Available online at www.sciencedirect.com
SCIENCE DIRECT®
Future Generation Computer Systems 21 (2005) 1047–1055
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

On the construction of a reduced rank square-root Kalman filter for efficient uncertainty propagation

Dimitri Treebushny [a,*], Henrik Madsen [b]

[a] *Institute of Mathematical Machines and System Problems, NAS Ukraine, Prospekt Glushkova 42, Kiev, Ukraine*
[b] *DHI Water and Environment, Agern Allé 11, DK-2970 Horsholm, Denmark*

Available online 2 April 2004

Abstract

The Kalman filter is a sequential estimation procedure that combines a stochastic dynamical model with observations in order to update the model state and the associated uncertainty. In the situation where no measurements are available the filter works as an uncertainty propagator. The most computationally demanding part of the Kalman filter is to propagate the covariance through the dynamical system, which may be completely infeasible in high-dimensional models. The reduced rank square-root (RRSQRT) filter is a special formulation of the Kalman filter for large-scale applications. In this formulation, the covariance matrix of the model state is expressed in a limited number of modes M. In the classical implementation of the RRSQRT filter the computational costs of the truncation step grow very fast with the number of modes ($>M^3$). In this work, a new approach based on the Lanzcos algorithm is formulated. It provides a more cost-efficient scheme and includes a precision coefficient that can be tuned for specific applications depending on the trade-off between precision and computational load.

Keywords: Kalman filter; Reduced rank square-root; RRSQRT; Lanczos algorithm

1. Introduction

The Kalman filter [6] is a sequential estimation procedure that combines a stochastic dynamical model with observations in order to update the model state and the associated uncertainty. The filter works with the mean and the covariance of the state vector and contains two steps: (i) model propagation and (ii) measurement update. If no measurements are available, the filter can be treated as an uncertainty propagator. The most computationally demanding part of the Kalman filter is to propagate the covariance through the dynamical system.

In absolute precision mathematics for low-dimensional, linear, Gaussian, dynamical systems the classical formulation of the Kalman filter works fine. Problems appear when we are interested in real-life applications, which involve finite computer precision, high-dimensional, non-linear system dynamics, and non-additive, non-Gaussian system noise.

* Corresponding author.
E-mail addresses: dima@env.com.ua (D. Treebushny), hem@dhi.dk (H. Madsen).

0167-739X/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2004.03.005

The question of development, choice and implementation of suboptimal Kalman filtering [12] in engineering applications has been investigated since the first work of Kalman [6]. Suboptimal Kalman filter procedures should fulfill the following requirements:

1. to be robust in finite precision arithmetic;
2. to effectively reduce the computational burden compared to a straightforward implementation of the classical Kalman filter, which may be completely infeasible in high-dimensional models;
3. to be able to deal with non-linear systems;
4. to be able to deal with non-additive and non-Gaussian noise.

An important step towards filter robustness was made by dealing with a square-root of the covariance matrix S, $S{\cdot}S^{\mathrm{T}} = P$ instead of the full covariance matrix itself P. This approach automatically preserves the property of P to be a covariance matrix:

$$P \geq 0 \quad \text{or} \quad \lambda_i \geq 0, \ \ \lambda_i \text{ is eigenvalue of } P. \tag{1}$$

Calculation of a square-root of a positive definite matrix P can be made by any known algorithm such as Cholesky decomposition or SVD decomposition of a symmetric matrix. Verlaan and Heemink in Ref. [16] remarked that the square-root approach has an additional important feature. Because the square-root matrix has a much smaller range of eigenvalues, or in other words a much smaller condition number, square-root-based algorithms are numerically more stable than the classical Kalman filter.

The second requirement, the filter being feasible for high-dimensional systems, leads to the logical construction:

$$P \approx S{\cdot}S^{\mathrm{T}}, \tag{2}$$

where a huge matrix $P \sim N \times N$ is approximated with its small rank square-root $S \sim N \times M$, $M \ll N$.

There are two known approaches for construction of a small rank square-root approximation: either a stochastic or a deterministic approach. The former is realized in the ensemble Kalman filter [3]. It is based on the fundamental theorem of statistical estimation theory. Let $\{\xi_i;\ i \geq 1\}$ be independent realizations of the same random variable ξ with mean $a = E\xi$ and covariance $P = E\lfloor(\xi - a)(\xi - a)^{\mathrm{T}}\rfloor$. Let

$$a_M = \frac{1}{M}\sum_{i=1}^{M}\xi_i,$$

$$S_M = \frac{1}{\sqrt{M-1}}[\,\xi_1 - a_M \quad \xi_2 - a_M \quad \cdots \quad \xi_M - a_M\,], \tag{3}$$

then with probability 1

$$\begin{aligned} P_M &= S_M \cdot (S_M)^{\mathrm{T}} \\ &= \frac{1}{M-1}\sum_{i=1}^{M}(\xi_i - a_M)\cdot(\xi_i - a_M)^{\mathrm{T}} \to P, \\ &\quad M \to +\infty. \end{aligned} \tag{4}$$

With the ensemble Kalman filter approach all the requirements outlined above are fulfilled:

1. The ensemble KF uses a square-root approximation of the covariance matrix, and hence it is robust in finite precision arithmetic.
2. The associated computational cost for propagation of the covariance matrix is in the order of 100–500 model runs [8,13] which can be easily parallelized. Thus, it has the potential to reduce drastically the computational costs as compared to the classical Kalman filter implementation.
3. It works with any system propagator, linear or non-linear. In this respect it is able to interface the model in a highly transparent way (the model is just a black box for the filter), and hence is feasible for most real-life applications.
4. The filter has no specific restrictions with respect to the definition of system noise. The noise can be introduced in any part of the model.

The only drawback of this approach is the relatively slow convergence of the covariance estimate:

$$P_M - P \sim \frac{1}{\sqrt{M}}, \quad M \to +\infty. \tag{5}$$

In the other approach, a deterministic low rank square-root approximation is constructed in the following way. If P is represented according to its SVD decomposition

$$\begin{aligned} &P = LDL^{\mathrm{T}}, \qquad D = \mathrm{diag}\{\lambda_i\}, \quad \lambda_i \geq \lambda_{i+1} \geq 0, \\ &LL^{\mathrm{T}} = L^{\mathrm{T}}L = I, \end{aligned} \tag{6}$$

then

$$P \approx S \cdot S^{\mathrm{T}},$$
$$S = [LD^{1/2}]_{1:M} = \left\lfloor \sqrt{\lambda_1} v_1 \quad \sqrt{\lambda_2} v_2 \quad \cdots \quad \sqrt{\lambda_M} v_M \right\rfloor. \tag{7}$$

In other words, the main attention is focused on the most essential part of the square-root of the covariance corresponding to the leading eigenvalues. This approach also generally fulfills the requirements outlined above. For many real-life applications the number M that is needed to generate an adequate approximation of the covariance is in the order 50–100 [8,16], which is usually much less than the system dimension.

Unfortunately, the deterministic uncertainty propagator can not be constructed in the same transparent way as the ensemble Kalman filter. One of the main problems is how to incorporate new noise into the total system uncertainty. A procedure that solves this problem was presented by Verlaan and Heemink in Ref. [15]. The corresponding filter is called the reduced rank square-root filter (RRSQRT KF).

According to the RRSQRT approach the error covariance matrix is expressed in a small number of modes, stored in a lower rank square-root matrix. The number of modes is a measure of the storage and computation time required by the filter, and should be kept as low as possible. The RRSQRT algorithm includes a reduction part that reduces the number of modes if it becomes too large in order to ensure that the filter problem is feasible. However, one cannot prevent the filter from loosing some covariance information during this reduction. Additionally, when considering variables of different magnitudes as part of the state vector some sort of normalization is needed before the truncation.

Here a new and more appropriate truncation procedure is presented that needs much less computational time than the original procedure presented in Ref. [14]. The introduced approach is fully applicable in the case of a single-variable dynamical system or when the system state vector does not include variables of different orders of magnitude. In the general case of a multivariable dynamical system some normalization may still be needed.

The paper is organized as follows. First an overview of the classical RRSQRT KF is given. Then a procedure known as the Lanczos algorithm [4] is presented, followed by a short discussion and a presentation of the new TRUE RRSQRT algorithm. Finally, some tests are shown that compare the TRUE RRSQRT KF and the classical RRSQRT KF for application in the 1D radionuclide transport model RIVTOX [17].

2. Classical RRSQRT KF

We consider the following stochastic dynamical system:

$$x_k = \Phi(x_{k-1}, u_k, \varepsilon_k), \tag{8}$$

where x_k is the state vector at time step k, u_k is the forcing of the system, ε_k is the model error with covariance matrix Q_k, and $\Phi()$ is the model operator. Note that a general non-linear model operator is assumed. The propagated covariance matrix P_k consists of two parts: (i) propagation of the covariance at the previous time step P_{k-1} according to the model dynamics, and (ii) injection of new noise represented by the noise covariance matrix Q_k.

The RRSQRT KF algorithm is based on a square-root factorization of the covariance matrix $P = SS^{\mathrm{T}}$ where S is a matrix that contains a small number (M) of leading eigenvectors s_i, $i = 1, \ldots, M$. The algorithm can be summarized as follows:

1. Forecast of the state vector
$$x_k^f = \Phi(x_{k-1}^a, u_k, 0), \tag{9}$$
where indices f and a denote, respectively, forecast and analysis (update).
2. Propagation of the leading eigenvectors:
$$s_{i,k}^f = \frac{1}{\omega}[\Phi(x_{k-1}^a + \omega s_{i,k-1}^a, u_k, 0) - \Phi(x_{k-1}^a, u_k, 0)], \quad i = 1, \ldots, M, \tag{10}$$
where ω is a scale factor. Heemink et al. [5] argue that $\omega = 1$ is a suitable choice. Smaller values of ω will increase the danger of filter divergence, whereas larger values will provide unrealistic inputs to the dynamical model.
3. Propagation of the new system noise
$$s_{q+i,k}^f = \frac{1}{\omega}[\Phi(x_{k-1}^a, u_k, \omega q_{i,k}) - \Phi(x_{k-1}^a, u_k, 0)], \quad i = 1, \ldots, l, \tag{11}$$

where $q_{i,k}$ is the ith column of the square-root of the covariance matrix of the system noise Q_k, and l is the number of modes that are used to approximate the noise covariance matrix.

4. Reduction of the covariance matrix: the propagation steps (2) and (3) increases the number of columns in the error covariance matrix from M to $M+\mathrm{l}$. To reduce the number of columns, and hence keep the rank of the matrix constant throughout the simulation, a lower rank approximation of S_k^f is applied by keeping only the M leading eigenvectors of the error covariance matrix. The reduction is achieved by an eigenvalue decomposition of the matrix $(S_k^f)^{\mathrm{T}} S_k^f$. If the state vector contains variables with different scales of magnitude, S_k^f should be normalized prior to the eigenvalue decomposition.
5. Measurement update: the state vector is updated using the Kalman filter update scheme. By using a sequential updating algorithm it is not necessary to calculate the full forecast error covariance matrix, and the updating can be performed using S_k^f directly. Corresponding formulae can be found in Ref. [2].

3. Reduction step: problem formulation and discussion

Mathematically, the reduction step can be formulated in the following way. Calculate in the best possible way matrix $\overline{S}$ such that

$$\overline{S} \cdot \overline{S}^{\mathrm{T}} \cong P_k^f = S_k^f \cdot (S_k^f)^{\mathrm{T}},$$
$$\overline{S} \sim N \times M, \qquad S_k^f \sim N \times (M+l), \tag{12}$$

where N is the dimension of the state vector ($N \gg 1$), M the rank of the approximate square-root matrix of the state covariance ($M \ll N$), and l the rank of the approximate square-root matrix of the system noise covariance, $l \ll N$.

An elegant and effective solution of the truncation problem was suggested by Verlaan in Ref. [14]. It is based on the well-known linear algebra fact [11] that two matrices (omitting indices) $S{\cdot}S^{\mathrm{T}}$ (large) and $S^{\mathrm{T}}{\cdot}S$ (small) have the same nonzero eigenvalues, or in other words the same leading eigenvalues. The truncation problem has then a straightforward solution:

$$(S_k^f)^{\mathrm{T}} \cdot S_k^f = L \cdot D \cdot L^{\mathrm{T}}, \qquad \overline{S} = [S_k^f \cdot L]_{i=1,M}. \tag{13}$$

The first operation is an SVD decomposition of a symmetric matrix with additional non-increasing ordering of eigenvalues in a diagonal matrix:

$$D = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_{M+l}\}, \quad \lambda_i \geq \lambda_{i+1}. \tag{14}$$

The second operation $[]_{i=1,M}$ takes only the first M columns of the corresponding matrix. This solution is operational and attractive for its computational efficiency. One needs only $O(N*(M+l)^2+(M+l)^3)$ operations, which is much less than $O(N^3)$ in the case of calculation of the full covariance matrix and its SVD decomposition (M, $l \ll N$).

The algorithm works fine in cases where the state vector components are the values of the same physical variable. The situation becomes much worse when one has to include into the state vector two or more different physical variables of different orders of magnitude. Then the truncation in Eq. (13) will mix all the variables together, and optimality of the truncation procedure cannot be guaranteed. A simple example explains this situation.

Example. In a 1D water flow simulation the modeled variables are water levels and discharges. They have strong interrelations so they have to be included in one state vector. The corresponding low rank approximation of the covariance matrix has a blocked structure $S = [\,H \quad G\,]^{\mathrm{T}}$ where H and G are low rank approximations of the covariance matrix of, respectively, water levels and discharges.

Usually in the case of large rivers the errors associated with water levels are of the order 10^0 (m), while those associated with discharges could be of the order 10^1–10^2 ($\mathrm{m}^3\,\mathrm{s}^{-1}$). The structure of the two matrices $P = SS^{\mathrm{T}}$ (large) and $\hat{P} = S^{\mathrm{T}}S$ (small) illustrates the problem

$$P = SS^{\mathrm{T}} = \begin{bmatrix} HH^{\mathrm{T}} & HG^{\mathrm{T}} \\ GH^{\mathrm{T}} & GG^{\mathrm{T}} \end{bmatrix},$$
$$\hat{P} = S^{\mathrm{T}}S = H^{\mathrm{T}}H + G^{\mathrm{T}}G.$$

In the case of P all the errors of the different variables are separated into different blocks, while for $\hat{P}$ the first

item $H^{\mathrm{T}}H$ can be even dropped because of the huge difference of orders when compared to $G^{\mathrm{T}}G$ (10^1 and 10^2–10^4, respectively). Thus, the problem of proper weighting of the components of S appears.

In Ref. [15] an extension of Eq. (13) was suggested for the case of a mixed state vector. The extended algorithm can be written in the following form:

```
Start with (random) vector b, compute v0 = b/||b||2
For j = 1, 2, ... until Convergence < ε
     vj = Avj-1;
     orthogonalize (if needed) vj to all {v0, ..., vj-1};
     form a new (tridiagonal) matrix T based on the information about ''present'' and ''past'';
     compute eigenvalues of T which are approximate eigenvalues of A;
     compute Convergence;
Compute approximate eigenpairs of A.
```

$$\hat{S} = W \cdot S_k^f, \qquad \hat{S}^{\mathrm{T}} \cdot \hat{S} = L \cdot D \cdot L^{\mathrm{T}},$$
$$S = [\hat{S} \cdot L]_{i=1,M}, \qquad \overline{S} = W^{-1} \cdot S. \tag{15}$$

Matrix W in Eq. (15) is a normalization matrix and provides normalization of all variables to be of approximately equal order of magnitude. The question of choice of matrix W is discussed in Refs. [1,14,15] and others. An interesting approach of choosing W independently of variable units is suggested in Segers et al. [9]. Nevertheless, it has to be mentioned that there is no universal and correct procedure of choosing the normalization matrix.

Before proceeding let us remind once again the formulation of the truncation problem given by Eq. (12). The problem can also be formulated as:

Calculate the small number M of leading
eigenpairs of the (large) matrix

$$P_k^f = S_k^f \cdot (S_k^f)^{\mathrm{T}}.$$

The following section contains a description of an algorithm that solves approximately the same problem.

4. Lanczos procedure

The Lanczos algorithm first appeared in the work by Lanczos [7]. The main problem to be solved is formulated as:

Find several extreme eigenpairs of a (large)
real symmetric sparse matrix A.

The algorithm itself can be characterized as an iterative procedure, and, omitting details, has the following pseudocode structure (for a detailed description of the algorithm see Ref. [4]):

It has to be noted that one needs only several matrix-vector multiplications and some vector operations such as scalar product during the whole algorithm. The most essential part of the procedure can be formulated as follows:

For the calculation of several extreme eigenpairs of
a real symmetric matrix A one needs to compute
several members of the sequence :

$$v_0 = \frac{b}{||b||_2}, \qquad v_j = A v_{j-1}, \quad j \geq 1 \tag{16}$$

with some additional vector–vector calculations.

In the case of a large sparse matrix A the operation Av can be performed in a fast way using information about the sparsity. It has to be noted, however, that sparsity is not an essential part of the algorithm but an additional useful feature. Additionally, the criterion of convergence has a specific precision parameter ε, so one can vary it to obtain optimized performance taking into account the trade-off between precision and computational time.

5. The true RRSQRT KF

At this point we know how the truncation problem is formulated and, in addition, we know an effective way of solving similar problems. Before proceeding to the formulation of the new algorithm some remarks should be noted.

Remark 1. In every time step in the RRSQRT KF algorithm we have to compute the fixed (small) number M of leading eigenvalues and corresponding eigenvectors of the symmetric covariance matrix $P^* = S^* \cdot S^{*\mathrm{T}}$, where the matrix S^* is formed in the propagation step of the RRSQRT KF filter.

Remark 2. The Lanczos algorithm provides the opportunity to compute a small number of leading eigenvalues and corresponding eigenvectors of a symmetric matrix A using several matrix-vector multiplications.

Remark 3. There is no restriction on the matrix to be sparse in order to use the Lanczos algorithm. Sparsity is used only for fast matrix-vector multiplications.

Remark 4. $P^* v = S^* \cdot S^{*\mathrm{T}} v = S^* \cdot (S^{*\mathrm{T}} v)$.

Thus we can formulate the new truncation step of the TRUE (TRUncation Enhanced) RRSQRT KF algorithm:

```
Start with (random) vector b, compute v_0 = b/||b||_2
For j = 1, 2, ... until Convergence < ε
     w_j = S^{*T} v_{j-1};
     v_j = S^* w_j;
     orthogonalize (if needed) v_j to all {v_0, ..., v_{j-1}};
     form a new (tridiagonal) matrix T based on the information about ''present'' and ''past'';
     compute eigenvalues of T which are approximate eigenvalues of P^* = S^* · S^{*T};
     compute Convergence;
Compute approximate eigenpairs of P^*.
```

All other steps are similar to those of the classical RRSQRT KF.

The important features of the new truncation step are:

- The operations $S^{*\mathrm{T}} v$ and $S^* w$ need a rather small amount of computational time because the matrix S^* has a small number of elements. The number of iterations needed for the convergence is relatively small. Overall, the algorithm is very efficient.
- The algorithm has a specific coefficient $($ that regulates the precision of the algorithm. The classical RRSQRT KF is not tunable at all.

It should be noted that in the case of a multi-variable system some normalization might still be needed.

6. Performance tests

Some tests for evaluating the performance of the new filter were made using the RIVTOX model. The main model equation is of an advection-dispersion-reaction type. For more details see Refs. [10,17,18]. The purpose of the tests is to compare the precision of the approximation of the "true" covariance matrix of the state vector. The "true" covariance matrix was derived from the propagation of 1000 ensemble members using the ensemble KF methodology [3].

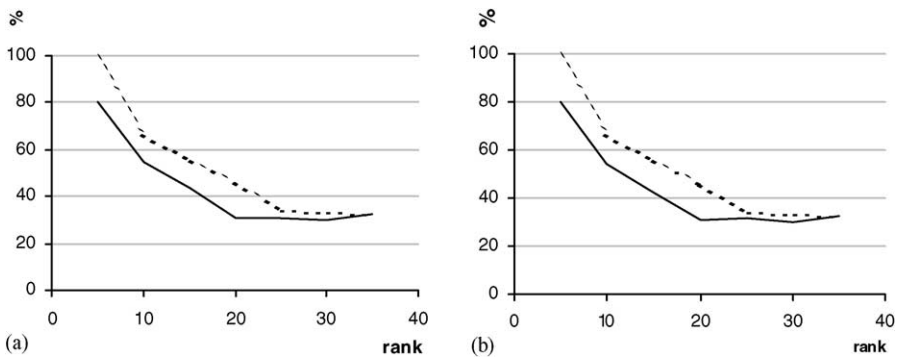

Fig. 1. Relative trace *trace* (a) and relative second norm *norm2* (b) of the error of the covariance matrix approximation of the TRUE RRSQRT filter (solid line) and the classical RRSQRT filter (dotted line).

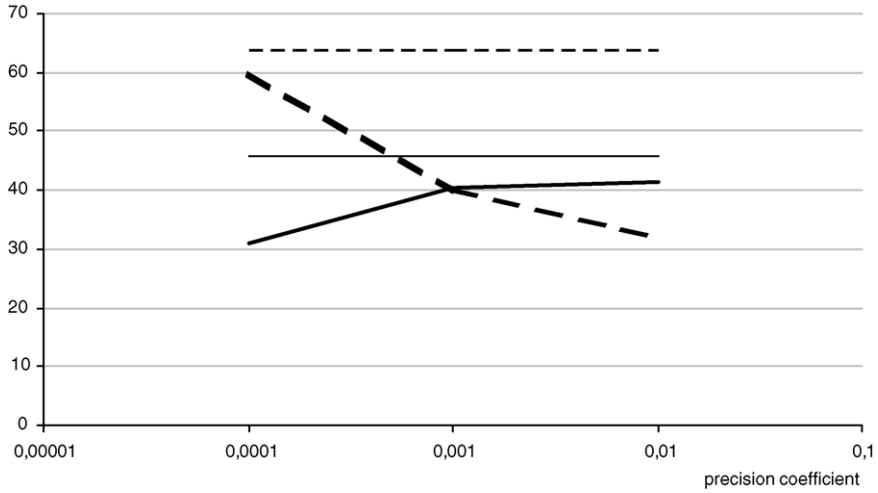

Fig. 2. Trade-off between the precision (%) (in terms of the relative second norm *norm2*) and the computational time (s). The precision of the TRUE RRSQRT filter (solid thick line) and corresponding computational time (dotted thick line) compared to the precision of the classical RRSQRT filter (solid thin line) and corresponding computational time (dotted thick line).

Two measures of precision are defined:

$$\text{trace} = \frac{\text{Tr}(P_{\text{true}} - P_{\text{appr}})}{\text{Tr}(P_{\text{true}})} \times 100\%,$$

$$\text{norm2} = \frac{\|P_{\text{true}} - P_{\text{appr}}\|_2}{\|P_{\text{true}}\|_2} \times 100\%,$$

where $\text{Tr}(P)$ is the trace of matrix P (the sum of all diagonal elements), P_{true} is the referred "true" system covariance matrix, computed from the statistics of 1000 ensemble members in the corresponding ensemble KF run, P_{appr} is the approximation of the covariance matrix, $P_{\text{appr}} S \cdot S^{\text{T}}$, S is the approximation of the square-root of the system covariance matrix, $||P||_2$ is the second norm of matrix P (the square-root of the sum of squares of all matrix elements).

For both measures a perfect approximation corresponds to trace = norm2 = 0.

In the tests the main variable parameter was the number of modes M for approximation of the square-root

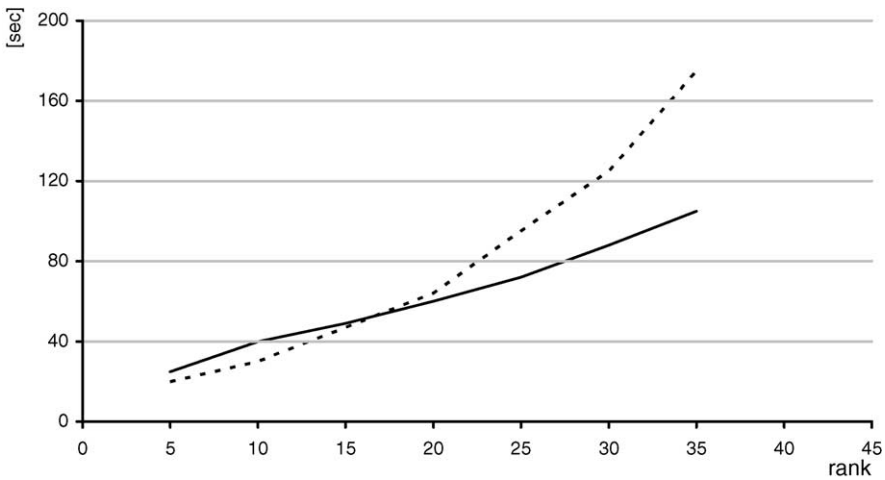

Fig. 3. Computational time (s) of the TRUE RRSQRT filter (solid line) and the classical RRSQRT filter (dotted line).

matrix, and it was set to values of 5, 10, 15, 20, 25, 30 and 35. The number of modes l for approximation of the system noise covariance matrix was fixed and set equal to 10.

The results of the tests are presented in Figs. 1–3. From the figures the following can be seen:

- For the same rank the TRUE RRSQRT provides a better approximation of the covariance matrix than the classical RRSQRT. The gain in using the TRUE RRSQRT is more pronounced for smaller ranks.
- The TRUE RRSQRT is computationally more efficient than the classical RRSQRT. The classical RRSQRT requires more than twice the computational time of the TRUE RRSQRT to obtain the same level of precision of the covariance approximation. This becomes more pronounced for larger ranks since the computational time required by the classical RRSQRT has a cubic dependence on the rank and the TRUE RRSQRT has an almost linear dependence.

7. Conclusions

A new data assimilation algorithm has been developed, the TRUE RRSQRT (TRUncation Enhanced Reduced Rank SQuare-RooT) Kalman filter. Some tests were carried out using the mathematical model of radionuclide pollution transport with the 1D river model RIVTOX to evaluate the performance of the TRUE RRSQRT filter and compare with the classical implementation of the RRSQRT KF. The following conclusions were obtained:

- The TRUE RRSQRT is computationally more efficient than the classical RRSQRT. Compared to the classical RRSQRT approach the TRUE RRSQRT filter gives a reduction of a factor 2 or more in computation time for the same level of precision of the approximation of the covariance matrix.
- The TRUE RRSQRT provides a significant improvement of the estimation of the covariance matrix compared to the classical RRSQRT, especially for lower rank approximations.
- The TRUE RRSQRT includes a precision coefficient that can be tuned for specific applications depending on the trade-off between precision and computational load. The classical RRSQRT filter cannot be tuned.

It should be noted that the TRUE RRSQRT filter is fully applicable for non-linear systems because it uses a finite difference approximation of the Jacobian. The performance of the TRUE RRSQRT has been demonstrated for a single-variable system. In the case of a multi-variable system the performance of the covariance approximation depends on the choice of normalization. Research is presently being conducted for further development of the TRUE RRSQRT suitable for multi-variable systems.

Acknowledgements

This work was supported by the European Commission under the contract no. FIKR-CT-2001-50021 “Data Assimilation for Off-site Nuclear Emergency Management (DAONEM-CTI)”.

References

[1] R. Cañizares, On the application of data assimilation in regional coastal models, Ph.D. Thesis, IHE Delft and TU Delft, The Netherlands, 1999.
[2] C.K. Chui, G. Chen, Kalman Filtering: With Real-Time Applications, 2nd ed., Springer-Verlag, New York, 1991.
[3] G. Evensen, Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. 99 (C5) (1994) 10143–10162.
[4] G. Golub, C. Van Loan, Matrix Computations, John Hopkins University Press, Baltimore, MD, 1989.
[5] A.W. Heemink, M. Verlaan, A.J. Segers, Variance reduced ensemble Kalman filtering, Report 00-03, Department of Applied Mathematical Analysis, Delft University of Technology, The Netherlands, 2000.
[6] R.E. Kalman, A new approach to linear filter and prediction theory, J. Basic Eng. 82D (1960) 35–45.
[7] C. Lanczos, An iterative method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bureau Standards B 45 (1950) 225–280.
[8] H. Madsen, R. Cañizares, Comparison of extended and ensemble Kalman filters for data assimilation in coastal area modelling, Int. J. Numer. Methods Fluids 31 (1999) 961–981.
[9] A.J. Segers, A.W. Heemink, M. Verlaan, M. van Loon, A modified RRSQRT-filter for assimilating data in atmospheric chemistry models, Environ. Model. Softw. 15 (2000) 663–671.
[10] O. Slavik, M. Zheleznyak, N. Dzuba, A. Marinets, G. Lyashenko, L. Papush, T. Shepeleva, B. Mihaly, Implementation of the decision support system for the river-reservoir network

affected by releases from the Bohunice NPP Slovakia, Radiat. Prot. Dosim. 73 (1–4) (1997) 171–175.
[11] G. Strang, Linear Algebra and Its Applications, 3rd ed., Saunders, Philadelphia, PA, 1988.
[12] R. Todling, S.E. Cohn, Suboptimal schemes for atmospheric data assimilation based on the Kalman filter, Mon. Weather Rev. 122 (1994) 2530–2557.
[13] J. Van Leeuwen, Comment on data assimilation using an ensemble Kalman filter technique, Mon. Weather Rev. 127 (1999) 1374–1377.
[14] M. Verlaan, Efficient Kalman Filtering Algorithms for Hydrodynamic Models, TU Delft, The Netherlands, 1998.
[15] M. Verlaan, A.W. Heemink, Reduced rank square root filters for large scale data assimilation problems, in: Proceedings of the Second International Symposium on Assimilation of Observations in Meteorology and Oceanography, World Meteorological Organization, 1995, pp. 247–252.
[16] M. Verlaan, A.W. Heemink, Tidal flow forecasting using reduced rank square root filters, Stoch. Hydrol. Hydraul. 11 (1997) 349–368.
[17] M. Zheleznyak, G. Donchytz, V. Hygynyak, A. Marinetz, G. Lyashenko, P. Tkalich, 1997. RIVTOX—one dimensional model for the simulation of the transport of radionuclides in a network of river channels, RODOS Report WG4-TN(97)05, Forschungszentrum Karlsruhe, 2000, 48 pp.
[18] M.J. Zheleznyak, P.V. Tkalich, G.B. Lyashenko, A.V. Marinets, Radionuclide aquatic dispersion model—first approach to integration into the EC decision support system on a basis of Post-Chernobyl experience, Radiat. Prot. Dosim. N6 (1993) 37–43.

Dimitri Treebushny received his MSc in calculus in 1999 from Kiev State University, Ukraine. In 2000 he started doing PhD at the Institute of Mathematical Machines and System Problems, National Academy of Science of Ukraine. Presently, he has a position as junior scientific researcher in the Department of Environmental Modeling at the institute. His research interests are applications of suboptimal data assimilation procedures to large-scale problems, particularly black-box DA algorithms and development of numerical solvers for advection–diffusion processes, as well as conservation law equations.

Henrik Madsen received his MSc in civil engineering in 1993 from the Technical University of Denmark (DTU). In 1996, he obtained his PhD from DTU on work on stochastic hydrology. Since 1996 he has been employed as a research scientist at DHI Water and Environment, Denmark. His main research interests are data assimilation, optimization, parameter estimation, and uncertainty assessment with special emphasis on application in high-dimensional, numerical hydrodynamic and hydrological modeling systems.