ISSN 0005-1179, Automation and Remote Control, 2009, Vol. 70, No. 5, pp. 855–871. © Pleiades Publishing, Ltd., 2009.
Original Russian Text © M.V. Kulikova, 2009, published in Avtomatika i Telemekhanika, 2009, No. 5, pp. 122–139.

On Scalarized Calculation of the Likelihood Function in Array Square-root Filtering Algorithms[1]

M. V. Kulikova

School of Computational and Applied Mathematics,
University of the Witwatersrand, Johannesburg, Republic of South Africa

Received November 6, 2007

Abstract—An efficient method of scalarized calculation of the logarithmic likelihood function based on the array square-root implementation methods for Kalman filtering formulas was proposed. The algorithms of this kind were shown to be more stable to the roundoff errors than the conventional Kalman filter. The measurement scalarization technique enables a substantial reduction in the computational complexity of the algorithm. Additionally, the new implementations are classified with the array filtering algorithms and thereby are oriented to the parallel calculations. Computational results corroborated effectiveness of the new algorithm.

PACS numbers: 02.70.Rr, 02.50.Ey, 02.60.Dc
DOI: 10.1134/S0005117909050129

1. INTRODUCTION

Solution of many everyday problems such as recognition, detection, estimation, signal filtering, and so on relies on the statistical methods [1]. For example, the classical approach to fault detection in the stochastic systems is based on the hypotheses testing procedure. The decision strategy is built upon calculating each time the likelihood ratio and its subsequent comparison with the threshold values [2, 3].

For the linear discrete-time stochastic systems corrupted by white Gaussian noises, calculation of the likelihood function generating the ratio is based on using the discrete-time Kalman filter [4,5]. Its conventional implementation is known to be unstable to the roundoff errors, which also leads to an incorrect calculation of the likelihood function. In the practice of solving the fault detection problem, this may mean that if the real cause of divergence of the optimal filter lies in the roundoff errors, then an erroneous hypothesis of going to a regime will be accepted. Therefore, we face the need to develop new efficient and reliable algorithms to calculate the logarithmic likelihood function based on the numerically stable filtering algorithms.

The present paper discusses the square-root filtering algorithms which can be regarded as currently most promising implementations of the discrete-time Kalman filter (see, for example, the discussions in [6–8] and in many other publications). Yet, despite the numerical stability of the square-root algorithms, their computational complexity is much higher than that of the conventional Kalman filter [9], which restricts the possibility of their practical use. The situation is aggravated by the fact that the calculations must be carried out in real time.

[1] This work was supported by the School of Computational and Applied Mathematics, University of the Witwatersrand, Republic of South Africa.

There exist two main time optimization techniques for the filtering algorithms: the so-called observation scalarization[2] dating back to the 1970s and the design of the algorithms enabling efficient parallelization which is characteristic of the state-of-the-art of the Kalman filtering theory [10, 11, and others]. An entire class of the so-called array algorithms that are oriented to the multiprocessor computer systems has been developed recently. They are based on generating a block matrix containing all information required at the given stage of filtering and applying the orthogonal transformation to obtain the desired (upper or lower) triangular form (see, for example, [7]). Therefore, the use of the numerically stable orthogonal transformations at each step of recursion is another significant advantage of the array filtering algorithms.

An algorithm to calculate the likelihood function on the basis of the conventional discrete-time Kalman filter using the technique of measurement scalarization was suggested and substantiated by Semushin and Tsyganova [12], as well as Kulikova [13], who proved that the "scalar" evaluation of the likelihood function is more effective in terms of computations than the "vector" one, that is, the conventional filter implementation.

The main task of the present paper is to develop the ideas of the scalar updating of data and array square-root algorithms with the aim of developing new efficient methods for the likelihood function evaluation. This will allow one to use the advantages of each algorithm such as stability to the roundoff errors featured by all square-root algorithms, efficient parallelization characteristic of all array filtering algorithms, and finally, reduction of the calculation time of each processor of a multiprocessor systems characteristic of measurement "scalarization" results.

2. CONVENTIONAL AND ARRAY SQUARE-ROOT IMPLEMENTATION METHODS OF THE DISCRETE KALMAN FILTER

Let us consider the linear discrete-time stochastic system

$$x_k = F_k x_{k-1} + G_k w_k, \tag{1}$$

$$z_k = H_k x_k + v_k, \tag{2}$$

where the values $F_k \in \mathbb{R}^{n\times n}$, $G_k \in \mathbb{R}^{n\times q}$, $H_k \in \mathbb{R}^{m\times n}$, $Q_k \in \mathbb{R}^{q\times q}$ and $R_k \in \mathbb{R}^{m\times m}$ are certain parameter matrices of the system, x_k is the n-dimensional state vector, z_k is the observable m-dimensional measurement vector, $k = 0,\dots,N$ is the discrete system time, that is, x_k are $x(t_k)$. The sequences $\{w_0, w_1, \dots\}$ and $\{v_1, v_2, \dots\}$ are independent normally distributed white sequences with zero mean and covariance matrices $Q_k > 0$ and $R_k > 0$, respectively. Additionally, $w_k \sim \mathcal{N}(0, Q_k)$ and $v_k \sim \mathcal{N}(0, R_k)$ are independent of $x_0 \sim \mathcal{N}(\bar{x}_0, \Pi_0)$. Under the given parameter matrices F_k, G_k, H_k, Q_k, and R_k and the initial conditions $\hat{x}_{0|0} = \bar{x}_0$, $P_{0|0} = \Pi_0$, the optimal mean square estimate of unobservable state vector x_k is obtained using the discrete-time Kalman filter whose conventional implementation obeys the following recurrent equation.

Algorithm 1 (conventional Kalman filter).

(1) *Time updates*:

$$\text{estimate:}\quad \hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}, \tag{3}$$

$$\text{error covariance matrix:}\quad P_{k|k-1} = F_k P_{k-1|k-1} F_k^{\mathrm{T}} + G_k Q_k G_k^{\mathrm{T}}. \tag{4}$$

[2] It is worth noting here that the discrete-time Kalman filter with scalar updates allows one to avoid inversion of the innovation covariance matrix $R_{e,k}$ and, therefore, is more stable to the roundoff errors. The array filtering algorithms considered in the present paper have this property initially.

(2) *Measurement updates*:

$$\text{Kalman Gain:} \quad K_{p,k} = P_{k|k-1} H_k^{\mathrm{T}} (H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k)^{-1}, \tag{5}$$

$$\text{estimate:} \quad \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - H_k \hat{x}_{k|k-1}), \tag{6}$$

$$\text{error covariance matrix:} \quad P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}, \tag{7}$$

where $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ are, respectively, the predicted and filtered estimates of the state vector x_k of system (1), (2), $P_{k|k-1}$ and $P_{k|k}$ are, respectively, the predicted and filtered error covariance matrices, that is, $P_{k|k-1} = \mathbb{E}\left[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^{\mathrm{T}}\right]$ and $P_{k|k} = \mathbb{E}\left[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^{\mathrm{T}}\right]$.

It is well known that the conventional implementation (3)–(7) is unstable to roundoff errors. One of the reasons is the loss of positive definiteness of the matrices $P_{k|k-1}$, $P_{k|k}$. Square-root implementations of the discrete Kalman filter were developed to avoid such situations. At the current stage of research, preference is given to the so-called array square-root algorithms whose advantage lies in using the numerically stable orthogonal transformations at each recursion step.

The following notation is introduced to make the presentation below more convenient. We consider a Cholesky decomposition of the form $A = A^{\mathrm{T}/2} A^{1/2}$, where $A^{1/2}$ is an upper triangular matrix. Then, $A^{\mathrm{T}/2} = (A^{1/2})^{\mathrm{T}}$, $A^{-1/2} = (A^{1/2})^{-1}$, $A^{-\mathrm{T}/2} = (A^{-1/2})^{\mathrm{T}}$ and $\left(a^{(i)}\right)^{\mathrm{T}}$ is the ith row of the matrix A. In what follows, we consider two implementations from the class of array square-root algorithms proposed first in [7].

Algorithm 2 (extended square-root covariance filter).

(1) *Time updates*: Predict $P_{k|k-1}^{1/2}$ and $P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1}$:

$$O_{k,1} \left[\begin{array}{c|c} P_{k-1|k-1}^{1/2} F_k^{\mathrm{T}} & P_{k-1|k-1}^{-\mathrm{T}/2} \hat{x}_{k-1|k-1} \\ Q_k^{1/2} G_k^{\mathrm{T}} & 0 \end{array} \right] = \left[\begin{array}{c|c} P_{k|k-1}^{1/2} & P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1} \\ 0 & -\widetilde{Q}_k^{-1/2} K_{b,k} \hat{x}_{k|k-1} \end{array} \right], \tag{8}$$

where $O_{k,1}$ is any orthogonal rotation that upper-triangularizes the first (block) column of the matrix on the left-hand side of (8). Additionally, $K_{b,k} = Q_k G_k P_{k|k-1}^{-1}$ and $\widetilde{Q}_k = Q_k - Q_k G_k^{\mathrm{T}} P_{k|k-1}^{-1} G_k Q_k$.

(2) *Measurement updates.* Obtain filtered estimates $P_{k|k}^{1/2}$ and $P_{k|k}^{-\mathrm{T}/2} \hat{x}_{k|k}$:

$$O_{k,2} \left[\begin{array}{cc|c} R_k^{1/2} & 0 & -R_k^{-\mathrm{T}/2} z_k \\ P_{k|k-1}^{1/2} H_k^{\mathrm{T}} & P_{k|k-1}^{1/2} & P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1} \end{array} \right] = \left[\begin{array}{cc|c} R_{e,k}^{1/2} & \overline{K}_{p,k}^{\mathrm{T}} & -\bar{e}_k \\ 0 & P_{k|k}^{1/2} & P_{k|k}^{-\mathrm{T}/2} \hat{x}_{k|k} \end{array} \right], \tag{9}$$

where $O_{k,2}$ is any orthogonal rotation that upper-triangularizes the first two (block) columns of the matrix on the left-hand side of (9). Additionally, $\overline{K}_{p,k} = P_{k|k-1} H_k^{\mathrm{T}} R_{e,k}^{-1/2}$ is the normalized Kalman gain and $\bar{e}_k = R_{e,k}^{-\mathrm{T}/2} e_k$ is the normalized innovation with the covariance matrix $R_{e,k}$.

Algorithm 3 (combined square-root filter).

(1) *Time updates.* Predict $P_{k|k-1}^{1/2}$, $P_{k|k-1}^{-\mathrm{T}/2}$ and $P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1}$:

$$O_{k,1} \left[\begin{array}{cc|c|c} P_{k-1|k-1}^{-\mathrm{T}/2} F_k^{-1} & -P_{k-1|k-1}^{-\mathrm{T}/2} F_k^{-1} G_k & P_{k-1|k-1}^{1/2} F_k^{\mathrm{T}} & P_{k-1|k-1}^{-\mathrm{T}/2} \hat{x}_{k-1|k-1} \\ 0 & Q_k^{-\mathrm{T}/2} & Q_k^{1/2} G_k^{\mathrm{T}} & 0 \end{array} \right]$$

$$= \left[\begin{array}{cc|c|c} P_{k|k-1}^{-\mathrm{T}/2} & 0 & P_{k|k-1}^{1/2} & P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1} \\ -\widetilde{Q}_k^{-\mathrm{T}/2} K_{b,k} & \widetilde{Q}_k^{-\mathrm{T}/2} & 0 & -\widetilde{Q}_k^{-\mathrm{T}/2} K_{b,k} \hat{x}_{k|k-1} \end{array} \right], \tag{10}$$

where $O_{k,1}$ is any orthogonal rotation that either lower-triangularizes the first two (block) columns or upper-triangularizes the third (block) column of the matrix on the left-hand side of (10).

(2) *Measurement updates.* Obtain $P_{k|k}^{1/2}$, $P_{k|k}^{-\mathrm{T}/2}$ and $P_{k|k}^{-\mathrm{T}/2}\hat{x}_{k|k}$:

$$O_{k,2}\left[\begin{array}{cc|c|c} R_k^{1/2} & 0 & -R_k^{-\mathrm{T}/2}H_k & -R_k^{-\mathrm{T}/2}z_k \\ P_{k|k-1}^{1/2}H_k^{\mathrm{T}} & P_{k|k-1}^{1/2} & P_{k|k-1}^{-\mathrm{T}/2} & P_{k|k-1}^{-\mathrm{T}/2}\hat{x}_{k|k-1} \end{array}\right] = \left[\begin{array}{cc|c|c} R_{e,k}^{1/2} & \overline{K}_{p,k}^{\mathrm{T}} & 0 & -\bar{e}_k \\ 0 & P_{k|k}^{1/2} & P_{k|k}^{-\mathrm{T}/2} & P_{k|k}^{-\mathrm{T}/2}\hat{x}_{k|k} \end{array}\right], \qquad (11)$$

where $O_{k,2}$ is any orthogonal rotation that either upper-triangularizes the first two (block) columns or lower-triangularizes the third (block) column of the matrix on the left-hand side of (11).

Remark 1. Algorithm 3 is classified with the so-called combined filters because if enables one to calculate simultaneously the covariance, $P_{k|k}^{1/2}$, and information, $P_{k|k}^{-1/2}$, matrices (correspondingly, $P_{k|k-1}^{1/2}$ and $P_{k|k-1}^{-1/2}$).

Remark 2. It also deserved noting that the above combined Algorithm 3 offers an alternative in choosing the orthogonal transformation in (10), (11). At that, the result of filtering will be the same, but the computational characteristics and numerical behavior of the algorithms depend strongly on the choice of one or another variant of the orthogonal transformations. Strict theoretical analysis of the effect of the roundoff errors on some implementations of the discrete Kalman filter can be found in [8].

3. MEASUREMENT "SCALARIZATION" RESULTS IN THE ARRAY SQUARE-ROOT FILTERING ALGORITHMS

The procedure of scalar updating lies in successive (*componentwise* updating of the newly arrived measurement vector z_k, thereby enabling one to improve both the computational characteristic and reliability of filtering:

(1) The scalarization technique allows one to avoid the $m \times m$ matrix inversion of the $R_{e,k}$. As the result, the scalarized algorithms are more stable to the roundoff errors and more efficient in terms of the computer runtime.

(2) The "vector" algorithms, that is, those based on updating the *entire* observation vector, require that the entire observation matrix H_k be stored and used. In practice, the amount of the experimental data m in the vector z_k may be excessive, whereas the number of the estimated parameters n in the vector x_k, relatively low. Under these conditions, updating of the entire data array $[H_k \mid z_k]$ is, obviously, inadvisable, whereas application of the scalarized algorithms is expedient.

Remark 3. The case of $n \ll m$ is characteristic of the so-called repeating surveys [14] where during each accounting period the same statistical information is acquired and processed as, for example, in the medico-biological studies, socio-economic studies of the levels of unemployment or inflation, and so on.

We precede construction of the "scalar" implementation of the square-root Algorithms 2 and 3 by the well-known result.

Lemma 1 (see [9, p. 125]). *Let the covariance matrices of the measurer noise (2) be diagonal, that is,* $R_k = \operatorname{diag}\left\{\left(r_k^{(1)}\right)^2, \ldots, \left(r_k^{(m)}\right)^2\right\}$. *Then, the stage of measurement updating of the discrete Kalman filter (5)–(7) is equivalent to the following successive procedure where the matrix* H_k *is processed row by row:*

- *Set* $P_k^{(0)} = P_{k|k-1}$ *and* $\hat{x}_k^{(0)} = \hat{x}_{k|k-1}$.
- *For* $i = 1, \ldots, m$ *calculate*

$$\alpha_k^{(i)} = \left(h_k^{(i)}\right)^{\mathrm{T}} P_k^{(i-1)} h_k^{(i)} + \left(r_k^{(i)}\right)^2, \tag{12}$$

$$K_{p,k}^{(i)} = P_k^{(i-1)} h_k^{(i)} / \alpha_k^{(i)}, \tag{13}$$

$$P_k^{(i)} = P_k^{(i-1)} - K_{p,k}^{(i)} \left(h_k^{(i)}\right)^{\mathrm{T}} P_k^{(i-1)}, \tag{14}$$

$$\hat{x}_k^{(i)} = \hat{x}_k^{(i-1)} + K_{p,k}^{(i)} \left[z_k^{(i)} - \left(h_k^{(i)}\right)^{\mathrm{T}} \hat{x}_k^{(i-1)}\right]. \tag{15}$$

- *Assign* $P_{k|k} = P_k^{(m)}$ *and* $\hat{x}_{k|k} = \hat{x}_k^{(m)}$.

Remark 4. The requirement of diagonality of the matrices R_k is an unessential constraint. If it is not satisfied, then the so-called pseudo-observations $\bar{z}_k = L_k^{-1} z_k$ are used, where L_k are obtained using the Cholesky decomposition of the matrices $R_k = L_k L_k^{\mathrm{T}}$. For pseudo-observations, the measurer Eq. (2) is replaced by its algebraic equivalent: $\bar{z}_k = \overline{H}_k x_k + \bar{v}_k$, where $\overline{H}_k = L_k^{-1} H_k$, $\bar{v}_k = L_k^{-1} v_k$, and then indeed $\overline{R}_k = \mathbb{E}\left[\bar{v}_k \bar{v}_k^{\mathrm{T}}\right] = L_k^{-1} \mathbb{E}\left[v_k v_k^{\mathrm{T}}\right] L_k^{-\mathrm{T}} = L_k^{-1} L_k L_k^{\mathrm{T}} L_k^{-\mathrm{T}} = I$, where I is the identity matrix.

We apply to Algorithms 2 and 3 the technique of scalar updates.

Algorithm 4 (covariance square-root filter with scalar updates).

(1) *Time updates.* Use (8) to predict $P_{k|k-1}^{1/2}$ and $P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1}$.

(2) *Measurement updates.* Find the filtered estimates $P_{k|k}^{1/2}$ and $P_{k|k}^{-\mathrm{T}/2} \hat{x}_{k|k}$:

- Set $\left(P_k^{(0)}\right)^{1/2} = P_{k|k-1}^{1/2}$ and $\left(P_k^{(0)}\right)^{-\mathrm{T}/2} \hat{x}_k^{(0)} = P_{k|k-1}^{-\mathrm{T}/2} \hat{x}_{k|k-1}$.
- For $i = 1, \ldots, m$, calculate

$$O_{k,2}^{(i)} \left[\begin{array}{cc|c} r_k^{(i)} & 0 & -z_k^{(i)}/r_k^{(i)} \\ \left(P_k^{(i-1)}\right)^{1/2} h_k^{(i)} & \left(P_k^{(i-1)}\right)^{1/2} & \left(P_k^{(i-1)}\right)^{-\mathrm{T}/2} \hat{x}_k^{(i-1)} \end{array}\right]$$
$$= \left[\begin{array}{cc|c} \sqrt{\alpha_k^{(i)}} & \left(\overline{K}_{p,k}^{(i)}\right)^{\mathrm{T}} & -\bar{e}_k^{(i)} \\ 0 & \left(P_k^{(i)}\right)^{1/2} & \left(P_k^{(i)}\right)^{-\mathrm{T}/2} \hat{x}_k^{(i)} \end{array}\right], \tag{16}$$

where $O_{k,2}^{(i)}$ is any orthogonal rotation that upper-triangularizes the first two (block) columns of the matrix on the left-hand side of (16). Additionally, $\overline{K}_{p,k}^{(i)} = K_{p,k}^{(i)} \sqrt{\alpha_k^{(i)}}$ and $\bar{e}_k^{(i)} = e_k^{(i)} / \sqrt{\alpha_k^{(i)}}$.

- Assign $P_{k|k}^{1/2} = \left(P_k^{(m)}\right)^{1/2}$ and $P_{k|k}^{-\mathrm{T}/2} \hat{x}_{k|k} = \left(P_k^{(m)}\right)^{-\mathrm{T}/2} \hat{x}_k^{(m)}$.

Algorithm 5 (combined square-root filter with scalar updates).

(1) *Time updates.* Use (10) to predict $P^{1/2}_{k|k-1}$, $P^{-\mathrm{T}/2}_{k|k-1}$ and $P^{-\mathrm{T}/2}_{k|k-1}\hat{x}_{k|k-1}$.

(2) *Measurement updates.* Determine $P^{1/2}_{k|k}$, $P^{-\mathrm{T}/2}_{k|k}$ and $P^{-\mathrm{T}/2}_{k|k}\hat{x}_{k|k}$:

- Assign $\left(P^{(0)}_k\right)^{1/2} = P^{1/2}_{k|k-1}$, $\left(P^{(0)}_k\right)^{-\mathrm{T}/2} = P^{-\mathrm{T}/2}_{k|k-1}$, $\left(P^{(0)}_k\right)^{-\mathrm{T}/2}\hat{x}^{(0)}_k = P^{-\mathrm{T}/2}_{k|k-1}\hat{x}_{k|k-1}$.
- For $i = 1, \ldots, m$, calculate

$$O^{(i)}_{k,2}\left[\begin{array}{cc|c|c} r^{(i)}_k & 0 & -\left(h^{(i)}_k\right)^{\mathrm{T}}/r^{(i)}_k & -z^{(i)}_k/r^{(i)}_k \\ \left(P^{(i-1)}_k\right)^{1/2} h^{(i)}_k & \left(P^{(i-1)}_k\right)^{1/2} & \left(P^{(i-1)}_k\right)^{-\mathrm{T}/2} & \left(P^{(i-1)}_k\right)^{-\mathrm{T}/2}\hat{x}^{(i-1)}_k \end{array}\right]$$

$$= \left[\begin{array}{cc|c|c} \sqrt{\alpha^{(i)}_k} & \left(\overline{K}^{(i)}_{p,k}\right)^{\mathrm{T}} & 0 & -\bar{e}^{(i)}_k \\ 0 & \left(P^{(i)}_k\right)^{1/2} & \left(P^{(i)}_k\right)^{-\mathrm{T}/2} & \left(P^{(i)}_k\right)^{-\mathrm{T}/2}\hat{x}^{(i)}_k \end{array}\right], \qquad (17)$$

where $O^{(i)}_{k,2}$ is any orthogonal rotation that either upper-triangularizes the first two (block) columns or lower-triangularizes the third (block) column of the matrix on the left-hand side of (17).

- Assign $P^{1/2}_{k|k} = \left(P^{(m)}_k\right)^{1/2}$, $P^{-\mathrm{T}/2}_{k|k} = \left(P^{(m)}_k\right)^{-\mathrm{T}/2}$, $P^{-\mathrm{T}/2}_{k|k}\hat{x}_{k|k} = \left(P^{(m)}_k\right)^{-\mathrm{T}/2}\hat{x}^{(m)}_k$.

The following theorem proves equivalence of the developed filtering algorithms with scalar updates 4 and 5 and the conventional implementation of the Kalman filter.

Theorem 1 (theorem of equivalence). *Let the matrices R_k of noise covariance in measurer* (2) *be diagonal ones. Then in the scalarized Algorithms 4 and 5* (*formulas* (16) *and* (17), *respectively*) *the stages of filtering are algebraically equivalent to the stage of measurement updates of the discrete Kalman filter* (5)–(7) *where the observation matrix H_k is processed row by row.*

The proof is given in the Appendix.

4. CALCULATION OF THE LIKELIHOOD FUNCTION IN THE SCALARIZED ARRAY FILTERING ALGORITHMS

Let us consider the problem of parametric identification. Let the linear discrete stochastic system (1), (2) depend on some parameter $\theta \in \mathbb{R}^p$, that is, $F_k(\theta)$, $G_k(\theta)$, $H_k(\theta)$, $Q_k(\theta)$ and $R_k(\theta)$. It is required to estimate $\hat{\theta}$ from the available observations $Z_1^N = [z_1, \ldots, z_N]^{\mathrm{T}}$. The problem can be solved by the method of maximum likelihood. (For convenience, the argument θ is omitted in the matrices F_k, G_k, H_k, Q_k, and R_k, but we keep in mind this dependence.)

In the conditions of system (1), (2), the logarithmic likelihood function constructed for the observations Z_1^N obeys a formula from [5]:

$$L_\theta\left(Z_1^N\right) = \frac{1}{2}\sum_{k=1}^{N}\left\{-\frac{m}{2}\ln(2\pi) - \ln\left[\det(R_{e,k})\right] - e_k^{\mathrm{T}} R^{-1}_{e,k} e_k\right\}. \qquad (18)$$

As was already noted, the "scalar" updating allows one to improve the computational characteristics of the filtering algorithms (see also the results of computational experiments in Section 5). In this connection, we demonstrate how the likelihood function (18) can be calculated using the above scalarized Algorithms 4 and 5. Now, we formulate the main result.

Theorem 2. *Let the elements of the matrices $F_k \in \mathbb{R}^{n\times n}$, $G_k \in \mathbb{R}^{n\times q}$, $H_k \in \mathbb{R}^{m\times n}$, $Q_k \in \mathbb{R}^{q\times q}$, and $R_k \in \mathbb{R}^{m\times m}$ characterizing system* (1), (2) *be defined for all $\theta \in \mathbb{R}^p$, and the matrices R_k of noise covariance in measurer* (2) *be diagonal. We represent the matrix H_k row by row and the measurement vector z_k element by element. Then, the logarithmic likelihood function can be calculated in terms of the scalarized Algorithms 4 and 5 using Eq.* (18) *and allowing for validity of*

$$\ln\left[\det(R_{e,k})\right] = \sum_{i=1}^{m} \ln\left[\alpha_k^{(i)}\right], \quad k = 1, \ldots, N, \tag{19}$$

$$e_k^{\mathrm{T}} R_{e,k}^{-1} e_k = \sum_{i=1}^{m} \left(\bar{e}_k^{(i)}\right)^2, \quad k = 1, \ldots, N. \tag{20}$$

The proof can be found in the Appendix.

Remark 5. One can readily see that the scalarized Algorithms 4 and 5 with the suspension (18)–(20) enable one to solve not only the problem of estimating the state vector of system (1), (2), but also an additional problem of calculating the logarithmic likelihood function. Therefore, the extended filter functionality enables one simultaneously to perform estimation of the system state vector and identification (detection of malfunctions, and others).

5. COMPARATIVE ANALYSIS OF THE "VECTOR" AND "SCALAR" METHODS

Theorems 1 and 2 established equivalence of the two methods of filtering and calculation of the logarithmic likelihood function—the "vector" method where the observation vector z_k is processed entirely and the "scalar" one where it is processed componentwise. However, despite equivalence of these two approaches, the computational characteristics of the algorithm are essentially different. We compare below the "vector" and "scalar" methods.

Remark 6. The present paper considers only the covariance implementations of the discrete-time Kalman filter. For the combined square-root filtering algorithm this means that only the first variant of orthogonal transformations is used. In other words, it is assumed that in the formulas of "vector" (11) and "scalar" (17) measurement updates we use any orthogonal rotation that upper-triangularizes the first two (block) columns of the matrix on the left-hand side of, respectively, (11) and (17).

5.1. Estimation of the Computational Burden

Let us calculate (for one step) the total number of the arithmetical operations required to implement the proposed "scalar" Algorithms 4 and 5 and compare the data obtained with the similar data of their "vector" counterparts 2 and 3, respectively. One can readily see that the changes apply only to the stages of filtering. In this connection, we disregard below the stages of time updates, that is, (8) and (10), as well as the total number of arithmetical operations for calculation of the observation vector z_k. We consider separately the cases of using the modified Gram–Schmidt orthogonalization algorithm and the Hausholder orthogonalization method. The results are compiled in Tables 1 and 2.

Now we estimate the costs of determining the logarithmic likelihood function (18) and add to Tables 1 and 2 one more column indicating the amount of calculations introduced to the filter by new functionality. The data obtained enable us to estimate the total number of arithmetical operation required for modification of the basic implementation of the filter with the aim of solving the additional problem.

One can easily see from Tables 1 and 2 that in the case of $m \gg n$ the passage from the "vector" updating of the observation vector z_k to the componentwise ("scalar") one reduces dramatically the total computational burden. Namely, the number of arithmetical operations for the parameter m, where $z_k \in R^m$, required in the scalarized Algorithms 4 and 5 is smaller approximately by the order of two. As the result, the methods for calculation of the likelihood function inherit the advantages of the basic filtering algorithms.

We also note that the aforementioned conclusions are independent of the choice of one or another method of orthogonalization, that is, complexity of the "scalar" square-root implementations 4 and 5 and, correspondingly, the methods for calculation of the logarithmic likelihood function is much smaller than in the case of full updating of the vector z_k independently of the orthogonalization algorithm,

Example 1. Let us estimate in terms of seconds the computer runtime required to estimate the state vector of the system $\hat{x}_{k|k}$ and calculate the likelihood function (18) using the two above methods, that is, the "vector" and "scalar" calculations. We run two experimental series. In the

Table 1. Total number of arithmetical operations for one step (without the time updates stage) of "vector" and "scalar" filter algorithms and calculation of the logarithmic likelihood function (*using a modified Gram–Schmidt orthogonalization algorithm*)

Type of operation	Total number of arithmetical operations	
	data updating (filtering)	overhead
"Vector" Algorithm 2 and calculation of the likelihood function (18)		
multiplication	$\frac{1}{6}\left[6(m+n)^3+12(m+n)^2-6(m+n)\right]$	$2m$
division	$m+n$	0
subtraction and addition	$\frac{1}{6}\left[6(m+n)^3+3(m+n)^2-9(m+n)\right]$	$2(m-1)$
square root generation	$m+n$	0
"Vector" Algorithm 3 and calculation of the likelihood function (18)		
multiplication	$\frac{1}{6}\left[6(m+n)^3+6(n+2)(m+n)^2-6(m+n)\right]$	$2m$
division	$m+n$	0
subtraction and addition	$\frac{1}{6}\left[6(m+n)^3+3(2n-1)(m+n)^2-3(2n+3)(m+n)\right]$	$2(m-1)$
square root generation	$n+m$	0
"Scalar" Algorithm 4 and calculation of the likelihood function (18)–(20)		
multiplication	$\frac{1}{6}m\left[6(1+n)^3+12(1+n)^2-6(1+n)\right]$	$2m$
division	$m(n+1)$	0
subtraction and addition	$\frac{1}{6}m\left[6(1+n)^3+3(1+n)^2-9(1+n)\right]$	$2(m-1)$
square root generation	$m(n+1)$	0
"Scalar" Algorithm 5 and calculation of the likelihood function (18)–(20)		
multiplication	$\frac{1}{6}m\left[12(1+n)^3+6(1+n)^2-6(1+n)\right]$	$2m$
division	$m(n+1)$	0
subtraction and addition	$\frac{1}{6}m\left[6(1+n)^3+3(2n-1)(1+n)^2-3(2n+3)(1+n)\right]$	$2(m-1)$
square root generation	$m(n+1)$	0

Table 2. Total number of arithmetical operations for one step (without the time updates stage) of "vector" and "scalar" filter algorithms and calculation of the logarithmic likelihood function (*using a modified Gram–Schmidt orthogonalization algorithm*)

Type of operation	Total number of arithmetical operations: data updating (filtering)	overhead
	"Vector" Algorithm 2 and calculation of the likelihood function (18)	
multiplication	$\frac{1}{6}\left[4(m+n)^3+12(m+n)^2+2(m+n)\right]$	$2m$
division	$m+n$	0
subtraction and addition	$\frac{1}{6}\left[4(m+n)^3+6(m+n)^2+2(m+n)\right]$	$2(m-1)$
square root generation	$m+n$	0
	"Vector" Algorithm 3 and calculation of the likelihood function (18)	
multiplication	$\frac{1}{6}\left[4(m+n)^3+6(m+n)^2(n+2)+2(m+n)\right]$	$2m$
division	$m+n$	0
subtraction and addition	$\frac{1}{6}\left[4(m+n)^3+6(m+n)^2(n+1)+8n(m+n)\right]$	$2(m-1)$
square root generation	$n+m$	0
	"Scalar" Algorithm 4 and calculation of the likelihood function (18)–(20)	
multiplication	$\frac{1}{6}m\left[4(1+n)^3+12(1+n)^2+2(1+n)\right]$	$2m$
division	$m(n+1)$	0
subtraction and addition	$\frac{1}{6}m\left[4(1+n)^3+6(1+n)^2+2(1+n)\right]$	$2(m-1)$
square root generation	$m(n+1)$	0
	"Scalar" Algorithm 5 and calculation of the likelihood function (18)–(20)	
multiplication	$\frac{1}{6}m\left[4(1+n)^3+6(1+n)^2(2+n)+2(1+n)\right]$	$2m$
division	$m(n+1)$	0
subtraction and addition	$\frac{1}{6}m\left[4(1+n)^3+6n(1+n)^2+2(1+n)(4-3n)\right]$	$2(m-1)$
square root generation	$m(n+1)$	0

first case, the number of estimated system parameters is $n = 1$ and $F_k = I_1$, in the second, $n = 2$ and $F_k = I_2$, where I_n is the $n\times n$ identity matrix. The discrete time of filtering is $N = 100$. At each time instant, $k = 1,\ldots,N$ and the amount of the observed data is $M = 150, 200, 250, 300, 350, 400$.

The results of the experiment[3] are compiled in Table 3 showing for each value of M the time gain brought about by the developed scalarized algorithms as compared with their "vector" implementations. Now we calculate the percentage of the specific weight of the overhead introduced in the filter by its new functionality, that is, the costs of calculating the logarithmic likelihood function.

The generalized data of Table 3 corroborate the previous conclusion about advisability of the "scalar" implementations. For example, in the case of $n = 1$ and $M = 200$ the scalarized Algorithm 4 (see Eqs. (8), (16)) is faster by the factor of 5 as compared with its "vector" counterpart 2

[3] All experiments were carried out using the *MatLab* package run on a PC with Pentium-II processor with CPU clock 266 MHz.

Table 3. Processor time (in seconds) for filtering and calculation of the logarithmic likelihood function (LLF) by the "vector" and "scalar" algorithms

M	"Vector" Algorithms 2 and 3			"Scalar" Algorithms 4 and 5			Advantage	
							filter	LLF
	filter t_v^F	computation of LLF t_v^{LF}	overhead, %	filter t_s^F	computation of LLF t_s^{LF}	overhead, %	t_v^F/t_s^F	t_v^{LF}/t_s^{LF}
				Random walk, $n=1$				
150	27.95	32.07	12.9	8.46	9.23	8.3	3.30	3.47
200	56.14	79.70	29.6	11.20	12.25	8.6	5.02	6.50
250	108.86	164.34	33.8	14.01	15.32	8.6	7.77	10.73
300	218.44	270.84	19.3	16.97	18.68	9.2	12.87	14.50
350	343.37	421.89	18.6	19.61	22.74	13.8	17.51	18.55
400	447.62	786.24	43.0	22.63	25.49	11.2	19.78	30.85
				Random walk, $n=2$				
150	32.68	36.09	9.4	9.17	10.00	8.3	3.56	3.61
200	57.84	73.10	20.9	12.52	13.35	6.2	4.62	5.48
250	113.42	150.72	24.7	15.33	16.59	7.6	7.40	9.09
300	191.86	257.54	25.5	18.51	20.22	8.5	10.37	12.34
350	350.77	481.65	27.2	22.63	24.38	7.2	15.50	19.76
400	456.87	815.54	44.0	25.60	26.85	4.7	17.85	30.37

(see (8), (9)), and calculation of the logarithmic likelihood function is faster already by the factor 6.5. For $M = 400$, the speed increases approximately by the factors of 20 and 31.

The following fact deserves noting in conclusion. In Example 1, introduction of a small number of additional calculations amounting approximately to 10% of the total cost enables one to modify the above filtering algorithms with scalar updates 4 and 5 so that, in addition to their main task of estimating the state vector of system $\hat{x}_{k|k}$, they enable calculations of the likelihood function and thus introduce a new functionality.

5.2. Stability of the Roundoff Errors

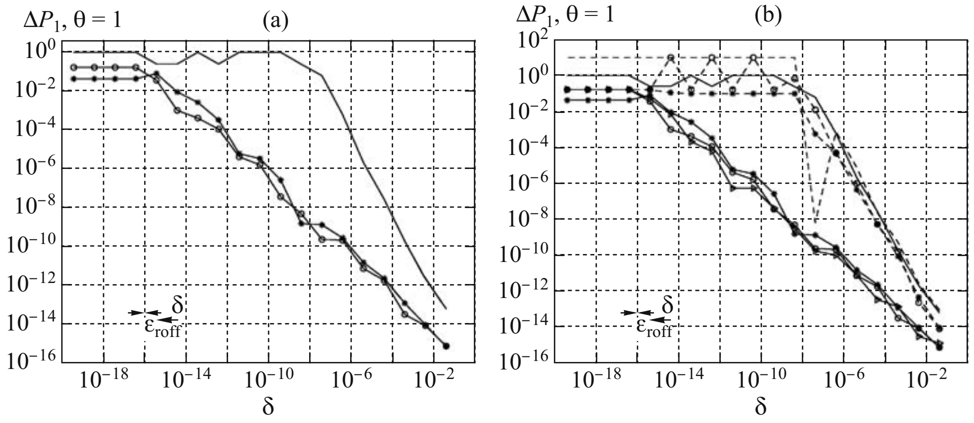
Computer-aided calculations inevitably lead to roundoff errors. To demonstrate their impact on the results of the "scalar" implementations of the discrete Kalman filter and compare them with the existing methods, we consider the popular benchmark [9].

Example 2. We consider the problem of estimating an unknown system parameter θ by the method of maximum likelihood. The parameter matrices of system (1), (2) are defined as

$$F_k = I_3, \quad Q_k = 0, \quad G_k = 0, \quad R_k = \delta^2\theta I_2 \quad \text{and} \quad H_k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\delta \end{bmatrix}$$

with the initial conditions are $x_0 \sim \mathcal{N}(0, \theta I_3)$, where $\delta^2 < \epsilon_{\text{roff}}$, but $\delta > \epsilon_{\text{roff}}$. ϵ_{roff} is unit roundoff error, that is, $\epsilon_{\text{roff}} + 1 \neq 1$, but $\epsilon_{\text{roff}}/2 + 1 \overset{\text{roff}}{=} 1$.
<u>Calculate</u> the value of the covariance matrix of the estimation error P_k after updating one measurement under some fixed value of the parameter θ, say $\theta = 1$. In other words, calculate $P_1|_{\theta=1}$.

This benchmark was not chosen by mere chance. The set of the ill-conditioned problems of Example 2 presents difficulties only to the filters of the covariance types discussed in the present paper (see Remark 6).



Loss of calculation precision by some implementations of the discrete Kalman filter: conventional Kalman filter (solid line), "vector" algorithms 2 and 3 (solid line with marker ∘), "scalar" Algorithms 4 and 5 (solid line with marker ∗), Joseph stabilized version (dashed line with marker ∘), Swerling algorithm (dashed line), square-root Potter algorithm (dashed line with marker ∗), and Bierman *UD* filter (solid line with marker ▷).

One can readily see that under the conditions of Example 2 the matrix H_k has rank 2. We calculate the covariance matrix of the measurement innovation after updating the first measurement, that is, $R_{e,1} = H_1\Pi_0 H_1^{\mathrm{T}} + R_1$, for $\delta \to \epsilon_{\text{roff}}$. As the result, we obtain

$$\begin{array}{ccc} & \text{exact value} & \text{rounded value} \\ R_{e,1}|_{\theta=1}: & \begin{bmatrix} 3+\delta^2 & 3+\delta \\ 3+\delta & 2+(1+\delta)^2+\delta^2 \end{bmatrix} & \begin{bmatrix} 3 & 3+\delta \\ 3+\delta & 3+2\delta \end{bmatrix}. \end{array}$$

For smaller δ, therefore, the matrix $R_{e,1}|_{\theta=1}$ is ill-conditioned, and as the result of roundoff, for $\delta \to \epsilon_{\text{roff}}$ its rank is one. Reference [8] gives a strict theoretical analysis of the influence of the roundoff errors on some implementations of the discrete Kalman filter. In particular, its authors proved that the condition number of the covariance matrix of the measurement innovation $R_{e,k}$ is the key parameter influencing the numerical behavior of the covariance filters (for more detail, see [8]). Example 2 illustrates this situation. For $\delta \to \epsilon_{\text{roff}}$, the conventional Kalman algorithm, as well as any other its covariance-type implementation are unable to handle this problem.

The following experiment is carried out to illustrate the loss of calculation precision by various implementations of the discrete Kalman filter in the conditions of Example 2. We calculate the error covariance matrix $P_1|_{\theta=1}$ using the conventional implementation of the discrete Kalman filter (3)–(7), the "vector" matrix square-root Algorithms 2 and 3 (see Eqs. (8)–(11)) and their scalarized implementations 4 and 5 (see (8), (16) (10), and (17)) developed in the present paper. For each value of δ, among the elements of the matrix $P_1|_{\theta=1}$ we calculate by means of computer the maximal absolute error ΔP_1 by comparing the exact solution with the numerical solution. The results of this experimental series are plotted in the figure. For each implementation of the discrete Kalman filter using in the comparative analysis, depicted is ΔP_1 (we recall that $\theta = 1$) vs. the conditionality parameter δ, where $\delta \to \epsilon_{\text{roff}}$ is the parameter of the computer roundoff. Different lines in the figure correspond to different implementations of the filter.

The calculations of the above example were carried out on a PC with representation of numbers with the relative error 10^{-16}. The program codes were written in *MatLab* where the computer roundoff parameter ϵ_{roff} is stored in the variable $\texttt{eps}$ with the value $\epsilon_{\text{roff}} = \texttt{eps}/2 = 1.1102 \times 10^{-16}$. The value of δ satisfying the conditions $\delta^2 < \epsilon_{\text{roff}}$, but $\delta > \epsilon_{\text{roff}}$ is, for example, $\delta = \texttt{eps}^{2/3}$. In this connection, the calculations were carried out on a set of ill-conditioned problems of Example 2 for the values of δ over the interval $[10^{-9}\texttt{eps}^{2/3},\ 10^{9}\texttt{eps}^{2/3}]$.

Let us turn to the results of this experiment and first consider the left graphs in the figure. One can readily see that the conventional Kalman filter (solid line) is incapable of fulfilling the formulated problem. For $\delta \to \epsilon_{\text{roff}}$ is loses precision of calculations faster than the rest of implementations. For example, for $\delta = 10^{-8}$ the conventional implementation of the filter calculates $P_1|_{\theta=1}$ to within 10^{-1}, whereas the "vector" (solid line with marker $\circ$) and proposed "scalar" (solid line with marker $*$) array filtering algorithms have accuracy 10^{-9}. Therefore, the conventional algorithm fails to give even one correct decimal digit already for $\delta = 10^{-8}$ and further, which is indicative of its full inoperability. The "vector" Algorithms 2 and 3 and their scalarized counterparts developed above succeed in maintaining an acceptable accuracy of calculations up to the boundary value $\delta = \epsilon_{\text{roff}}$, machine precision limit.

Let us discuss the results of the second experimental series, that is, the right graphs in the figure. The following existing implementations of the discrete Kalman filter were used for comparison: the Joseph stabilized algorithm (dashed line with marker $\circ$), Swerling algorithm (dashed line), Potter square-root algorithm (dashed line with marker $*$), and Bierman UD filter (solid line with marker $\triangleright$).

One can easily see that the posed problem is dealt best by the square-root array filtering algorithms ("vector" and "scalar") and also the Bierman UD filter. They have actually the same precision of calculations for all values of $\delta \to \epsilon_{\text{roff}}$ differ just a little one from another. Among the rest of the implementations of the discrete Kalman filter, only the Potter algorithm belongs to the class of the square-root methods. Despite this fact, it is much inferior to the aforementioned implementation in terms of calculation precision mostly because of the loss of special superdiagonal or subdiagonal form by the matrices $P_{k|k}^{1/2}$. The Joseph and Swerling algorithms belong to the class of conventional (or full) covariance methods, that is, are formulated in terms of the covariance matrices $P_{k|k}$ and $P_{k|k-1}$. The graph demonstrates their inability to solve the formulated problem. Therefore, under the conditions of Example 2 the discussed matrix square-root Algorithms 2 and 3, as well as their "scalar" counterparts 4 and 5 and the Bierman UD filter represent the implementations of the discrete Kalman filter that are least sensitive to the roundoff errors.

We conclude by noting that the "scalar" method of filtering as proposed in the present paper (Algorithms 4 and 5) and the method of calculation of the likelihood function (18)–(20) enable dramatic reduction in the computational complexity of the algorithms (in the case of $m \ll n$) at the same precision of calculations as with the "vector" calculations (Algorithms 2 and 3 and Eq. (18), respectively).

6. CONCLUSIONS

Two methods of effective calculation of the logarithmic likelihood function were developed for the linear discrete-time stochastic systems corrupted by white Gaussian noises. The algorithms are based on the numerically stable square-root implementation methods for Kalman filtering formulas with the use of the scalar measurement updates. This procedure enables substantial reduction in the total number of arithmetical operations for implementation without loss in the calculation precision. The fact that the algorithms are based on the array square-root implementations of the discrete-time Kalman filter and, as the result, are oriented to parallel calculations is their additional advantage. All the aforementioned advantages make these methods preferable for introduction and use in practice.

APPENDIX

Proof of Theorem 1. It is necessary to prove that the measurement updates of scalarized Algorithms 4, 5 and conventional Kalman filter (5)–(7) are algebraically equivalent. We make use of Lemma 1 and demonstrate equivalence of (16), (17) to Eqs. (12)–(15).

For convenience of further presentation, we denote by A_k and B_k the matrices in the left-hand and right-hand sides of (16) and (17), respectively. Then, it follows from (16), (17) and the properties of orthogonal transformations that

$$A_k^{\mathrm{T}} \left(O_k^{(i)}\right)^{\mathrm{T}} O_k^{(i)} A_k = A_k^{\mathrm{T}} A_k = B_k^{\mathrm{T}} B_k. \tag{A.1}$$

By elementwise comparison of the matrices $A_k^{\mathrm{T}} A_k$ and $B_k^{\mathrm{T}} B_k$ one can easily establish that (12)–(15) is valid. Indeed, equality (12) is valid for the element $(1,1)$ of Algorithms 4 and 5:

$$\left(\sqrt{\alpha_k^{(i)}}\right)^2 = \left(h_k^{(i)}\right)^{\mathrm{T}} P_k^{(i-1)} h_k^{(i)} + \left(r_k^{(i)}\right)^2.$$

For the elements $(1,2) = (2,1)$, we obtain

$$\overline{K}_{p,k}^{(i)} \sqrt{\alpha_k^{(i)}} = P_k^{(i-1)} h_k^{(i)} \quad \Longrightarrow \quad K_{p,k}^{(i)} \alpha_k^{(i)} = P_k^{(i-1)} h_k^{(i)} \tag{A.2}$$

which is nothing more nor less than Eq. (13). The following is true for the element $(2,2)$:

$$\overline{K}_{p,k}^{(i)} \left(\overline{K}_{p,k}^{(i)}\right)^{\mathrm{T}} + P_k^{(i)} = P_k^{(i-1)}. \tag{A.3}$$

By allowing for (A.2) and symmetricity of the matrices $P_k^{(i)}$, $i = 1, \ldots, m$, we prove that (A.3) is Eq. (14) of the scalarized conventional implementation of the discrete Kalman filter. Indeed,

$$P_k^{(i)} = P_k^{(i-1)} - \overline{K}_{p,k}^{(i)} \left(\overline{K}_{p,k}^{(i)}\right)^{\mathrm{T}} = P_k^{(i-1)} - K_{p,k}^{(i)} \sqrt{\alpha_k^{(i)}} \left(\overline{K}_{p,k}^{(i)}\right)^{\mathrm{T}} = P_k^{(i-1)} - K_{p,k}^{(i)} \left(h_k^{(i)}\right)^{\mathrm{T}} P_k^{(i-1)}.$$

According to (16), the estimate of the state vector $\hat{x}_k^{(i)}$ is updated recurrently:

$$\hat{x}_k^{(i-1)} = -\overline{K}_{p,k}^{(i)} \bar{e}_k^{(i)} + \hat{x}_k^{(i)} \quad \Longrightarrow \quad \hat{x}_k^{(i)} = \hat{x}_k^{(i-1)} + \overline{K}_{p,k}^{(i)} \bar{e}_k^{(i)}, \tag{A.4}$$

where

$$-\sqrt{\alpha_k^{(i)}} \bar{e}_k^{(i)} = -z_k^{(i)} + \left(h_k^{(i)}\right)^{\mathrm{T}} \hat{x}_k^{(i-1)}. \tag{A.5}$$

It follows from (A.4), (A.5) that (15) is true:

$$\hat{x}_k^{(i)} = \hat{x}_k^{(i-1)} + \overline{K}_{p,k}^{(i)} \bar{e}_k^{(i)} = \hat{x}_k^{(i-1)} + K_{p,k}^{(i)} \left[z_k^{(i)} - \left(h_k^{(i)}\right)^{\mathrm{T}} \hat{x}_k^{(i-1)}\right].$$

Consequently, (16) is equivalent to (12)–(15). Finally, by Lemma 1 the stages of measurement updating of the extended square-root covariance filter with scalar updates 4 and the conventional implementation of the Kalman filter (Algorithm 1) are algebraically equivalent. For the scalarized combined square-root filter (Algorithm 5), validity of (12)–(15) is proved in a similar manner. However, this implementation method is classified with the combined algorithms. One can readily see its measurement updates (17) were obtained from (16) by adding a block column, which enables one to calculate along with the covariance values $P_{k|k-1}^{1/2}$ and $P_{k|k}^{1/2}$ also the information matrices

$P_{k|k-1}^{-1/2}$ and $P_{k|k}^{-1/2}$. For the componentwise updating of the measurement vector z_k, the information matrix $P_{k|k}^{-1}$ is updated recursively (see, for example, [9]):

$$\left(P_k^{(i)}\right)^{-1} = \left(P_k^{(i-1)}\right)^{-1} + h_k^{(i)} \left(h_k^{(i)}\right)^{\mathrm{T}} / \left(\sigma_k^{(i)}\right)^2. \tag{A.6}$$

Therefore, it remains to prove validity of (A.6) for Algorithm 5. It follows from (17) and the properties of the orthogonal transformations that

$$\left(P_k^{(i)}\right)^{-1/2} \left(P_k^{(i)}\right)^{-\mathrm{T}/2} = \left(P_k^{(i-1)}\right)^{-1/2} \left(P_k^{(i-1)}\right)^{-\mathrm{T}/2} + h_k^{(i)} \left(h_k^{(i)}\right)^{\mathrm{T}} / \left(r_k^{(i)}\right)^2,$$

which proves Theorem 1.

Proof of Theorem 2. It is required to prove that (19) and (20) are valid for Algorithms 4 and 5 with scalar updates. The proof is carried out by induction by dimension of the observation vector $z_k \in \mathbb{R}^m$, and first we prove the theorem in what concerns formula (19).

(1) For $m = 1$, validity of (19) is evident. Indeed, with regard for Algorithms 4 and 5, namely, formulas (8), (16) and (10), (17), we obtain

$$\ln\left[\det(R_{e,k})\right] \overset{def}{=} \ln\left[\left(h_k^{(1)}\right)^{\mathrm{T}} P_{k|k-1} h_k^{(1)} + \left(r_k^{(1)}\right)^2\right] = \ln\left[\alpha_k^{(1)}\right].$$

(2) Let the theorem be true for some natural number $m = l$, that is,

$$\ln\left[\det(R_{e,k})\right] = \sum_{i=1}^{l} \ln\left[\alpha_k^{(i)}\right], \tag{A.7}$$

where $R_{e,k}$ is the $l \times l$ residual covariance matrix of the discrete Kalman filter. We prove validity of (19) also for the next $m = l+1$ and assume for that l components of the vector z_k already have been processed, that is, that is remains to process the last, $(l+1)$st component of the observation vector. We decompose the matrices H_k, R_k and the observation vector z_k into blocks as follows:

$$H_k = \left[\left(H_k^{l\times n}\right)^{\mathrm{T}}, \left(h_k^{(l+1)}\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad R_k = \mathrm{diag}\left\{R_k^{l\times l}, \left(r_k^{(l+1)}\right)^2\right\}, \quad z_k = \left[z_k^{l\times 1}, z_k^{(l+1)}\right]^{\mathrm{T}},$$

where the superscripts indicate the dimension of the corresponding blocks, that is, $H_k^{l\times n}$ is a block of the matrix H_k consisting of the l first rows and n first columns of H_k. The same notation is used for $R_k^{l\times l}$, $z_k^{l\times 1}$. The value $h_k^{(l+1)}$ denotes the $l+1$st row of the matrix H_k and $z_k^{(l+1)}$ is the $l+1$st component of the vector z_k. We use the extended square-root covariance algorithm with scalar updates 4 (only the measurement updates defined by Eq. (16)):

$$O_{k,2}\left[\begin{array}{ccc|c} \left(R_k^{l\times l}\right)^{1/2} & 0 & 0 & -\left(R_k^{l\times l}\right)^{-\mathrm{T}/2} z_k^{l\times 1} \\ 0 & r_k^{(l+1)} & 0 & -z_k^{(l+1)}/r_k^{(l+1)} \\ P_{k|k-1}^{1/2}\left(H_k^{l\times n}\right)^{\mathrm{T}} & P_{k|k-1}^{1/2}\left(h_k^{(l+1)}\right)^{\mathrm{T}} & P_{k|k-1}^{1/2} & P_{k|k-1}^{-\mathrm{T}/2}\hat{x}_{k|k-1} \end{array}\right]$$

$$= \left[\begin{array}{ccc|c} \left(R_{e,k}^{l\times l}\right)^{1/2} & 0 & \left(\overline{K}_{p,k}^{l\times n}\right)^{\mathrm{T}} & -\bar{e}_k^{l\times 1} \\ 0 & \gamma_k & \left(\overline{K}_{p,k}^{1\times n}\right)^{\mathrm{T}} & -\bar{e}_k^{(l+1)} \\ 0 & 0 & P_{k|k}^{1/2} & P_{k|k}^{-\mathrm{T}/2}\hat{x}_{k|k} \end{array}\right], \tag{A.8}$$

where γ_k is some value obtained as the result of applying the orthogonal transformation $O_{k,2}$ to the matrix in the left-hand side of (A.8). We note that validity of (A.8) is evident for the scalarized combined square-root Algorithm 5.

So, $\left(R_{e,k}^{(l+1)\times(l+1)}\right)^{1/2}$ is a block-diagonal matrix. Then,

$$R_{e,k}^{(l+1)\times(l+1)} = \left(R_{e,k}^{(l+1)\times(l+1)}\right)^{\mathrm{T}/2}\left(R_{e,k}^{(l+1)\times(l+1)}\right)^{1/2} = \operatorname{diag}\left\{R_{e,k}^{l\times l}, \gamma_k^2\right\},$$

where the assumptions of induction are satisfied for the $l \times l$ submatrix $R_{e,k}^{l\times l}$. With regard for the above, it follows from validity of (A.7) and the properties of the determinant that

$$\ln\left[\det\left(R_{e,k}^{(l+1)\times(l+1)}\right)\right] = \ln\left[\det\left(R_{e,k}^{l\times l}\right)\gamma_k^2\right] = \sum_{i=1}^{l}\ln\left[\alpha_k^{(i)}\right] + \ln\left[\gamma_k^2\right]. \tag{A.9}$$

It remains to demonstrate that γ_k^2 in the scalarized Algorithms 4 and 5 is nothing more nor less than $\alpha_k^{(l+1)}$. Indeed, taking into consideration (A.8), we obtain that

$$\gamma_k^2 = \left(h_k^{(l+1)}\right)^{\mathrm{T}} P_{k|k-1} h_k^{(l+1)} + \left(r_k^{(l+1)}\right)^2,$$

where $P_{k|k-1}$ is the error covariance matrix obtained by updating the l first components of the observation vector z_k. From Lemma 1 of equivalence of the "scalar" method of filtering, that is, (12)–(15), and the conventional implementation of the Kalman filter (5)–(7), it follows that $P_{k|k-1} = P_k^{(l)}$. Then,

$$\gamma_k^2 = \left(h_k^{(l+1)}\right)^{\mathrm{T}} P_k^{(l)} h_k^{(l+1)} + \left(r_k^{(l+1)}\right)^2. \tag{A.10}$$

Taking into account (12) and (A.10), we obtain

$$\gamma_k^2 = \alpha_k^{(l+1)}. \tag{A.11}$$

Finally, it follows from (A.9) and (A.11) that

$$\ln\left[\det\left(R_{e,k}\right)\right] = \sum_{i=1}^{l}\ln\left[\alpha_k^{(i)}\right] + \ln\left[\alpha_k^{(l+1)}\right] = \sum_{i=1}^{l+1}\ln\left[\alpha_k^{(i)}\right],$$

which is what we set out to prove.

It remains to demonstrate that equality (20) is true for Algorithms 4 and 5.

(1) For the case of $m = 1$, (20) is evident. Indeed, since $m = 1$, the vector e_k consists of one component and $R_{e,k}$ is a scalar value. Taking into consideration the implementations with scalar updates (8), (16) and (10), (17), we obtain that

$$e_k^{\mathrm{T}} R_{e,k}^{-1} e_k \overset{def}{=} \frac{e_k^{(1)} e_k^{(1)}}{\left(h_k^{(1)}\right)^{\mathrm{T}} P_{k|k-1} h_k^{(1)} + \left(r_k^{(1)}\right)^2} = \frac{e_k^{(1)} e_k^{(1)}}{\alpha_k^{(1)}} = \frac{e_k^{(1)}}{\sqrt{\alpha_k^{(1)}}}\frac{e_k^{(1)}}{\sqrt{\alpha_k^{(1)}}} = \left(\bar{e}_k^{(1)}\right)^2.$$

(2) Let the theorem be true for some natural $m = l$, that is, valid be

$$e_k^{\mathrm{T}} R_{e,k}^{-1} e_k = \sum_{i=1}^{l}\left(\bar{e}_k^{(i)}\right)^2, \tag{A.12}$$

where $e_k \in \mathbb{R}^l$ and $R_{e,k} \in \mathbb{R}^{l\times l}$. We prove (20) also for the next $m = l + 1$.

We assume as above that l components of the vector z_k were already processed. As above, by decomposing into blocks the matrices H_k, R_k and the vector z_k, we again consider equality (A.8) and establish that

$$R_{e,k}^{(l+1)\times(l+1)} = \left(R_{e,k}^{(l+1)\times(l+1)}\right)^{\mathrm{T}/2}\left(R_{e,k}^{(l+1)\times(l+1)}\right)^{1/2} = \operatorname{diag}\left\{R_{e,k}^{l\times l}, \gamma_k^2\right\}$$

is true for the matrix $R_{e,k} \in \mathbb{R}^{(l+1)\times(l+1)}$ (the superscripts are used to emphasize dimensions of the corresponding matrix blocks). It follows from the properties of the diagonal matrices that

$$\left(e_k^{(l+1)\times 1}\right)^{\mathrm{T}}\left(R_{e,k}^{(l+1)\times(l+1)}\right)^{-1} e_k^{(l+1)\times 1} = \left(e_k^{l\times 1}\right)^{\mathrm{T}}\left(R_{e,k}^{l\times l}\right)^{-1} e_k^{l\times 1} + \frac{\left(e_k^{(l+1)}\right)^2}{\gamma_k^2}.$$

For $e_k \in \mathbb{R}^l$ and $R_{e,k} \in \mathbb{R}^{l\times l}$, (A.12) is valid by the assumption of induction. Then

$$\left(e_k^{(l+1)\times 1}\right)^{\mathrm{T}}\left(R_{e,k}^{(l+1)\times(l+1)}\right)^{-1} e_k^{(l+1)\times 1} = \sum_{i=1}^{l}\left(\bar{e}_k^{(i)}\right)^2 + \frac{\left(e_k^{(l+1)}\right)^2}{\gamma_k^2}, \tag{A.13}$$

and $\gamma_k^2 = \alpha_k^{(l+1)}$ was proved above. By substituting in (A.13) the expression of γ_k, we finally obtain

$$e_k^{\mathrm{T}} R_{e,k}^{-1} e_k = \sum_{i=1}^{l}\left(\bar{e}_k^{(i)}\right)^2 + \frac{e_k^{(l+1)}}{\sqrt{\alpha_k^{(l+1)}}}\frac{e_k^{(l+1)}}{\sqrt{\alpha_k^{(l+1)}}} = \sum_{i=1}^{l}\left(\bar{e}_k^{(i)}\right)^2 + \left(\bar{e}_k^{(l+1)}\right)^2 = \sum_{i=1}^{l+1}\left(\bar{e}_k^{(i)}\right)^2,$$

which proves Theorem 2.

REFERENCES

1. Khazen, E.M., *Metody optimal'nykh statisticheskikh reshenii i zadachi optimal'nogo upravleniya* (Methods of Optimal Statistical Solutions and Problems of Optimal Control), Moscow: Sovetskoe Radio, 1968.

2. Van Trees, H.L., *Detection, Estimation, and Modulation Theory. Part II. Theory of Nonlinear Modulation*, New York: Wiley, 1971. Translated under the title *Teoriya obnaruzheniya, otsenok i modulyatsii. T. 2. Teoriya nelineinoi modulyatsii*, Moscow: Sovetskoe Radio, 1975.

3. Basseville, M. and Nikiforov, I., *Detection of Abrupt Changes: Theory and Applications*, New Jersey: Prentice-Hall, 1993.

4. Gupta, N.K. and Mehra, R.K., Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations, *IEEE Trans. Automat. Control*, 1974, vol. AC-19, no. 6, pp. 774–783.

5. Schweppe, F.C., Evaluation of Likelihood Functions for Gaussian Signals, *IEEE Trans. Inform. Theory*, 1965, vol. IT-11, no. 1, pp. 61–70.

6. Kaminski, P.G., Bryson, A.E., and Schmidt, S.F., Discrete Square-root Filtering: A Survey of Current Techniques, *IEEE Trans. Automat. Control*, 1971, vol. AC-16, no. 6, pp. 727–735.

7. Park, P. and Kailath, T., New Square-root Algorithms for Kalman Filtering, *IEEE Trans. Automat. Control*, 1995, vol. 40, no. 5, pp. 895–899.

8. Verhaegen, M. and Van Dooren, P., Numerical Aspects of Different Kalman Filter Implementations, *IEEE Trans. Automat. Control*, 1986, vol. AC-31, no. 10, pp. 907–917.

9. Grewal, M.S. and Andrews, A.P., *Kalman Filtering: Theory and Practice*, New Jersey: Prentice-Hall, 2001.

10. Chui, C., Cheh, G., and Chui, H., Modified Extended Kalman Filtering and Real-Time Parallel Algorithm Error System Parameter Identification, *IEEE Trans. Automat. Control*, 1991, vol. 35, no. 1, pp. 100–104.

11. Lee, E.K.B. and Maykin, S., Parallel Implementation of the Extended Square-root Covariance Filter for Tracking Applications, *IEEE Trans. Parallel Distr. Syst.*, 1993, vol. 4, no. 3, pp. 446–457.

12. Semoushin, I.V. and Tsyganova, J.V., An Efficient Way to Evaluate Likelihood Functions in Terms of Kalman Filter Variables, *Proc. Adaptive, Cooperat. Competitive Proc. Syst. Modelling, Design Anal.*, Windsor, Ontario, Canada, 2002, pp. 67–74.

13. Kulikova, M.V., On Effective Computation of the Logarithm of the Likelihood Ratio Function for Gaussian Signals, in *Lecture Notes Comput. Sci.*, 2003, vol. 2658, pp. 427–435.

14. Lind, J.T., Repeated Surveys and the Kalman Filter, *Econometric J.*, 2005, vol. 8, pp. 418–427.

This paper was recommended for publication by A.V. Nazin, a member of the Editorial Board