents the most widely applied and demonstrably useful result to emerge from the state variable approach of 'modern control theory' (Introduction in Sorensen (1985)). However, from the mid 1960s it has been a well-known fact that the behaviour of the Kalman filter in some cases is sensitive to the *a priori* statistics assumptions (Soong 1965), and that divergence may result if the initial state and parameter estimates are not sufficiently good (Schlee *et al.* 1967). Based on averaging theory and weak convergence theory the asymptotic behaviour of the AKF and the RPEM has been analysed in both discrete (Ljung 1979) and continuous (DeWolf and Wiberg 1993) time. The result in both cases is that the AKF fails, with probability one, to converge to the true values of the parameters of a system *whose process noise covariance is unknown* while the RPEM, with probability one, converges to the maximum of the likelihood function. Although asymptotic convergence is a desired property, the question is how relevant the underlying assumptions are to practical implementation of an online estimation scheme:

- The proofs are based on unprojected trajectories, an assumption that is frequently violated.
- The presence on nonlinearities will, generally, make the maximum likelihood estimate biased.
- The idealistic noise characteristics on which the proofs are based, are often violated.

From a practical point of view, the main difference is the choice of tuning-factors, and it is often difficult to tell beforehand from theoretical considerations which algorithm will work best. For example, is it important to estimate the process noise covariance or is the main purpose to obtain reasonably good estimates of the model parameters?

The results of a detailed algorithmic AKF–RPEM comparison are summarized in Table 1. In the fourth column the most essential quantities of the RPEM are given in Ljung and Söderström's (1983) notation, while the corresponding quantities are given

AKF		RPEM	
Traditional notation	Unifying notation	Unifying notation	Traditional notation
X_2	Θ	Θ	P
X_{12}	$X_{x\theta}$	$\frac{dx}{d\theta}\Theta$	—
X_1	X	$\frac{dx}{d\theta}\Theta\frac{dx^{\mathrm{T}}}{d\theta}$	—
V	V	λ	λ
W	W	$\hat{\mathscr{E}}$	$\hat{\Lambda}$
K_1	K_x	K_x	$K(\theta)$
K_2	K_θ	K_θ	L
—	$\frac{dy}{d\theta}$	$\frac{dy}{d\theta}$	Ψ^{T}

Table 1. AKF–RPEM comparison.

in one of many notations used for the AKF in the first column. In the two columns in the middle a unified notation is summarized. Note that the state-parameter covariance corresponds to the product of the state sensitivity matrix and the parameter covariance matrix, and that the state covariance matrix corresponds to the state sensitivity matrix times the parameter covariance matrix times the transposed state sensitivity matrix. At first sight the second approximation seems very rough, and asymptotically it is incorrect since the parameter covariance matrix tends to zero while the state covariance matrix does not. However, these results have been used in Ljungquist (1990) to obtain the following approximate expression of the state-updating matrix

$$K_x \simeq \frac{dx}{d\theta}\Theta\frac{dx^{\mathrm{T}}}{d\theta}D_x^{\mathrm{T}}\left(D_x\frac{dx}{d\theta}\Theta\frac{dx^{\mathrm{T}}}{d\theta}D_x^{\mathrm{T}}+\hat{\mathscr{E}}\right)^{-1} \tag{3}$$

The RPEM with the parameterized state-updating matrix replaced by Eqn. (3) and the forgetting factor $\lambda=1$, turned out to result in an estimator *almost* identical to the AKF with $V_1=V_2=0$.

According to Table 1 $\hat{\mathscr{E}}$ plays the same role as the measurement noise covariance matrix W in the AKF. This may seem odd, but it is a natural consequence of the fact that the Kalman filter equation can be derived by solving

$$\left.\frac{dJ_1(k)}{dx(k)}\right|_{x(k)=\hat{x}(k)}=0 \tag{4}$$

where

$$J_1(k)=(x(k)-\bar{x}(k))^{\mathrm{T}}\bar{X}^{-1}(k)(x(k)-\bar{x}(k))+\varepsilon^{\mathrm{T}}(k)W^{-1}(k)\varepsilon(k) \tag{5}$$

while the RPEM results from minimizing the criterion

$$J_2(k)=\frac{\gamma(k)}{2}\sum_{j=1}^{k}\beta(k,j)\varepsilon^{\mathrm{T}}(j)\hat{\mathscr{E}}^{-1}(j)\varepsilon(j), \qquad \gamma(k)=\left(\sum_{j=1}^{k}\beta(k,j)\right)^{-1} \tag{6}$$

where β is a scalar data weighting factor, by a recursive Gauss–Newton algorithm.

The fact that $\hat{\mathscr{E}}$ can be estimated in a natural way makes the RPEM more flexible than the AKF and this may improve the convergence. A frequently used estimate is

$$\hat{\mathscr{E}}=\hat{\mathscr{E}}(k-1)+\alpha(\varepsilon\varepsilon^{\mathrm{T}}-\hat{\mathscr{E}}(k-1)) \tag{7}$$

where α is a tuning factor often chosen in accordance with λ. Then $\alpha=1/N$ corresponds to $\lambda=(N-1)/N$ and $\alpha=1/k$ corresponds to $\lambda=1{\cdot}0$ (no forgetting).

Another difference which becomes evident from the comparison is that the sensitivity matrix of the state-updating matrix with respect to the parameters is not included in the AKF. This has been known for a long time, and Ljung (1979) showed that this is the reason why the AKF fails to converge. Ljung also suggested a modified Kalman filter to cope with the problem and proved convergence of the resulting algorithm. The RPEM on which the comparison in Table 1 is based, is derived from an innovations state-space system representation. This means that in principle, the algorithm is based on the assumption that the state-updating matrix, K_x, is constant at least asymptotically. For nonlinear systems, K_x will vary rapidly as the inputs and the operational point change, and the RPE algorithm will not compute the K_x which minimizes the variance of the states. On the other hand, simulation results show that the state-updating matrix which minimizes the variances of the states, makes it harder to detect parameter changes. Hence, the algorithm which gives the most 'optimal' K_x is not necessarily the best choice for solving a given problem. This is especially so when good tracking capabilities and long-range predictive capabilities are important.

A drawback with the RPEM for an innovations model is that the parameterization of K_x results in an estimation problem with many parameters. The RPE algorithm for the general state-space model (Appendix 3.B in Ljung and Söderström (1983)) can be applied to reduce this number. The main drawback of this algorithm, as well as of the modified Kalman filter, is that computation of the sensitivities of K_x with respect to the parameters increases the computational requirements significantly compared to the AKF and the RPEM based on an innovations model.

The third main difference between the AKF and the RPEM is the way of avoiding the parameter covariance matrix tending to zero, which is an important tuning factor in online estimation. Simulation results indicate that a scalar forgetting factor should never be used for online estimation in state-space models except, perhaps, in *very* simple cases. The reasons are that when insufficient information enters into the system, the forgetting factor results in exponentially increasing elements of the parameter covariance matrix, and that all the elements are increased by the same percentage. The latter problem can be reduced by using a vector forgetting factor. However, the RPEM with additive parameter noise given in Table 2 has increased tuning flexibility. Moreover, it is more robust than the RPEM with a forgetting factor since usually only the diagonal elements of the covariances matrix are updated and since the covariance matrix increases only linearly when insufficient information enters into the system. Note that in Table 2 W is used instead of the estimate in Eqn. (7). The main reason is that simulation experience indicates that the robustness is reduced considerably without a corresponding improvement of the transient response when the estimate in Eqn. (7) is used.

$$
\begin{aligned}
\bar{x}(k+1) &= f(\hat{x}, \hat{\theta}, u) \\
\bar{\theta}(k+1) &= \hat{\theta} \\
\bar{X}_\theta(k+1) &= \hat{F}_x \hat{X}_\theta + \hat{F}_\theta \\
\bar{\Theta}(k+1) &= \hat{\Theta} + V_2 \\
\bar{y} &= g(\bar{x}, \bar{\theta}, u) \\
\varepsilon &= y - \bar{y} \\
Y_\theta &= \bar{D}_x \bar{X}_\theta + \bar{D}_\theta \\
S &= Y_\theta \bar{\Theta} Y_\theta^{\mathrm{T}} + W \\
K_x &= K_x(\bar{\theta}) \\
K_\theta &= \bar{\Theta} Y_\theta^{\mathrm{T}} S^{-1} \\
\hat{x} &= \bar{x} + K_x \varepsilon \\
\hat{\theta} &= \bar{\theta} + K_\theta \varepsilon \\
\hat{\Theta} &= (I_p - K_\theta Y_\theta) \bar{\Theta} (I_p - K_\theta Y_\theta)^{\mathrm{T}} + K_\theta W K_\theta^{\mathrm{T}} \\
\hat{X}_\theta &= (I_n - K_x \bar{D}_x) \bar{X}_\theta + \frac{d\hat{K}_x}{d\theta} \varepsilon
\end{aligned}
$$

Table 2. RPEM with additive parameter noise.

From the above discussion it is obvious that the algorithms can be modified to become almost identical to each other (a more detailed discussion is given in Ljungquist (1992)). The main difference is the choice of tuning factors. Which of these algorithms should be used to solve a specific problem depends on the application as well as the available *a priori* information.

4. A recursive line-search method

Alternative recursive line-search algorithms in both discrete and continuous-discrete mode have been presented by Ljungquist (1990, 1992). Here an algorithm which is very close to the RPEM given in Table 2 will be discussed. In this algorithm the optimal parameter-updating step is computed by minimizing the one-sample criterion

$$J_1^j(k) = (\varepsilon^j(k))^{\mathrm{T}}\hat{\mathscr{E}}^{-1}\varepsilon^j(k) + \gamma(\Delta\theta^j)^{\mathrm{T}}\bar{\Theta}^{-1}(k)\Delta\theta^j \tag{8}$$

where

$$\Delta\theta^j = \theta^j(k) - \hat{\theta}(k-1)$$

$\bar{\Theta}$ is a symmetric positive definite matrix, dimension $p \times p$

The scalar tuning factor γ will be discussed below. During the line search, a new parameter vector is computed according to

$$\theta^j(k) = \theta(k-1) + \alpha^j(k)L(k) \tag{9}$$

Assume that the search direction $L(k)$ is computed. Different stepsizes will then result in different parameters and different prediction errors. Since a one-sample criterion is used, the criterion can easily be computed for each stepsize, and the criterion is a function of the stepsize alone. A one-dimensional optimization algorithm, a line-search algorithm, can then be used to compute the optimal stepsize. In this way errors in the generated search direction are detected, and stepsizes which are too long or too short are avoided. The main problem is to establish a good one-sample criterion since it tends to be sensitive to measurement noise. The problem is solved by weighting the parameter change in the criterion. Note the close relationship between the criterion in Eqn. (8) and the criterion in Eqn. (5) which may be used to derive the Kalman filter. The idea can be extended to include more samples, but more than five is not recommended. In this case the more general concept of moving-horizon batch methods should be used.

Different stepsize selection procedures can be used (Ljungquist (1990)). Here the simplest alternative will be applied. Let the optimal stepsize α^0 be defined as the stepsize which results in the smallest criterion J_1^0 for a given search direction L. This optimal value can be computed by a conventional line-search strategy. The stepsize is then chosen according to Table 3. In Table 3 $\underline{J}$ denotes the estimation activation threshold. A value $\underline{J} > 0$ is used to increase the robustness of online estimation schemes at the cost of a slower convergence. A discrete recursive line-search method based on the RPEM in Table 2 is summarized in Table 4. In Table 4 superscript ^{0} denotes initial values, which corresponding to a stepsize $\alpha = 0$. R is a counter which counts the number of successive times the optimal stepsize is equal to zero or equal to the predefined upper limit, α^m. If

If $J_1(\alpha=0) \leqslant \underline{J}$	then	$\alpha = 0{\cdot}0$
If $J_1(\alpha=0) > \underline{J}$	then	$\alpha = \max\{0{\cdot}0, \min(\alpha^0, \alpha^m)\}$

Table 3. The stepsize selection procedure.

(i) $\bar{x}^0(k)=f(\hat{x}^0(k-1),\hat{\theta}(k-1),u(k-1))$

$\bar{\theta}(k)=\hat{\theta}(k-1)$

$\bar{X}^0_\theta(k)=\hat{F}^0_x(k-1)\hat{X}^0_\theta(k-1)+\hat{F}^0_\theta(k-1)$

$\bar{\Theta}(k)=\hat{\Theta}(k-1)+V_2$

(ii) $\varepsilon^0(k)=y(k)-g(\bar{x}^0(k),\bar{\theta}(k),u(k))$

$Y^0_\theta(k)=\bar{D}^0_x(k)\bar{X}^0_\theta(k)+\bar{D}^0_\theta(k)$

(iii) Compute $J^0_1(k)$ according to Eqn. (8).

$K^0_\theta(k)=\bar{\Theta}(k)(Y^0_\theta)^{\mathrm{T}}(Y^0_\theta(k)\bar{\Theta}(k)(Y^0_\theta)^{\mathrm{T}}(k)+\hat{\mathscr{E}}(k-1))^{-1}$

(iv) $L(k)=K^0_\theta(k)\varepsilon^0(k)$

(v) If $J^0_1\leqslant \underline{J}$, set $\hat{\theta}^0(k)=\bar{\theta}(k)$ and go to (*xi*)

(vi) Set $j=1$. Choose an initial stepsize $\alpha^1(k)$.

(vii) $\hat{\theta}^j(k)=\bar{\theta}(k)+\alpha^j(k)L(k)$

$\Delta\theta^j=\hat{\theta}^j(k)-\bar{\theta}(k)$

(viii) $\hat{x}^j(k-1)=\bar{x}(k-1)+K_x(\hat{\theta}^j(k))\varepsilon(k-1)$

$\bar{x}^j(k)=f(\hat{x}^j(k-1),\hat{\theta}^j(k),\ u(k-1))$

(ix) $\varepsilon^j(k)=y(k)-g(\bar{x}^j(k),\hat{\theta}^j(k),u(k))$

Compute $J^j_1(k)$ according to Eqn. (8).

(x) $j=j+1$. Adjust the stepsize $\alpha^j(k)$ according to a line-search strategy and repeat steps (*vii*) through (*ix*) until the stepsize satisfies the conditions in Table 3.

(xi) If $\alpha^0(k)=0$ or $\alpha^0(k)=\alpha^m$ set $R=R+1$. Else set $R=0$

(xii) If $R\geqslant R_U$ set $\bar{X}_\theta(k-1)=0$ and $\bar{\Theta}(k)=\hat{\Theta}(0)$.

(xiii) $\hat{\theta}(k)=\bar{\theta}(k)+\alpha^0(k)L(k)$

$\hat{x}(k-1)=\bar{x}(k-1)+K_x(\hat{\theta}(k))\varepsilon(k-1)$

$$\bar{X}(k-1)=(I_n-K_x(\hat{\theta}(k))\bar{D}_x(k-1))\bar{X}_\theta(k-1)+\frac{d\hat{K}_x}{d\theta}(k-1)\varepsilon(k-1)$$

(xiv) $\bar{x}(k)=f(\hat{x}(k-1),\hat{\theta}(k),u(k-1))$

$\bar{X}_\theta(k)=\hat{F}_x(k-1)\hat{X}_\theta(k-1)+\hat{F}_\theta(k-1)$

$\varepsilon(k)=y(k)-g(\bar{x}(k),\hat{\theta}(k),u(k))$

$$\hat{\mathscr{E}}(k)=\hat{\mathscr{E}}(k-1)+\frac{1}{k}(\varepsilon(k)\varepsilon^{\mathrm{T}}(k)-\hat{\mathscr{E}}(k-1))$$

$Y_\theta(k)=\bar{D}_x(k)\bar{X}_\theta(k)+\bar{D}_\theta(k)$

$K_\theta(k)=\alpha^0(k)\bar{\Theta}(k)Y^{\mathrm{T}}_\theta(Y_\theta(k)\bar{\Theta}(k)Y^{\mathrm{T}}_\theta(k)+\hat{\mathscr{E}}(k))^{-1}$

$\hat{\Theta}(k)=(I_p-K_\theta(k)Y_\theta(k))\bar{\Theta}(k)(I_p-K_\theta(k)Y_\theta(k))^{\mathrm{T}}+K_\theta(k)\hat{\mathscr{E}}(k)K^{\mathrm{T}}_\theta(k)$

(xv) $\hat{x}^0(k)=\bar{x}(k)+K_x(\hat{\theta}(k))\varepsilon(k)$

$$\hat{X}^0_\theta(k)=(I_n-K_x(\hat{\theta}(k))\bar{D}_x(k))\bar{X}_\theta(k)+\frac{d\hat{K}_x}{d\theta}(k)\varepsilon(k)$$

(xvi) Set $k=k+1$ and go to (*i*).

Table 4. A recursive line-search method.

R_U is chosen equal to one, the sensitivity matrix and the search direction will be reset each time it is impossible either to obtain improvement in the computed search direction or the stepsize attains its upper limit. In the presence of large measurement noise, the algorithm will tend to be reset too often so that the estimate of the parameter covariance matrix, which corresponds to the inverse Hessian matrix, will be poor. The result is an algorithm with approximately first order convergence properties. On the other hand, a large R_U means that it will take a long time to detect errors in the computed search direction. A reasonable tradeoff is to choose $R_U \in \{1, 2, 3\}$. The estimate of the innovations covariance matrix $\mathscr{E}$ can be replaced by a constant matrix.

The tuning parameter γ is used to control the computed stepsize and reflects the confidence in the computed parameter covariance matrix compared to the prediction error covariance matrix. An increased γ will result in a reduced stepsize. Note that the same effect can be obtained by reducing the elements of the V_2 matrix. However, by changing γ instead of V_2, problems such as Θ tending to zeros or blow-up of Θ during periods when little new information is available, are reduced. Usually γ is chosen so that $\gamma \in [0{\cdot}1, 10{\cdot}0]$.

In the above RLM both the search direction and the Θ-matrix used in the criterion (Eqn. (8)) are computed according to the same equations as the RPEM in Table 2. Note that a constant matrix can be used instead of the recursively computed matrix, or a constant matrix can be added to the Θ-matrix before entering it into the criterion. The result is a regularized estimation problem and a more robust algorithm. The transient response, however, will tend to be worse.

It is pointed out in Ljungquist (1990, 1992) that it is usually easier to take into account boundaries in the parameters in recursive line-search methods than in ordinary recursive methods. For the RLM presented here, however, this is not the case.

The RLM given in Table 4 can be simplified by not recomputing the sensitivity matrix with the most recent parameter estimates in Steps (xiii) to (xiv). As a result the computational cost is reduced by approximately 20 per cent.

The main advantages of recursive line-search methods compared to ordinary recursive methods are that the robustness is increased and that the transient response is improved in some cases. The stepsize selection procedure can easily be modified in such a way that the RLM obtains the same asymptotic convergence properties as the RPEM. Loosely speaking, the stepsize must in average tend asymptotically to zero like $1/k$, where k is the discrete time. The RLM gives results very close to the RPEM in Table 2 when the model is linear, the noise assumptions are satisfied, and the parameter identifiability is low. The RLM will show better transient response when the parameter identifiability is high and the nonlinearities predominant as illustrated by the simulation example below. Results presented in Ljungquist (1990) indicate that nonlinearities in the measurement equations make the RLM preferable. The robustness is higher for the RLM since overlong updating steps during transients are avoided, and since it can more easily be tuned to obtain good tracking capabilities without lack of robustness. In summary, the drawbacks of the RLM are the increased complexity and the increased computational costs compared to ordinary recursive algorithms.

5. Simulation examples

In this section some of the above conclusions are illustrated by simulation experiments on a simple model of a continuous stirred tank reactor (CSTR). A simple model of a CSTR is summarized in Eqn. (10). The feed consists of an unknown concentration of component A and has a given temperature. Component A reacts endothermically in the reactor A→B, and the heat required for the reaction is supplied through a heater. Assuming constant volume of the contents of the reactor, the following equations result from elementary mass and energy balances:

$$\frac{dC_A}{dt}=\frac{q}{V}(C_{Af}-C_A)-r, \qquad r=k_cC_A\exp\left(-\left(\frac{1}{T}-\frac{1}{T_c}\right)\frac{E}{R}\right)$$
$$\frac{dT}{dt}=\frac{q}{V}(T_f-T)+\frac{1}{c_{pm}C_{Af}V}[(-\Delta H_r)rV+Q] \tag{10}$$

The following notation is used

C_A	concentration of component A in the reactor	$(0{\cdot}3[\mathrm{mol/l}])$
C_{Af}	concentration of component A in the feed	$(\theta_1[\mathrm{mol/l}], \theta_1^0=1{\cdot}0)$
T	temperature in the reactor	$(0{\cdot}414\cdot10^3[\mathrm{K}])$
T_f	temperature in the feed	$(0{\cdot}350\cdot10^3[\mathrm{K}])$
Q	energy supply	$(3{\cdot}1\cdot10^8[\mathrm{J/mol/min}])$
ΔH_r	heat of reaction	$(\theta_4\cdot10^6[\mathrm{J/mol}], \theta_4^0=0{\cdot}671)$
E	activation energy	$\left(\frac{E}{R}=\theta_3\cdot10^4[\mathrm{K}], \theta_3^0=1{\cdot}0\right)$
R	universal gas constant	
c_{pm}	heat capacity of A and B	$(3{\cdot}35\cdot10^3[\mathrm{J/K/mol}])$
k_c	reaction rate constant	$(\theta_2\cdot10^{-1}[\mathrm{min}^{-1}], \theta_2^0=1{\cdot}515)$
T_c	reference temperature	400[K])
V	volume of the reactor	$(3{\cdot}0\cdot10^3[\mathrm{l}])$
q	feedrate	$\left(\frac{q}{V}=0{\cdot}152[\mathrm{min}^{-1}]\right)$

The model was discretized with a stepsize equal to 1/3 min and scaled to be numerically more robust. All the estimation algorithms under consideration were tuned in the same way. The four parameters θ_1 to θ_4 were estimated, and identifiability examination verified that all four parameters can be estimated simultaneously provided that the energy supply is an approximate pseudo-random binary signal with minimum value $2{\cdot}48\cdot10^8$, maximum value $3{\cdot}72{\cdot}10^8$, minimum switching interval 1 min and maximum switching interval 25 mins (sufficient condition). However, the parameter identifiability was low, so the parameter convergence was rather slow even for well-tuned algorithms. Process and measurement noise was added to the simulated process, and the signal to noise ratio was approximately 10. Further details can be found in Appendix A in Ljungquist (1992).

In Fig. 1 the estimated parameters for an RPEM with a scalar forgetting factor and the RPEM in Table 2 are compared. Parameter number 3, θ_3 was increased by 1 after 50 samples, and θ_1 was increased by 0·4 after 200 samples. Both algorithms converged to the true parameters, but the transient responses were slower when a scalar forgetting factor was used. The parameter variances corresponding to the different forgetting strategies in Fig. 1 (not included) showed that the forgetting factor used ($\lambda=0{\cdot}98$) was too low for online implementation since the variances increased exponentially. The RPEM in Table 2 was tuned so that the parameter variances reached a constant level with an average value 2–10 times their absolute values after approximately 300 samples. It is clear that a forgetting based on additive parameter noise (Eqn. (2)) is superior to a scalar forgetting factor.

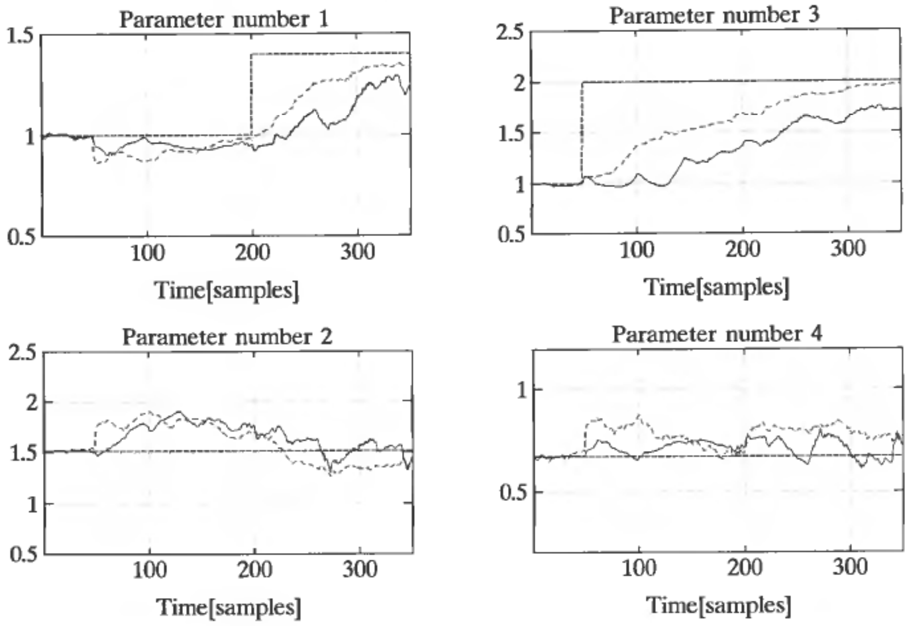

Figure 1. A comparison of the RPEM $w/\lambda=0{\cdot}98$ and the RPEM w/V_2. Solid lines—parameters estimated by the RPEM $w/\lambda=0{\cdot}98$, dashdotted (-·) lines—parameters estimated by the RPEM w/V_2, dashed (- -) lines—process parameters.

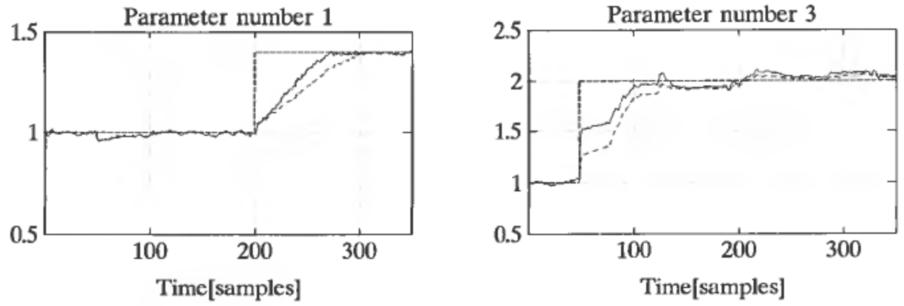

Figure 2. A comparison of the RLM and the RPEM in Table 2. Solid lines—RLM estimates dashdotted (-·) lines—RPEM estimates, dashed (- -) lines—process parameters.

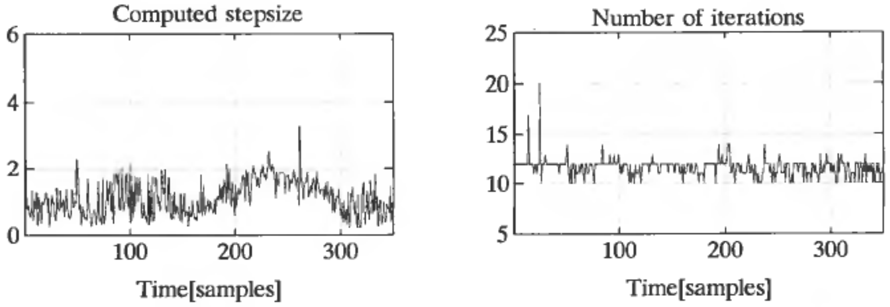

Figure 3. The stepsize and the number of iterations for the RLM.

When four parameters were estimated simultaneously, both the AKF and the RLM in Table 4 gave responses very similar to those of the RPEM with additive parameter noise. The parameterized state-updating matrix computed by the RPEM, however, behaved quite differently from the 'optimal' K_x computed by the AKF. The optimality of K_x is consequently not always an important issue.

When the parameter identifiability was increased by reducing the parameter set to include only θ_1, θ_3 and the parameterized state-updating matrix, the RLM with $\gamma = 0{\cdot}5$ and constant $\mathscr{E}$ resulted in better transient response than the RPEM as illustrated in Fig. 2. When V_2 was increased by a factor of two in the RPEM, the responses became closer to that of the RLM. The robustness of the RPEM, however, was then clearly reduced due to the increasing parameter covariance matrix.

The computed stepsize and the number of iterations performed in the RLM are shown in Fig. 3. The average stepsize increases for some time after the parameter changes. The number of iterations used in this simulation was 12 on average. This number can be reduced to about 8 by using the line-search algorithm described in Wolfe (1978) instead of the Matlab function *fmin* used in the simulations presented here.

The simulation experiments carried out in this section were very idealistic since the simulated process and the model were identical except for the noise added to the process, the measurement equations were linear, no parameter constraints were present, and the number of samples was low. The fact that the RLM was never reset (R was never $\geqslant R_U = 2$, see Table 4) illustrates that the simulation experiments did not fully exploit the advantages which result from the increased robustness of the RLM.

6. Conclusions

Simulations have shown that the algorithms under consideration here can be tuned to give very similar results and an algorithmic comparison shows that the main difference lies in the choice of tuning factors. It is consequently impossible to state that one algorithm is always preferable to another. The degree of the nonlinearities and the available *a priori* information are very important in order to select the best method for a given application and to tune the algorithm as well as possible. On the other hand, it is very difficult to construct examples for which the AKF or the RPEM outperform the RLM. The increased robustness of the RLM is especially important when the estimation scheme is to be applied online to real data due to the built-in detection of errors in the computed updating direction and the estimation activation threshold. In summary, the only drawback of the RLM is that the computational requirement is increased by a factor of three compared to the RPEM.

Notation

The notation used in addition to the problem formulation and the symbols in the simulation example are summarized below:

$$D_x = \frac{\partial g}{\partial x}$$

$$D_\theta = \frac{\partial g}{\partial \theta}$$

ε prediction error/model—system measurement mismatch

$E(.)$ the expected value of (.)

$$F_x = \frac{\partial f}{\partial x}$$

$$F_\theta = \frac{\partial f}{\partial \theta}$$

I_n identity matrix of dimension $n \times n$
K_x state update matrix
K_θ parameter update matrix
$X = \mathrm{cov}(x)$
$X_{x\theta} = E(\tilde{x}\tilde{\theta}^{\mathrm{T}})$ where $\tilde{x}$ is the state estimate error

$\tilde{\theta}$ is the parameter estimate error

$$X_\theta = \frac{dx}{d\theta}$$

$$Y_\theta = \frac{dy}{d\theta}$$

$$\Theta = \mathrm{cov}\,(\theta)$$

In addition, the bar '‾' and the hat '^' are used to distinguish between *a priori* and *a posteriori* estimates:

$$\bar{x}(k) = x(k|k-1)$$

$$\hat{x}(k) = x(k|k)$$

$$\bar{X}(k) = E((x-\bar{x})(x-\bar{x})^{\mathrm{T}})$$

$$\hat{X}(k) = E((x-\hat{x})(x-\hat{x})^{\mathrm{T}})$$

A similar notation is also used for the other matrices above. This means that

$$\bar{D}_x(k) = \left.\frac{\partial g(x,\theta,u)}{\partial x}\right|_{\substack{x=\bar{x}(k)\\ \theta=\bar{\theta}(k)}}$$

The following vector notation is used in the above definitions

$x = [x_1 x_2 \ldots x_n]^{\mathrm{T}}$ where x_i are scalars

$$\frac{dx}{d\theta} = \begin{vmatrix} \frac{dx_1}{d\theta_1} & \frac{dx_1}{d\theta_2} & \cdots & \frac{dx_1}{d\theta_p} \\ \frac{dx_2}{d\theta_1} & \frac{dx_2}{d\theta_2} & \cdots & \frac{dx_2}{d\theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dx_n}{d\theta_1} & \frac{dx_n}{d\theta_2} & \cdots & \frac{dx_n}{d\theta_p} \end{vmatrix} \quad \text{where } \theta_i \text{ are scalars}$$

Acknowledgments

This work was sponsored by the Royal Norwegian Council for Scientific and Industrial Research and the Norwegian State Oil Company (Statoil).

References

ANDERSON, B. D. O. and MOORE, J. B., (1979), *Optimal Filtering* (Prentice Hall, Englewood Cliffs, NJ).

DE WOLF, D. G., and WIBERG, D. M. (1993), An ordinary differential equation technique for continuous-time parameter estimation, *IEEE Trans. Automat. Contr.*, **34,** 514–528.

LJUNG, L. (1979). Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Trans. Aut. Control*, **24,** 36–50.

LJUNG, L. (1987). *System Identification, Theory for the User* (Prentice-Hall, Englewood Cliffs, New Jersey).

LJUNG, L. and SÖDERSTRÖM, T. (1983). *Theory and Practice of Recursive Identification* (MIT Press, Cambridge, Massachusetts).

LJUNGQUIST, D. (1990). Online Estimation in Nonlinear State-Space Models with Application to Catalytic Cracking, Dr. ing. thesis, Report no. 90-89-W, Department of Engineering Cybernetics, NTH, Trondheim, Norway.

LJUNGQUIST, D. (1992). Recursive Estimation Algorithms for Nonlinear State-Space Models. Statoil report no. 761.01-01, Trondheim, Norway.

SCHLEE, F. H., STANDISH, C. J. and TODA, N. F. (1967). Divergence in the Kalman filter. *AIAA Journal*, **5,** 1141–1120.

SORENSEN, H. W. (Ed.) (1985). *Kalman Filtering: Theory and Application* (IEEE Press, The Institute of Electrical and Electronics Engineers, Inc., New York).

SOONG, T. T. (1965). On a priori statistics in minimum-variance estimation problems, *Trans. of the ASME, Journal of Basic Engineering.*

WOLFE, M. A. (1978). *Numerical Methods for Unconstrained Optimization* (Van Nostrand Reinhold, Berkshire, England).