Numerically Robust SVD-based Kalman Filter Implementations

Maria V. Kulikova
Center for Computational and Stochastic Mathematics
Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal
maria.kulikova at ist.utl.pt

***Abstract*—The so-called *factored-form* Kalman filter (KF) implementations are designed to deal with the problem of numerical instability of the conventional KF. They include Cholesky factorization-based, UD-based and singular value decomposition (SVD) algorithms. The SVD-based estimators are the most recent developments in this realm. They were shown to be more robust with respect to roundoff than the classical KF implementation and the previously derived factored-form methods. This paper discusses further improvements in estimation accuracy and numerical robustness of the recently proposed SVD-based estimators.**

***Index Terms*—Kalman filter, singular value decomposition, numerical robustness**

I. INTRODUCTION

In the Kalman filtering (KF) community, the so-called *factored-form* (square-root) algorithms are the preferred implementations for treating a numerical instability problem of the classical KF implementation [1]. The key idea of square-root (SR) algorithms is to ensure the symmetric form and positive semi-definiteness of error covariance matrix P by decomposing it in the form $P = SS^T$ and, next, re-formulating the filter equations in terms of the resulted factors, only [2, Chapter 7]. Additionally, the Cholesky-based SR methodology allows the computations with double precision as discussed in [3]. In summary, the factorization in the form $P = SS^T$ yields a wide variety of the KF implementation methods, among which the most popular are the Cholesky and the UD factorization-based methods [4]–[7]. The most recent development in this realm is the SVD-based KF implementations proposed in [8], [9].

Although considerable research has been devoted to Cholesky- and UD-based filter implementations, rather less attention has been paid to discussion of robust SVD-based methods. So far, investigations have been confined to *linear* SVD-based filtering methods in [8], [9] and to *nonlinear* SVD-based strategy in [10]–[12] where some evidences of its better estimation quality have been revealed. Concerning *linear* SVD-based filters, the newly-developed SVD-based KF implementation in [9] was shown to outperform its earlier published counterpart in [8] as well as the conventional KF and numerically stable Cholesky- and UD-based KF methods for estimation accuracy and robustness (with respect to roundoff).

The author acknowledges the support from the Portuguese National Fund (*Fundação para a Ciência e a Tecnologia*) within project UID/Multi/04621/2013.

In this paper we discuss the class of SVD-based KF implementations in detail, provide their comparative study and suggest the way of enhancing their numerical stability when solving ill-conditioned state estimation problem.

The paper is organized as follows. Section II contains the problem statement. The main result and discussion of the class of SVD-based filtering algorithms can be found in Section III. Section IV represents the outcome of numerical experiments and Section V concludes this work.

II. PROBLEM STATEMENT

Consider a linear discrete-time stochastic model

$$x_k = F_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \quad (1)$$

$$y_k = H_k x_k + v_k, \; w_k \sim \mathcal{N}(0, \Theta_k), \; v_k \sim \mathcal{N}(0, R_k) \quad (2)$$

where $F_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times d}$, $G_k \in \mathbb{R}^{n\times q}$ and $H_k \in \mathbb{R}^{m\times n}$ are known at each time instance t_k. The vectors $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^d$ and $y_k \in \mathbb{R}^m$ are unknown dynamic state, known deterministic input and available measurement vector, respectively. Random variables x_0, w_k and v_k are assumed to be normally distributed and satisfy the following properties:

$$\begin{aligned} &\mathbf{E}\{x_0\} = \bar{x}_0, && \mathbf{E}\{(x_0-\bar{x}_0)(x_0-\bar{x}_0)^T\} = \Pi_0, \\ &\mathbf{E}\{w_k\} = \mathbf{E}\{v_k\} = 0, && \mathbf{E}\{w_k x_0^T\} = \mathbf{E}\{v_k x_0^T\} = 0, \\ &\mathbf{E}\{w_k v_k^T\} = 0, && \mathbf{E}\{w_k w_j^T\} = \Theta_k \delta_{kj}, \\ & && \mathbf{E}\{v_k v_j^T\} = R_k \delta_{kj} \end{aligned}$$

where covariance matrices $\Theta_k \in \mathbb{R}^{q\times q}$ and $R_k \in \mathbb{R}^{m\times m}$ are known. The symbol δ_{kj} is the Kronecker delta function.

The classical KF applied for estimating the hidden dynamic state process $\{x_k\}_{k=1}^N$ of *linear Gaussian* state-space model from the observed sequence $\{y_k\}_{k=1}^N$, yields the minimum expected mean square error estimate $\{\hat{x}_{k|k}\}_{k=1}^N$. The quantity $\hat{x}_{k|k}$ stands for state estimate at time instance t_k, given the available measurements $\{y_1, \dots, y_k\}$. The classical KF recursion is given as follows [13, Theorem 9.2.1]:

Algorithm 1. KF (*Conventional KF implementation*)

INITIALIZATION: $(k = 0)$ $\hat{x}_{0|0} = \bar{x}_0$ and $P_{0|0} = \Pi_0$.

TIME UPDATE: $(k = \overline{1, N})$ $\triangleright$ PRIORI ESTIMATION

1 $\quad \hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}$,

2 $\quad P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + G_{k-1}\Theta_{k-1}G_{k-1}^T$.

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

MEASUREMENT UPDATE: ($k = \overline{1,N}$) $\triangleright$ POSTERIORI

3 $\quad R_{e,k} = H_k P_{k|k-1} H_k^T + R_k,$ $\quad\triangleright$ ESTIMATION

4 $\quad K_k = P_{k|k-1} H_k^T R_{e,k}^{-1},$

5 $\quad \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k$ where $e_k = y_k - H_k \hat{x}_{k|k-1},$

6 $\quad P_{k|k} = (I - K_k H_k) P_{k|k-1}.$

The innovations of the KF are defined as $e_k = y_k - H\hat{x}_{k|k-1}$. It is worth noting here that $R_{e,k} = \mathbf{E}\{e_k e_k^T\}$ and $e_k \sim \mathcal{N}(0, R_{e,k})$ for Gaussian state-space models.

The main aspect to be explored in this paper is a numerical robustness of the class of SVD-based KF algorithms compared to the conventional KF implementation and other SR methods.

III. THE SVD-BASED KALMAN FILTERING

The SVD factorization-based KF implementations utilize the decomposition of symmetric and positive semi-definite matrix P in the form of $P = Q\Sigma Q^T$ where Q is an orthogonal matrix and Σ is a diagonal matrix containing singular values. Thus, the SVD-based implementations belong to the examined SR family, because the square-root factor of P can be easily defined as follows: $S = Q\Sigma^{1/2}$.

Definition 1 (see Theorem 1.1.6 in [14])**.** *Every matrix $A \in \mathbb{C}^{m\times n}$ of rank r can be written as follows:*

$$A = W\Sigma V^*, \; \Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{m\times n}, \; S = \mathrm{diag}\{\sigma_1, \ldots, \sigma_r\}$$

where $W \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary matrices, V^* is the conjugate transpose of V, and $S \in \mathbb{R}^{r\times r}$ is a real nonnegative diagonal matrix. Here $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ are called the singular values of A. (Note that if $r = n$ and/or $r = m$, some of the zero submatrices in Σ are empty.)

The SVD factorization-based KF methods have been recently developed in [8], [9]. The basic steps for designing such implementations is to factorize at the initialization step the matrix $\Pi_0 = Q_{\Pi_0} D_{\Pi_0} Q_{\Pi_0}^T$ where Q_{Π_0} and D_{Π_0} are an orthogonal and a diagonal matrices, respectively. Recall, the diagonal matrix D_{Π_0} contains the singular values of Π_0. Next, the classical KF equations (Algorithm 1) are re-formulated in terms of the SVD factors $Q_{P_{k|k}}$ and $D_{P_{k|k}}^{1/2}$, which are recursively updated in each iteration step instead of calculating $P_{k|k}$. As all SR methods, this strategy improves the estimation accuracy and numerical robustness with respect to roundoff errors in ill-conditioned situations. The SVD is known as the most accurate decomposition method, especially when the matrix to be factorize is close to singular; see also the results of numerical tests in [9].

Modern KF implementation methods are often expressed in *array* form. This means that each KF iterate of the SVD-based algorithms has the form of $A = \mathfrak{W}\Sigma\mathfrak{V}^T$ where $A \in \mathbb{R}^{(k+s)\times s}$ is given pre-array and the resulted post-array SVD factors are defined as follows: $\mathfrak{W} \in \mathbb{R}^{(k+s)\times(k+s)}$, $\Sigma \in \mathbb{R}^{(k+s)\times s}$ and $\mathfrak{V} \in \mathbb{R}^{s\times s}$. To be more precise, we consider the first SVD-based KF developed in [8].

Algorithm 2. SVD-SRKF (*SVD-based square-root KF*)

INITIALIZATION: ($k = 0$)

1 Apply SVD factorization: $\Pi_0 = Q_{\Pi_0} D_{\Pi_0} Q_{\Pi_0}^T$.

2 Set $\hat{x}_{0|0} = \bar{x}_0$ and $Q_{P_{0|0}} = Q_{\Pi_0}$, $D_{P_{0|0}}^{1/2} = D_{\Pi_0}^{1/2}$.

TIME UPDATE: ($k = \overline{1,N}$)

3 $\quad$ Apply Cholesky decomposition:
$\Theta_k = \Theta_k^{1/2}\Theta_k^{T/2}$ and $R_k = R_k^{1/2} R_k^{T/2}$
where $\Theta_k^{1/2}$, $R_k^{1/2}$ are lower triangular matrices.

4 $\quad \hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1} u_{k-1},$

5 $\quad$ Assemble the pre-array and apply SVD as follows:

$$\underbrace{\begin{bmatrix} D_{P_{k-1|k-1}}^{1/2} Q_{P_{k-1|k-1}}^T F_{k-1}^T \\ \Theta_{k-1}^{T/2} G_{k-1}^T \end{bmatrix}}_{\text{Pre-array}} = \underbrace{\mathfrak{W} \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \\ 0 \end{bmatrix} Q_{P_{k|k-1}}^T}_{\text{Post-array SVD factors}}$$

6 $\quad$ Read-off from the post-arrays: $Q_{P_{k|k-1}}$, $D_{P_{k|k-1}}^{1/2}$.

MEASUREMENT UPDATE: ($k = \overline{1,N}$)

7 $\quad$ Assemble the pre-array and apply SVD as follows:

$$\underbrace{\begin{bmatrix} R_k^{-1/2} H_k Q_{P_{k|k-1}} \\ D_{P_{k|k-1}}^{-1/2} \end{bmatrix}}_{\text{Pre-array}} = \underbrace{\mathfrak{W} \begin{bmatrix} D_{P_{k|k}}^{-1/2} \\ 0 \end{bmatrix} \mathfrak{V}_{MU}^T}_{\text{Post-array SVD factors}},$$

8 $\quad$ Read-off from the post-arrays: $D_{P_{k|k}}^{-1/2}$ and $\mathfrak{V}_{MU}$,

9 $\quad Q_{P_{k|k}} = Q_{P_{k|k-1}} \mathfrak{V}_{MU},$

10 $\quad K_k = \left(Q_{P_{k|k}} D_{P_{k|k}}^{1/2} D_{P_{k|k}}^{1/2} Q_{P_{k|k}}^T\right) H_k^T R_k^{-1},$

11 $\quad \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k e_k, \quad e_k = y_k - H_k \hat{x}_{k|k-1}.$

Next, consider the most recent development in this research domain that is the SVD factorization-based KF implementation proposed in [9].

Algorithm 3. SVD-KF (*SVD-based KF*)

INITIALIZATION: ($k = 0$)

1 Apply SVD factorization: $\Pi_0 = Q_{\Pi_0} D_{\Pi_0} Q_{\Pi_0}^T$,

2 Set $\hat{x}_{0|0} = \bar{x}_0$ and $Q_{P_{0|0}} = Q_{\Pi_0}$, $D_{P_{0|0}}^{1/2} = D_{\Pi_0}^{1/2}$.

TIME UPDATE: ($k = \overline{1,N}$) Apply SVD to the matrices

3 $\quad \Theta_k = Q_{\Theta_k} D_{\Theta_k} Q_{\Theta_k}^T$ and $R_k = Q_{R_k} D_{R_k} Q_{R_k}^T$.

4 $\quad \hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + B_{k-1} u_{k-1},$

5 $\quad$ Assemble the pre-array and apply SVD as follows:

$$\underbrace{\begin{bmatrix} D_{P_{k-1|k-1}}^{1/2} Q_{P_{k-1|k-1}}^T F_{k-1}^T \\ D_{\Theta_{k-1}}^{1/2} Q_{\Theta_{k-1}}^T G_{k-1}^T \end{bmatrix}}_{\text{Pre-array}} = \underbrace{\mathfrak{W} \begin{bmatrix} D_{P_{k|k-1}}^{1/2} \\ 0 \end{bmatrix} Q_{P_{k|k-1}}^T}_{\text{Post-array SVD factors}}$$

6 $\quad$ Read-off from the post-arrays: $Q_{P_{k|k-1}}$, $D_{P_{k|k-1}}^{1/2}$.

MEASUREMENT UPDATE: ($k = \overline{1,N}$)

7 $\quad$ Assemble the pre-array and apply SVD as follows:

$$\underbrace{\begin{bmatrix} D_{R_k}^{1/2} Q_{R_k}^T \\ D_{P_{k|k-1}}^{1/2} Q_{P_{k|k-1}}^T H_k^T \end{bmatrix}}_{\text{Pre-array}} = \underbrace{\mathfrak{W} \begin{bmatrix} D_{R_{e,k}}^{1/2} \\ 0 \end{bmatrix} Q_{R_{e,k}}^T}_{\text{Post-array SVD factors}},$$

8 $\quad$ Read-off from the post-arrays: $Q_{R_{e,k}}$ and $D_{R_{e,k}}^{1/2}$,

9 $\quad \bar{K}_k = P_{k|k-1} H_k^T Q_{R_{e,k}},$

10 $\quad \bar{e}_k = Q_{R_{e,k}}^T e_k, \quad e_k = y_k - H_k \hat{x}_{k|k-1},$

11 $\quad \hat{x}_{k|k} = \hat{x}_{k|k-1} + \bar{K}_k D_{R_{e,k}}^{-1} \bar{e}_k,$

12 $\quad$ Assemble the pre-array and apply SVD as follows:

$$\underbrace{\begin{bmatrix} D^{1/2}_{P_{k|k-1}} Q^T_{P_{k|k-1}} A^T \\ D^{1/2}_{R_k} Q^T_{R_k} K_k^T \end{bmatrix}}_{\text{Pre}-\text{array}} = \underbrace{\mathfrak{Q} \begin{bmatrix} D^{1/2}_{P_{k|k}} \\ 0 \end{bmatrix} Q^T_{P_{k|k}}}_{\text{Post}-\text{array SVD factors}}$$

where $A = (I - K_k H_k)$, $K_k = \bar{K}_k D^{-1}_{R_{e,k}} Q^T_{R_{e,k}}$,

13 Read-off from the post-arrays: $Q_{P_{k|k}}$ and $D^{1/2}_{P_{k|k}}$.

The readers are referred to [8] and [9] for a detailed derivation of the SVD-SRKF (Algorithm 2) and SVD-KF (Algorithm 3), respectively. It should be stressed that both of them are algebraically equivalent to the conventional KF implementation (Algorithm 1) and, hence, to each other. The proof is not difficult to carry out by taking into account the following two facts. First, the following formulas hold for the conventional KF algorithm [15, p. 128-129]:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (3)$$
$$= P_{k|k} H_k^T R_k^{-1}. \quad (4)$$
$$P_{k|k} = \left(P^{-1}_{k|k-1} + H_k^T R_k^{-1} H_k \right)^{-1} \quad (5)$$
$$= (I - K_k H_k) P_{k|k-1} \quad (6)$$
$$= (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T. \quad (7)$$

Second, taking into account that $A = \mathfrak{W}\Sigma\mathfrak{V}^T$ where $\mathfrak{W}$ and $\mathfrak{V}$ are orthogonal matrices, one obtains $A^T A = (\mathfrak{V}\Sigma\mathfrak{W}^T)(\mathfrak{W}\Sigma\mathfrak{V}^T) = \mathfrak{V}\Sigma^2\mathfrak{V}^T$ for each filtering pre-array to be factorized. Thus, by comparing both sides of the obtained matrix equalities in Algorithms 2 and 3, the related SVD-based KF equations are proved. In summary, formula in line 7 of the SVD-SRKF (Algorithm 2) implies equation (5) for computing *a posteriori* error covariance matrix and, hence, formula in line 10 of Algorithm 2 is, in fact, equation (4) for computing the feedback gain K_k. Unlikely, the SVD-KF formula for calculating *a posteriori* error covariance matrix in line 12 of Algorithm 3 implies equation (7). This symmetric reformulation of the discrete-time Riccati equation suggested in [16] is called the Joseph stabilized form. The symmetric form enhances the filter robustness against roundoff errors in some applications, although the *factored-form* algorithms perform better when solving ill-conditioned state estimation problems; see Example 7.2 and the obtained numerical results illustrated by Fig. 7.1. in [2, Chapter 7]. Hence, both the SVD-SRKF (Algorithm 2) and the SVD-KF (Algorithm 3) are algebraically equivalent to the classical KF implementation in Algorithm 1. However, the numerical behaviour of these methods is no longer agree. Table I summarizes main theoretical properties of the SVD-based KF implementations to be examined. The following notations are used: sign "+" means that the corresponding property is available, "–" implies that the corresponding property is absent, and the question sign "?" means that the corresponding question is still open and, hence, the SVD-based filter with this underlying property might be derived.

Having analyzed the information presented in Table I, the following few findings are revealed. Firstly, both the SVD-SRKF (Algorithm 2) and the SVD-KF (Algorithm 3) are developed in the so-called *covariance* form. This means that the SVD factors of the state error covariance $P_{k|k}$ are propagated in each iteration step. An alternative class of methods propagates the inverse of $P_{k|k}$ (called the information matrix) rather than propagating $P_{k|k}$. Such KF implementations are called *information*-type KF methods. The information filtering has been derived for solving the so-called problems without *prior* information, i.e. when the error covariance matrix at the initialization stage, Π_0, is too "large". Recall, the value $P_{k|k}$ represents the uncertainty in the state estimate $\hat{x}_{k|k}$ and, hence, the term $P^{-1}_{k|k}$ represents the certainty in $\hat{x}_{k|k}$. Additionally, the information filtering suggests a plausible solution to roundoff problem when the measurement update stage is a source of the problem as discussed in [2, p. 356-357]. To the best of author's knowledge, the *information* SVD-based filtering does not exist, so far. This is an open question in the family of SVD-based KF methods.

TABLE I
THEORETICAL COMPARISON OF THE SVD−SRKF AND SVD−KF IMPLEMENTATIONS

No.	Property	SVD−SRKF (Algorithm 2)	SVD−KF (Algorithm 3)
1.	Type of the filtering method:		
	∘ Covariance filtering	+	+
	∘ Information filtering	?	?
2.	Pre-array SVD factorizations:		
	∘ at Time Update	1	1
	∘ at Measurement Update	1	2
3.	Straightforward LF evaluation	–	+
4.	Matrix decompositions involved:		
	∘ initial error covariance Π_0	SVD	SVD
	∘ noise covariances Θ_k and R_k	Cholesky	SVD
5	Filter feasibility (Π_0, Θ_k, R_k):		
	∘ positive definiteness	+	+
	∘ positive semi-definiteness	–	+
6.	Matrix inversions required:	$D^{-1/2}_{P_{k\|k-1}}$, $D^{-1/2}_{P_{k\|k}}$, R_k^{-1}	$D^{-1/2}_{R_{e,k}}$
7.	Extended array form (avoids any matrix inversion)	?	?
8.	Numerical stability		
	∘ vs. the conventional KF	improved	improved
	∘ vs. the factored-form KFs	deteriorated	similar
9.	First-order error propagation	?	?

Second, we conclude that the SVD-SRKF (Algorithm 2) requires one SVD factorization less than the SVD-KF (Algorithm 3) and, hence, Algorithm 2 should be faster than Algorithm 3. However, this additional SVD factorization appeared at the measurement update stage of Algorithm 3 allows for a simple and straightforward calculation of the likelihood function (LF), since the required values $Q_{R_{e,k}}$ and $D^{1/2}_{R_{e,k}}$ are directly available from the SVD-KF; see formula (25) in [9]. In contrast, the SVD-SRKF (Algorithm 2) does not provide a possibility for straightforward LF computation.

Next, the SVD-SRKF (Algorithm 2) requires Cholesky factorization for the covariance matrices Θ_k and R_k, meanwhile the SVD-KF (Algorithm 3) implies their SVD factorization. Although the SVD is more time consuming than the Cholesky decomposition, its utilization makes the SVD-KF

(Algorithm 3) applicable for any matrices $\Theta_k \geq 0$ and $R_k \geq 0$. Meanwhile, the SVD-SRKF (Algorithm 2) works only when $\Theta_k \geq 0$ and $R_k \geq 0$ are positive definite matrices to ensure the existence and uniqueness of Cholesky decomposition [17]. The Cholesky decomposition exists for positive semi-definite matrices, however, it is not unique in this case [18]. Thus, the SVD-SRKF (Algorithm 2) might be abruptly interrupted by an error appeared while performing the Cholesky decomposition. Hence, Algorithm 3 is preferable for practical implementation, although it is expected to be slower than Algorithm 2 because of utilizing SVD factorization instead of Cholesky decomposition. It should be stressed that in case of time-invariant systems, the corresponding factorization of covariance matrices Θ and R is performed only once, i.e. it is pre-computed at the initialization stage. Hence, for constant (over time) covariances Θ and R, the difference in time consumption between Algorithms 2 and 3 will be negligible.

Concerning the robustness of the SVD-based KF algorithms, both the SVD-SRKF (Algorithm 2) and SVD-KF (Algorithm 3) are more numerically stable than the conventional KF implementation. Algorithm 3 shows similar numerical behaviour as all *factored*-form KF methods when solving ill-conditioned state estimation problems. It also outperforms the SVD-SRKF (Algorithm 2) for estimation accuracy and robustness [9]. However, to quantify theoretically the effect of roundoff on the underlying KF recursion, the first-order error propagation model should be derived for the SVD-based implementations as discussed in [19], [20]. This methodology provides upper bounds on the propagation of roundoff errors and, hence, clearly indicates that some implementations possess better bounds than others. For the family of SVD-based KF algorithms such theoretical investigation could be an area for a future research.

Following [2, p. 288], one of the sources of numerical instability of the KF is an inversion of the residual covariance matrix $R_{e,k}$ at the measurement update stage. The *factored*-form KF implementations (the SR-, the UD- and the SVD-based algorithms) typically require the inversion of the corresponding factors of this matrix rather than the full matrix $R_{e,k}$. Indeed, as can be seen from line 11 of Algorithm 3, the SVD-KF requires $D_{R_{e,k}}^{-1/2}$ calculation. The KF implementations can be improved further by avoiding any matrix inversion operation. The key idea is to express the corresponding filtering equations in the *extended array* form. This implies the utilization of numerically stable orthogonal transformations for computing the state estimate as shown in [7]. Under this strategy, the dynamic state is computed by a simple multiplication of the blocks that are read off from the *extended* post-array, directly. Thus, no matrix inversion operation is required; see [7, eq. (6)] and the explanation below that formula. The extended array implementations exist for the Cholesky-based filtering [7] and UD-based methods [21], [22]. The question whether or not it is possible to design the *extended* SVD-based implementations (to avoid any inversion of the SVD factors) is still open.

Finally, form Table I we see that the SVD-SRKF (Algorithm 2) requires the inversion of $D_{P_{k|k-1}}^{1/2}$, $D_{P_{k|k}}^{1/2}$ and R_k, meanwhile the SVD-KF (Algorithm 3) involves only $D_{R_{e,k}}^{-1/2}$ calculation. For time-invariant measurement noise covariance matrix R, the value R^{-1} can be pre-computed before the KF recursion. All other matrices are to be inverted at each iteration step of the filter. This partially explains a better numerical behavior of the SVD-KF (Algorithm 3) compared to the SVD-SRKF (Algorithm 2), because less matrix inversions are required in Algorithm 3.

We note that the matrices $D_{P_{k|k-1}}^{1/2}$, $D_{P_{k|k}}^{1/2}$ and $D_{R_{e,k}}^{1/2}$ are diagonal matrices and, hence, their inversion is a simple n and m divisions by the corresponding diagonal elements. Recall, in the SVD-based filters these diagonal entries are the square roots of sigma values of the corresponding matrices. The zero entries might also appear in $D_{P_{k|k-1}}^{1/2}$, $D_{P_{k|k}}^{1/2}$ and $D_{R_{e,k}}^{1/2}$. When implementing the SVD-based algorithms, the scalar divisions should be performed for the non-zero diagonal elements, only. If the matrix is close to singular, then some of the sigma values are very small numbers and, hence, the division causes large roundoff errors. To enhance the numerical robustness of the SVD-based algorithms, it is suggested to perform scalar divisions only for the square roots of sigma values that are larger than the unit roundoff error. Computer roundoff for floating-point arithmetic is often characterized by a single parameter $\epsilon_{roundoff}$, defined as the largest number such that either $1+\epsilon_{roundoff} = 1$ or $1+\epsilon_{roundoff}/2 = 1$ in machine precision. When implementing in Matlab, this means that we divide only by diagonal entries that are larger than `eps(class(A))` where A is the given variable. The resulted numerically robust SVD-based KF implementations are abbreviated to the NRSVD-SRKF and NRSVD-KF.

IV. NUMERICAL EXPERIMENTS

To perform the numerical comparative study, the set of ill-conditioned test problems in [9] is utilized for assessing the SVD-based filters.

Example 1. *A linearized version of a satellite traveling in a circular orbit is given as follows [23]:*

$$x_k = \begin{bmatrix} 1 & 1 & 0.5 & 0.5 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.606 \end{bmatrix} x_{k-1} + w_{k-1}, \quad w_k \sim \mathcal{N}(0, \Theta)$$

where $\Theta = diag\{[0, 0, 0, 0.63 \cdot 10^{-2}]\}$ *and* x_0 *is a zero-mean initial state with* $\Pi_0 = I_4\}$.

The measurement equations obey the following ill-conditioned scheme:

$$y_k = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1+\delta \end{bmatrix} x_k + v_k, \quad v_k \sim \mathcal{N}(0, \delta^2 I_2)$$

where for simulating the roundoff we assume that $\delta^2 < \epsilon_{roundoff}$*, but* $\delta > \epsilon_{roundoff}$*. The term* $\epsilon_{roundoff}$ *denotes the unit roundoff error.*

The following KF implementation methods are explored:

- `KF` (the conventional implementation) in Algorithm 1;

- SR-KF (the Cholesky-based method) designed in [13, p. 434-436];
- UD-KF (the UD-based method) presented in [2, p. 261];
- SVD-SRKF (the SVD-based method) developed in [8];
- SVD-KF (the SVD-based method) developed in [9];
- the newly proposed robust nrSVD-SRKF version of the SVD-SRKF;
- the newly proposed robust nrSVD-KF version of the SVD-KF;

The following set of numerical experiments is performed. For each value of ill-conditioning parameter δ (where $\delta \to \epsilon_{roundoff}$), the system is simulated for $k = 1, \dots, N$ with $N = 100$ discrete time points. When the "true" trajectory of the dynamic state x_k^{exact} and the related measurements y_k are generated, $k = 1, \dots, N$. Next, the examined filtering methods are applied for solving the inverse problem: given simulated measurements, each filter under assessment yields the state vector estimate $\hat{x}_{k|k}$, $k = 1, \dots, N$. For a fair comparative study, the same filtering initial values, the same system matrices and the same measurements are passed to all KF implementations listed above. The outlined experiment is repeated for $M = 500$ Monte-Carlo trials and, finally, the root *mean* square error (RMSE) is computed as follows:

$$\mathrm{RMSE}_{x_i} = \sqrt{\frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{i,k}^{j,exact} - \hat{x}_{i,k|k}^{j} \right)^2}$$

where $M = 500$ is the number of Monte-Carlo runs, $N = 100$ is the discrete time of the dynamic system, the $x_{i,k}^{j,exact}$ and $\hat{x}_{i,k|k}^{j}$ are the i-th entry of the "true" state vector (simulated) and its estimated value obtained in the j-th Monte Carlo trial, respectively. The norm $\|\mathrm{RMSE}_x\|_2$ is also reported in Table II.

When the obtained numerical results in Table II are analyzed, the following conclusions hold. For large δ, the problem is well-conditioned and all implementation methods under examination provide the same estimation accuracy. This is in line with the theoretical results on algebraic equivalence of all KF implementation methods. Second, while growing ill-conditioning $\delta \to \epsilon_{roundoff}$, the discrete-time algebraic Riccati equation degradation is observed. It is easy to note that the conventional KF technique possesses the worst performance among the examined filtering algorithms. It fails at $\delta = 10^{-8}$, while all other KF implementations manage this ill-conditioned situation accurately. Next, both the SVD-SRKF method and its nrSVD-SRKF variant fail when $\delta = 10^{-9}$. Thus, the conclusions made in [9] is substantiated, namely: i) the SVD-SRKF possesses the worst performance among all *factored*-form KF algorithms under examination, and ii) the SVD-KF outperforms the previously published SVD-SRKF for estimation accuracy and numerical robustness. Here, the previous comments are enriched in the following way: i) the suggested strategy for improving the SVD-based filters' robustness does not provide any additional benefit for the SVD-SRKF variant, because the nrSVD-SRKF possesses the worst performance among all *factored*-form KF algorithms under examination as well as the original SVD-SRKF, and ii) both the SVD-SRKF and the nrSVD-SRKF are the less accurate methods among all *factored*-form KF algorithms under examination.

In contrast to the SVD-SRKF variant, the way of improving the SVD-based filters' robustness suggested in this paper works nicely for the SVD-KF method. As can be seen from Table II, the new nrSVD-KF (see the last row in Table II) outperforms not only any other SVD-based algorithm (see the results for the SVD-SRKF, the nrSVD-SRKF and SVD-KF), but also outperforms any other *factored*-form KF implementation under examination (the SR-KF and UD-KF algorithms). Indeed, starting from $\delta = 10^{-9}$ and until $\delta = 10^{-16}$, the newly proposed nrSVD-KF algorithm is the most accurate method, because it provides the best estimation quality, i.e. it computes the dynamic state with the smallest estimation error.

In summary, the theoretical expectations discussed in this paper are realized. More precisely, the SVD-KF previously published in [9] is improved in this paper. The resulted nrSVD-KF implementation possesses a better robustness (with respect to roundoff errors) compared to the SVD-KF. An additional benefit of the novel SVD-based filtering via the nrSVD-KF is the best performance among all algorithms in the family of *factored*-form KF implementations, i.e. the nrSVD-KF is more accurate than the Cholesky- and the UD-based approaches.

V. Conclusion

In this paper, the class of SVD-based KF implementations is investigated. The theoretical properties of the previously proposed SVD-based estimators are discussed in details. Additionally, a new strategy for improving the filters' numerical robustness is suggested. As a result, the most recently developed SVD-based KF implementation is improved further. The newly suggested modification outperforms any other factored-form (square-root) implementation for estimation accuracy in ill-conditioned situations.

References

[1] M. S. Grewal and J. Kain, "Kalman filter implementation with improved numerical properties," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2058–2068, 2010.

[2] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*, 4th ed. New Jersey: Prentice Hall, 2015.

[3] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt, "Discrete square-root filtering: a survey of current techniques," *IEEE Trans. Automat. Contr.*, vol. AC-16, no. 6, pp. 727–735, Dec. 1971.

[4] M. Morf and T. Kailath, "Square-root algorithms for least-squares estimation," *IEEE Trans. Automat. Contr.*, vol. AC-20, no. 4, pp. 487–497, Aug. 1975.

[5] G. J. Bierman, *Factorization Methods For Discrete Sequential Estimation*. New York: Academic Press, 1977.

[6] A. H. Sayed and T. Kailath, "Extended Chandrasekhar recursion," *IEEE Trans. Automat. Contr.*, vol. AC-39, no. 3, pp. 619–622, Mar. 1994.

[7] P. Park and T. Kailath, "New square-root algorithms for Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 40, no. 5, pp. 895–899, May 1995.

[8] L. Wang, G. Libert, and P. Manneback, "Kalman Filter Algorithm based on Singular Value Decomposition," in *Proceedings of the 31st Conference on Decision and Control*. Tuczon, AZ, USA: IEEE, 1992, pp. 1224–1229.

TABLE II
NUMERICAL COMPARISON OF THE KF IMPLEMENTATIONS IN EXAMPLE 2.

Method	The $\|\mathrm{RMSE}_{\mathrm{x_i}}\|_2$, $i = 1, \dots, 4$, while growing ill-conditioning $\delta \to \epsilon_{roundoff}$												
	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}	10^{-13}	10^{-14}	10^{-15}	10^{-16}
`KF`	0.0531	0.0762	0.0716	0.0714	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`
`SR-KF`	0.0531	0.0762	0.0716	0.0714	0.0780	0.0498	0.0990	0.0651	0.0501	0.0698	0.0604	0.0821	0.0335
`UD-KF`	0.0531	0.0762	0.0716	0.0714	0.0780	0.0498	0.0990	0.0651	0.0501	0.0697	0.0588	0.0814	0.0295
`SVD-SRKF`	0.0531	0.0762	0.0716	0.0720	0.1836	> 10	`Inf`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`
`SVD-KF`	0.0531	0.0762	0.0716	0.0714	0.0780	0.0498	0.0990	0.0651	0.0502	0.0698	0.0645	0.0753	0.3784
`nrSVD-SRKF`	0.0531	0.0762	0.0716	0.0720	0.1836	> 10	`Inf`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`	`NaN`
`nrSVD-KF`	0.0531	0.0762	0.0716	0.0714	0.0706	0.0462	0.0813	0.0490	0.0407	0.0638	0.0392	0.0663	0.0292

[9] M. V. Kulikova and J. V. Tsyganova, "Improved discrete-time Kalman filtering within singular value decomposition," *IET Control Theory & Applications*, vol. 11, no. 15, pp. 2412–2418, 2017.

[10] X. Zhang, W. Hu, Z. Zhao, Y. Wang, and Q. Wei, "SVD based Kalman particle filter for robust visual tracking," in *Proceedings of the 19th International Conference on Pattern Recognition*. Tampa, FL, USA: IEEE, 2008, pp. 1–4.

[11] O. Straka, J. Dunik, M. Simandl, and J. Havlik, "Aspects and comparison of matrix decompositions in unscented Kalman filter," in *Proceedings of the IEEE American Control Conference (ACC)*, 2013, pp. 3075–3080.

[12] H. M. T. Menegaz, J. Ishihara, G. A. Borges, and A. N. Vargas, "A systematization of the unscented Kalman filter theory," *IEEE Trans. Automat. Contr.*, vol. 60, no. 10, pp. 2583–2598, 2015.

[13] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. New Jersey: Prentice Hall, 2000.

[14] A. Björck, *Numerical methods in matrix computations*. Springer, 2015.

[15] D. Simon, *Optimal state estimation: Kalman, H-infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[16] R. S. Bucy and P. D. Joseph, *Filtering for Stochastic Processes, with Applications to Guidance*. New York: John Wiley & Sons, 1968.

[17] G. H. Golub and C. F. Van Loan, *Matrix computations*. Baltimore, Maryland: Johns Hopkins University Press, 1983.

[18] N. J. Higham, "Analysis of the Cholesky decomposition of a semi-definite matrix," University of Manchester, Tech. Rep. MIMS EPrint: 2008.56, 1990.

[19] M. Verhaegen and P. Van Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 10, pp. 907–917, Oct. 1986.

[20] M. Verhaegen, "Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes," *Automatica*, vol. 25, no. 3, pp. 437–444, 1989.

[21] S. H. Chun, "A single-chip QR decomposition processor for extended square-root Kalman filters," in *Twenty-Fifth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA*, vol. 1, 1991, pp. 521–529.

[22] I. V. Semushin, Y. V. Tsyganova, A. V. Tsyganov, and E. F. Prokhorova, "Numerically efficient UD filter based channel estimation for OFDM wireless communication technology," *Procedia Engineering*, vol. 201, pp. 726–735, 2017.

[23] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.